

開發人員指南

AWS SDK for Ruby



AWS SDK for Ruby: 開發人員指南

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

Amazon 的商標和商業外觀不得用於任何非 Amazon 的產品或服務，也不能以任何可能造成客戶混淆、任何貶低或使 Amazon 名譽受損的方式使用 Amazon 的商標和商業外觀。所有其他非 Amazon 擁有的商標均為其各自擁有者的財產，這些擁有者可能隸屬於 Amazon，或與 Amazon 有合作關係，或由 Amazon 贊助。

Table of Contents

什麼是 AWS SDK for Ruby ?	1
其他文件與資源	1
部署到AWS雲端	1
開發套件主要版本的維護與支援	2
開始使用	3
使用 SDK 驗證AWS	3
啟動AWS存取入口網站會話	4
更多的驗證證	5
安裝軟體開發套件	5
先決條件	5
安裝 開發套件	5
你好教程	6
撰寫程式碼	6
執行程式	8
視窗使用者注意事項	8
後續步驟	8
與 SDK AWS Cloud9 搭配使用	9
步驟 1：設定您的AWS 帳戶使用 AWS Cloud9	9
步驟 2：設定開AWS Cloud9發環境	9
步驟 3：設置 AWS SDK for Ruby	10
步驟 4：下載範例程式碼	11
步驟 5：執行範例程式碼	12
設定軟體開發套件	13
憑證提供者鏈結	13
創建AWS STS訪問令牌	14
設定區域	15
使用共用config檔案設定區域	15
使用環境變量設置區域	15
設置區域 Aws.config	16
在客戶端或資源對象中設置區域	16
設定非標準端點	16
使用開發套件	17
使用 REPL 公用程式	17
先決條件	17

捆綁器設置	17
執行回應	18
使用 SDK 與 Rails 上的紅寶石	18
偵錯提示：從用戶端取得電線追蹤資訊	19
存根客戶端響應和錯誤	20
存根用戶端回應	20
存根用戶端錯誤	21
分頁	22
分頁響應可枚舉	22
手動處理分頁回應	22
分頁數據類	23
服務員	23
呼叫服務員呼叫一個	23
等待失敗	24
設定一個服務員	24
擴充一個服務員	25
指定用戶端重試行為	25
將第 1 版或第 2 版遷移到 AWS SDK for Ruby 3 版	26
Side-by-side 用法	26
一般差異	26
客戶差異	27
資源差異	28
使用AWS服務	29
帶有指導的代碼示例	29
AWS CloudTrail 範例	30
亞馬遜 CloudWatch 示例	36
AWS CodeBuild 範例	69
Amazon EC2 範例	72
AWS Elastic Beanstalk 範例	126
AWS Identity and Access Management(IAM) 範例	129
AWS KMS 範例	164
AWS Lambda 範例	167
Amazon Polly 例子	172
亞馬遜 RDS 示例	177
Amazon SES 範例	183
Amazon SNS 範例	189

Amazon SQS 範例	193
亞馬遜 WorkDocs 示例	220
程式碼範例	224
動作和案例	224
CloudTrail	225
CloudWatch	229
DynamoDB	241
Amazon EC2	266
Elastic Beanstalk	299
EventBridge	305
AWS Glue	326
IAM	353
Kinesis	410
AWS KMS	412
Lambda	416
Amazon Polly	435
Amazon RDS	439
Amazon S3	443
Amazon SES	472
Amazon SES API V2	477
Amazon SNS	479
Amazon SQS	488
AWS STS	501
Amazon WorkDocs	502
跨服務範例	505
建立應用程式以分析客戶意見回饋	505
安全	506
資料保護	506
身分和存取權管理	507
合規驗證	507
恢復能力	508
基礎設施安全性	508
強制執行最低 TLS 版本	509
正在檢 OpenSSL 版本	509
升級 TLS 支援	509
S3 加密用戶端移轉	510

移轉概觀	510
更新現有用戶端以讀取新格式	510
將加密和解密用戶端移轉至 V2	511
文件歷史記錄	515
.....	dxvi

什麼是 AWS SDK for Ruby ?

歡迎AWS使用適用於 Ruby 的 SDK 開發人員指南。適用 Ruby 的AWS開發套件為所有人提供支援程式AWS 服務，包括 Amazon Simple Storage Service (Amazon S3) Elastic Compute Cloud (Amazon EC2) DynamoDB 和 Amazon DynamoDB 和 Amazon DynamoDB。

AWSSDK for Ruby 開發人員指南提供有關如何安裝、設定和使用 Ruby AWS SDK 來建立使用的 Ruby 應用程式的資訊AWS 服務。

[開始使用適用 AWS SDK for Ruby](#)

其他文件與資源

如需適用於 Ruby 開發人員的 AWS SDK 的詳細資源，請參閱下列內容：

- [AWSSDK 和工具參考指南](#) — 包含 SDK 中常見的設定、功能和其他基礎概念 AWS
- [AWS SDK for RubyAPI 參考資料-第 3 版](#)
- [AWS程式碼範例儲存庫](#) GitHub
- [RubyGems.org](#)-SDK 的最新版本被模塊化為特定於服務的寶石，請點擊這裡
 - [支援的服務](#) — 列出適用於 Ruby 的 AWS SDK 支援的所有寶石
- AWS適用於紅寶石原始碼的 SDKGitHub：
 - [來源與讀我檔案](#)
 - [更改每個寶石下的日誌](#)
 - [從第 2 版移動到第 3 版](#)
 - [問題](#)
 - [核心升級說明](#)
- [開發者博客](#)
- [吉特通道](#)
- [@awsforruby](#) 在推特上

部署到AWS雲端

您可以使用AWS Elastic BeanstalkAWS OpsWorks、和AWS 服務等AWS CodeDeploy將應用程式部署到AWS雲端。如需使用 Elastic Beanstalk 部署 Ruby 應用程式，請參閱開發人員指南中的[使用 EB](#)

[CLI 和 Git 在 Ruby 中部署 Elastic Beanstalk 應用程式](#)。AWS Elastic Beanstalk 如需使用部署 Ruby on Rails 應用程式 AWS OpsWorks，請參閱將 [Ruby on Rails 應用程式部署到 AWS OpsWorks](#)。如需部署 AWS 服務的概觀，請參閱 [上的部署選項概觀 AWS](#)。

開發套件主要版本的維護與支援

如需開發套件主要版本及其基礎相依性之維護與支援的相關資訊，請參閱《[AWS 開發套件及工具參考指南](#)》中的以下內容：

- [AWSSDK 和工具維護政策](#)
- [AWSSDK 和工具版本 Support 對照表](#)

開始使用適用 AWS SDK for Ruby

了解如何安裝、設定及使用 SDK 建立 Ruby 應用程式，以程式設計方式存取AWS資源。

主題

- [使用 SDK 驗證AWS](#)
- [安裝 AWS SDK for Ruby](#)
- [你好教程為 AWS SDK for Ruby](#)
- [AWS Cloud9與 AWS SDK for Ruby 一起使用](#)

使用 SDK 驗證AWS

在開發AWS時，您必須建立程式碼的驗證方式AWS 服務。您可以根據環境和您可用的存取權，以不同的方式設定AWS資源的程式設計AWS存取。

若要選擇驗證方法並針對 SDK 進行設定，請參閱 SDK [和工具參考指南中的驗證和存取](#)。AWS

我們建議在本地開發，並且沒有給予雇主身份驗證方法的新用戶應該設置AWS IAM Identity Center。此方法包括安裝以便AWS CLI於設定，以及定期登入AWS存取入口網站。如果選擇此方法，則在AWSSDK 和工具參考指南中完成 [IAM 身分中心身份驗證](#)的程序後，您的環境應包含以下元素：

- 您可以在AWS CLI執行應用程式之前啟動AWS存取入口網站工作階段。
- 具有設定AWSconfig檔的[共用檔案](#)，其中包含一組可從 SDK 參考的設定值。[default]若要尋找此檔案的位置，請參閱《AWSSDK 和工具參考指南》中的[共用檔案的位置](#)。
- 共用config檔案會[region](#)設定設定。這會設AWS 區域定 SDK 用於AWS要求的預設值。此區域用於未指定與要使用的區域一起指定的 SDK 服務請求。
- SDK 會使用設定檔的 [SSO 權杖提供者組態](#)，在傳送要求之前取得認證AWS。此sso_role_name值是連接至 IAM 身分中心權限集的 IAM 角色，可讓您存取應用程式中AWS 服務使用的角色。

下列範例config檔案顯示使用 SSO 權杖提供者組態設定的預設設定檔。設定檔的sso_session設定是指已命名的[sso-session區段](#)。此sso-session區段包含用來啟動AWS存取入口網站工作階段的設定。

```
[default]
```

```
sso_session = my-sso
sso_account_id = 111122223333
sso_role_name = SampleRole
region = us-east-1
output = json

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

Ruby 適用的AWS SDK 不需要將其他套件 (例如SSO和SSOIDC) 新增至您的應用程式，即可使用 IAM 身分中心驗證。

啟動AWS存取入口網站會話

在執行存取的應用程式之前AWS 服務，您需要 SDK 的使用中存AWS取入口網站工作階段，才能使用 IAM 身分中心身分驗證來解析登入資料。根據您設定的工作階段長度，您的存取最終會過期，SDK 會遇到驗證錯誤。若要登入AWS存取入口網站，請在中執行下列命令AWS CLI。

```
aws sso login
```

如果您遵循指引並具有預設設定檔設定，則不需要使用`--profile`選項呼叫指令。如果您的 SSO 權杖提供者組態使用具名的設定檔，則命令為`aws sso login --profile named-profile`。

若要選擇性地測試您是否已有作用中的工作階段，請執行下列AWS CLI命令。

```
aws sts get-caller-identity
```

如果您的工作階段處於作用中狀態，對此命令的回應會報告共用config檔案中設定的 IAM 身分中心帳戶和權限集。

Note

如果您已經擁有作用中的AWS存取入口網站工作階段並執行`aws sso login`，則不需要提供認證。

登入程序可能會提示您允許AWS CLI存取您的資料。由AWS CLI於建置在適用於 Python 的開發套件之上，因此許可訊息可能會包含不同的botocore名稱。

更多的驗證證

人類使用者具有人類身分，是應用程式的相關人員、管理員、開發人員、操作員和消費者。他們必須要有一個身分，才能存取您的 AWS 環境和應用程式。擁有您組織成員身分的人類使用者，也具有人力身分。

存取時使用臨時認證AWS。您可以為人類使用者使用身分提供者，透過擔任角色 (提供暫時性憑證) 來提供對 AWS 帳戶的聯合存取權。如果是集中式存取管理，我們建議您使用AWS IAM Identity Center (IAM Identity Center) 管理您帳戶的存取權和這些帳戶內的許可。如需詳細資料，請參閱下列各項：

- 如需詳細資訊，請參閱 IAM 使用者指南中的 [IAM 安全最佳實務](#)。
- 若要建立短期AWS登入資料，請參閱 IAM 使用者指南中的 [臨時安全登入資料](#)。
- 若要了解 Ruby 認證提供者的其他AWS SDK，請參閱 AWSSDK 和工具參考指南中的 [標準化認證提供者](#)。

安裝 AWS SDK for Ruby

本節包含適用於 Ruby 之 AWS SDK 的必要條件和安裝說明。

先決條件

在您使用適用於 Ruby 的 AWS SDK 之前，您必須使用進行驗證AWS。如需設定驗證的詳細資訊，請參閱 [使用 SDK 驗證AWS](#)。

安裝 開發套件

您可以像任何紅寶石一樣安裝 Ruby 的 AWS SDK。寶石可在 [RubyGems](#). AWSSDK for Ruby 被設計為模塊化，並由AWS 服務. 安裝整個aws-sdk寶石很大，可能需要一個多小時。

我們建議您只安裝寶石供AWS 服務您使用。這些命名為類似，aws-sdk-*service_abbreviation*並且完整列表可在 Ruby README 文件的 AWS SDK 的「[支持的服務](#)」表中找到。例如，用於與 Amazon S3 服務連接的寶石可直接在下列位 [aws-sdk-s3](#)置取得。

Ruby 版本管理

我們建議您使用 Ruby 版本管理器，而不是使用系統 Ruby，如下所示：

- [RVM](#)

- [紅寶石](#)
- [Rbenv](#)

例如，如果您使用的是 Amazon Linux 2 作業系統，則可以使用下列命令來更新 RVM、列出可用的 Ruby 版本，然後選擇您要用於開發與 AWS SDK for Ruby 的版本。最低要求的 Ruby 版本是 2.3。

```
$ rvm get head
$ rvm list known
$ rvm install ruby-3.1.3
$ rvm --default use 3.1.3
```

捆綁器

如果您使用[捆綁器](#)，以下命令會安裝適用於 Amazon S3 的 Ruby 寶石的AWS開發套件：

1. 安裝捆綁器並創建：Gemfile

```
$ gem install bundler
$ bundle init
```

2. 打開創建的Gemfile並為您的代碼將使用的每個AWS服務 gem 添加一gem行。如需 Amazon S3 範例，請將下列一行新增至檔案底部：

```
gem "aws-sdk-s3"
```

3. 儲存 GEM 檔案。
4. 安裝以下指定的依賴關係Gemfile：

```
$ bundle install
```

你好教程為 AWS SDK for Ruby

使用 AWS SDK for Ruby 向 Amazon S3 打招呼。下列範例會顯示您的 Amazon S3 儲存貯體清單。

撰寫程式碼

將以下代碼複製並粘貼到新的源文件中。將檔案命名為 `hello-s3.rb`。

```
require "aws-sdk-s3"

# Wraps Amazon S3 resource actions.
class BucketListWrapper
  attr_reader :s3_resource

  # @param s3_resource [Aws::S3::Resource] An Amazon S3 resource.
  def initialize(s3_resource)
    @s3_resource = s3_resource
  end

  # Lists buckets for the current account.
  #
  # @param count [Integer] The maximum number of buckets to list.
  def list_buckets(count)
    puts "Found these buckets:"
    @s3_resource.buckets.each do |bucket|
      puts "\t#{bucket.name}"
      count -= 1
      break if count.zero?
    end
    true
  rescue Aws::Errors::ServiceError => e
    puts "Couldn't list buckets. Here's why: #{e.message}"
    false
  end
end

# Example usage:
def run_demo
  wrapper = BucketListWrapper.new(Aws::S3::Resource.new)
  wrapper.list_buckets(25)
end

run_demo if $PROGRAM_NAME == __FILE__
```

AWSSDK for Ruby 被設計為模塊化，並且由AWS 服務. 安裝 gem 之後，Ruby 來源檔案頂端的require陳述式會匯入 Amazon S3 服務的 AWS SDK 類別和方法。如需可用AWS服務 gem 的完整清單，請參閱 Ruby 讀我檔案 AWS SDK 的[支援服務](#)表格。

```
require 'aws-sdk-s3'
```

執行程式

打開命令提示符來運行你的 Ruby 程序。運行 Ruby 程序的典型命令語法是：

```
ruby [source filename] [arguments...]
```

此範例程式碼不使用任何引數。若要執行此程式碼，請在命令提示字元中輸入下列命令：

```
$ ruby hello-s3.rb
```

視窗使用者注意事項

當您在 Windows 上使用 SSL 憑證並執行 Ruby 程式碼時，您可能會看到類似下列內容的錯誤。

```
C:\Ruby>ruby buckets.rb
C:/Ruby200-x64/lib/ruby/2.0.0/net/http.rb:921:in `connect': SSL_connect returned=1
errno=0 state=SSLv3 read server certificate B: certificate verify failed
(Seahorse::Client::NetworkingError)
    from C:/Ruby200-x64/lib/ruby/2.0.0/net/http.rb:921:in `block in connect'

    from C:/Ruby200-x64/lib/ruby/2.0.0/timeout.rb:66:in `timeout'
    from C:/Ruby200-x64/lib/ruby/2.0.0/net/http.rb:921:in `connect'
    from C:/Ruby200-x64/lib/ruby/2.0.0/net/http.rb:862:in `do_start'
    from C:/Ruby200-x64/lib/ruby/2.0.0/net/http.rb:857:in `start'
...

```

若要修正此問題，請在第一次AWS呼叫之前，將下列行新增至 Ruby 原始碼檔案。

```
Aws.use_bundled_cert!
```

如果您只在 Ruby 程序中使用 `aws-sdk-s3` gem 並且想要使用捆綁的證書，則還需要添加 `aws-sdk-core` gem。

後續步驟

若要測試許多其他 Amazon S3 操作，請查看上的 [AWS 程式碼範例儲存庫](#) GitHub。

AWS Cloud9與 AWS SDK for Ruby 一起使用

AWS Cloud9是一種 Web 整合開發環境 (IDE)，可用來編寫、建置、執行、測試、偵錯和發行雲端軟體的一組工具。您可以使AWS Cloud9用 AWS SDK for Ruby 編寫和使用瀏覽器運行您的 Ruby 代碼。AWS Cloud9包括程式碼編輯器和終端機等工具。由於 AWS Cloud9 IDE 是以雲端為基礎，因此您可以使用連接網際網路的機器，從辦公室、家中或任何地方處理專案。有關的一般資訊AWS Cloud9，請參閱 [《AWS Cloud9使用者指南》](#)。

請依照下列指示來AWS Cloud9設定 Ruby 適用的 AWS SDK：

- [步驟 1：設定您的AWS 帳戶使用 AWS Cloud9](#)
- [步驟 2：設定開AWS Cloud9發環境](#)
- [步驟 3：設置 AWS SDK for Ruby](#)
- [步驟 4：下載範例程式碼](#)
- [步驟 5：執行範例程式碼](#)

步驟 1：設定您的AWS 帳戶使用 AWS Cloud9

若要使用AWS Cloud9，請從登入AWS Cloud9主控台AWS Management Console。

Note

如果您使用AWS IAM Identity Center驗證，則可能需要將所需的權限新增iam:ListInstanceProfilesForRole至 IAM 主控台的使用者附加政策。

若要在您的AWS帳戶中設定 IAM 實體以存取AWS Cloud9並登入AWS Cloud9主控台，請參閱AWS Cloud9使用者指南AWS Cloud9中的 [Team 設定](#)。

步驟 2：設定開AWS Cloud9發環境

當您登入 AWS Cloud9 主控台之後，使用主控台建立 AWS Cloud9 開發環境。在建立環境之後，AWS Cloud9 會開啟該環境的整合開發環境 (IDE)。

[若要取得更多資訊，請參閱《使用指南》AWS Cloud9中的AWS Cloud9〈建立環境〉。](#)

Note

第一次由主控台建立您的環境時，建議您選擇 Create a new instance for environment (EC2) (為環境建立新的執行個體) 選項。此選項可指示建立環境、啟動 Amazon EC2 執行個體，然後將新執行個體連接至新環境。AWS Cloud9這是開始使用 AWS Cloud9 最快的方式。

如果 IDE 中尚未開啟終端機，請開啟終端機。從 IDE 的選單列，選擇 Window, New Terminal (視窗、新增終端機)。您可以使用終端機視窗來安裝工具並建置應用程式。

步驟 3：設置 AWS SDK for Ruby

AWS Cloud9開啟開發環境的 IDE 之後，請使用終端機視窗在您的環境中為 Ruby 設定 AWS SDK。

您可以像任何紅寶石一樣安裝 Ruby 的 AWS SDK。寶石可在 [RubyGems](#). AWSSDK for Ruby 被設計為模塊化，並由AWS 服務. 安裝整個aws-sdk寶石很大，可能需要一個多小時。

我們建議您只安裝寶石供AWS 服務您使用。這些命名為類似，aws-sdk-*service_abbreviation*並且完整列表可在 Ruby README 文件的 AWS SDK 的「[支持的服務](#)」表中找到。例如，用於與 Amazon S3 服務連接的寶石可直接在下列位[aws-sdk-s3](#)置取得。

Ruby 版管理

我們建議您使用 Ruby 版本管理器，而不是使用系統 Ruby，如下所示：

- [RVM](#)
- [紅寶石](#)
- [Rbenv](#)

例如，如果您使用的是 Amazon Linux 2 作業系統，則可以使用下列命令來更新 RVM、列出可用的 Ruby 版本，然後選擇您要用於開發與 AWS SDK for Ruby 的版本。最低要求的 Ruby 版本是 2.3。

```
$ rvm get head
$ rvm list known
$ rvm install ruby-3.1.3
$ rvm --default use 3.1.3
```

捆綁器

如果您使用[捆綁器](#)，以下命令會安裝適用於 Amazon S3 的 Ruby 寶石的AWS開發套件：

1. 安裝捆綁器並創建：Gemfile

```
$ gem install bundler
$ bundle init
```

2. 打開創建的Gemfile並為您的代碼將使用的每個AWS服務 gem 添加一gem行。若要遵循 Amazon S3 範例，請將下列一行新增至檔案底端：

```
gem "aws-sdk-s3"
```

3. 儲存的 GEM 檔。
4. 安裝以下指定的依賴關係Gemfile：

```
$ bundle install
```

步驟 4：下載範例程式碼

使用終端機視窗將 AWS SDK for Ruby 的範例程式碼下載到AWS Cloud9開發環境中。

若要將官方 AWS SDK 文件中使用的所有程式碼範例複本下載到您環境的根目錄中，請執行下列命令：

```
$ git clone https://github.com/awsdocs/aws-doc-sdk-examples.git
```

AWSSDK for Ruby 的程式碼範例位於ENVIRONMENT_NAME/aws-doc-sdk-examples/ruby目錄，其中ENVIRONMENT_NAME是您開發環境的名稱。

若要繼續使用 Amazon S3 範例，我們建議您從程式碼範例開始ENVIRONMENT_NAME/aws-doc-sdk-examples/ruby/example_code/s3/bucket_list.rb。使用終端機視窗導覽至目s3錄並列出檔案。

```
$ cd aws-doc-sdk-examples/ruby/example_code/s3
$ ls
```

要在中打開文件AWS Cloud9，您可以bucket_list.rb直接在終端窗口中單擊。

如需瞭解程式碼範例的詳細支援，請參閱 [Ruby 程式碼範例的 AWS SDK](#)。

步驟 5：執行範例程式碼

要在AWS Cloud9開發環境中運行代碼，請選擇頂部菜單欄中的「運行」按鈕。AWS Cloud9將自動檢測文檔的 .rb 擴展名並使用 Ruby 運行器運行代碼。如需有關在中執行程式碼的詳細資訊，請參閱AWS Cloud9使用指南中的[執行程式碼](#)。

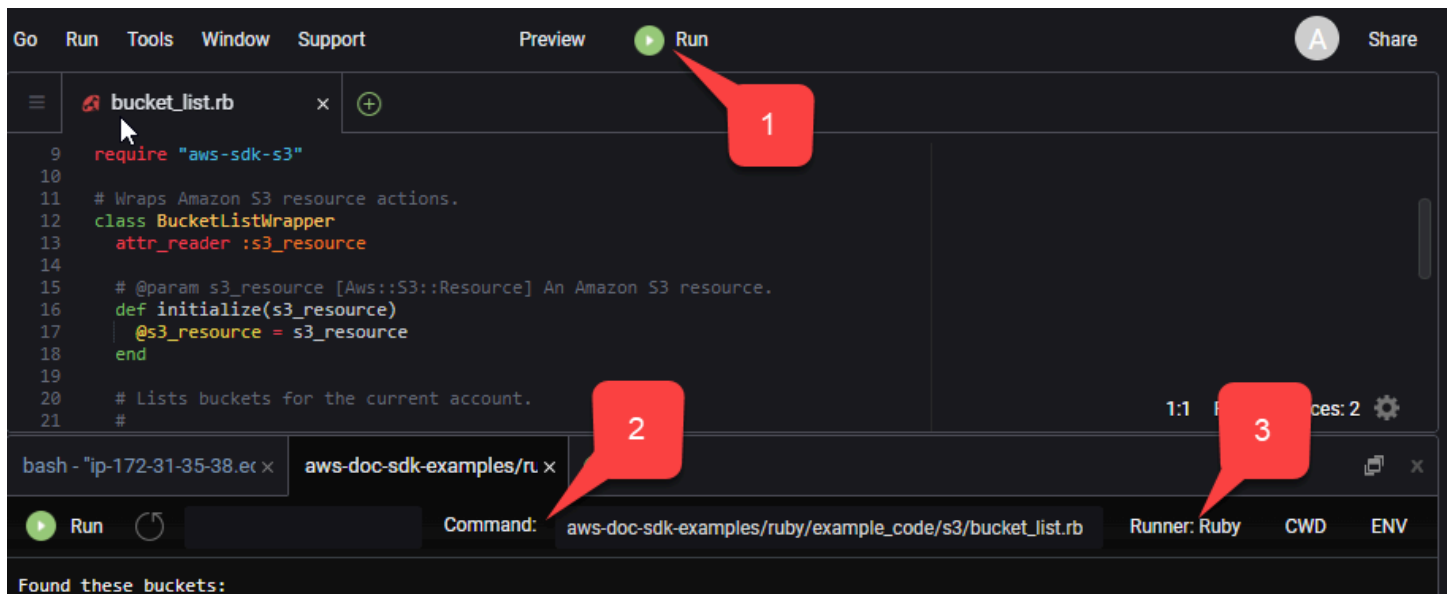
在下列螢幕擷取畫面中，請注意下列基本領域：

- 1：執行。「運行」按鈕位於頂部菜單欄上。這將為您的結果打開一個新標籤。

Note

您也可以手動建立新執行設定。在選單列上，選擇 Run (執行)、Run Configurations (執行組態)、New Run Configuration (新增執行組態)。

- 2：命令。AWS Cloud9在「指令」文字方塊中填入您執行的檔案的路徑和檔案名稱。如果您的程式碼預期傳入任何命令列參數，這些參數可以加入至命令列，就像透過終端機視窗執行程式碼時所做的一樣。
- 3：亞軍。AWS Cloud9檢測到您的文件擴展名是 .rb 並選擇 Ruby Runner 來運行您的代碼。



從執行中程式碼產生的任何輸出都會顯示在索引標籤中。

為紅寶石配置 AWS SDK

了解如何為 Ruby 配置AWS開發套件。在開發AWS時，您必須建立程式碼的驗證方式。AWS 服務您還必須設置AWS 區域要使用的。

憑證提供者鏈結

所有 SDK 都有一系列位置（或來源），他們檢查這些地方（或來源），以獲取有效的憑據以用於向 AWS 服務 找到有效的認證後，就會停止搜尋。此系統搜尋稱為預設認證提供者鏈結。

對於鏈中的每個步驟，都有不同的方法來設定值。直接在代碼中設置值始終優先，然後設置為環境變量，然後在共享AWSconfig文件中設置。如需詳細資訊，請參閱 [AWSSDK 和工具參考指南中的設定優先順序](#)。

AWSSDK 和工具參考指南包含所有 AWS SDK 和 AWS CLI 要了解有關如何通過共AWSconfig享文件配置 SDK 的更多信息，請參閱[共享配置和憑據文件](#)。若要深入瞭解如何透過設定環境變數來設定 SDK，請參閱[環境變數支援](#)。

若要使用驗證AWS，AWSSDK for Ruby 會依照下表中列出的順序檢查認證提供者。

憑證提供者 (依優先)	AWSSDK 和工具參考指南	AWS SDK for Ruby API 參考
靜態認證	AWS存取金鑰	Aws::Credentials Aws::SharedCredentials
來自AWS Security Token Service (AWS STS) 的 Web 身份令牌	假設角色認證提供者 使用role_arnrole_session_name、和web_identity_token_file	Aws::AssumeRoleWebIdentityCredentials
AWS IAM Identity Center。在本指南中，請參閱 使用 SDK 驗證AWS 。	IAM 身分中心憑證提供者	Aws::SSOCredentials
信任的實體提供者 (例如AWS_ROLE_ARN)。在本指	假設角色認證提供者	Aws::AssumeRoleCredentials

憑證提供者 (依優先 南中，請參閱 創建AWS STS訪問令牌 。	AWSSDK 和工具參考指南 使用role_arn和 role_session_name	AWS SDK for Ruby API 參考
程序認證提供者	程序認證提供者	Aws::ProcessCredentials
亞馬遜 Elastic Container Service (Amazon ECS) 登入資料	容器認證提供者	Aws::ECSCredentials
亞馬遜 Elastic Compute Cloud (Amazon EC2) 執行個體設定檔登入資料 (IMDS 登入資料提供者)	IMDS 認證提供者	Aws::InstanceProfileCredentials

如果設定了 Ruby 環境變數 AWS_SDK_CONFIG_OPT_OUT 則不會剖析共用 AWSconfig 檔案 (通常位於 `~/.aws/config`) 以取得認證。

如果您遵循建議的方法讓新使用者開始使用，您可以在 [開始使用] 主題期間 [使用 SDK 驗證 AWS 設定](#) AWS IAM Identity Center 驗證。其他驗證方法在不同的情況下很有用。為了避免安全風險，我們建議您始終使用短期憑證。如需其他驗證方法程序，請參閱 AWSSDK 和工具參考指南中的 [驗證和存取](#)。

創建 AWS STS 訪問令牌

假設角色涉及使用一組臨時安全認證，您可以用來存取通常無法存取的 AWS 資源。這些臨時登入資料由存取金鑰 ID、私密存取金鑰和安全字符組成。您可以使用該 [Aws::AssumeRoleCredentials](#) 方法來創建一個 AWS Security Token Service (AWS STS) 訪問令牌。

下列範例使用存取權杖建立 Amazon S3 用戶端物件，其中 `linked::account::arn` 是要假設角色的 Amazon 資源名稱 (ARN)，並且 `session-name` 是假設角色工作階段的識別碼。

```
role_credentials = Aws::AssumeRoleCredentials.new(
  client: Aws::STS::Client.new,
  role_arn: "linked::account::arn",
  role_session_name: "session-name"
)
```

```
s3 = Aws::S3::Client.new(credentials: role_credentials)
```

有關設置`role_arn`或`role_session_name`有關使用共享AWSconfig文件設置這些信息，請參閱AWSSDK 和工具參考指南中的[假設角色憑證提供者](#)。

設定區域

您需要在大多數時設置一個區域AWS 服務。AWSSDK for Ruby 會依照下列順序搜尋區域：

1. [在客戶端或資源對象中設置區域](#)
2. [使用設定區域 `Aws.config`](#)
3. [使用環境變數設定區域](#)
4. [使用共用`config`檔案設定區域](#)

如需`region`設定的詳細資訊，請參閱《AWSSDK 和工具參考指南》[AWS 區域](#)中的〈〉。本節的其餘部分說明如何設定區域，從最常用的方法開始。

使用共用`config`檔案設定區域

透過在共用AWSconfig檔案中設定`region`變數來設定區域。有關共享文`config`件的更多信息，請參閱 [AWSSDK 和工具參考指南中的共享配置和憑據文件](#)。

在`config`檔案中設定此值的範例：

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

如果設定了環境變數`AWS_SDK_CONFIG_OPT_OUT`，則不會檢查共用`config`檔案。

使用環境變量設置區域

透過設定`AWS_REGION`環境變數來設定「區域」。

使用`export`指令在 UNIX 系統上設定此變數，例如 Linux 或 macOS。下列範例會將 [區域] 設定為`us-west-2`。

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

若要在 Windows 上設定此變數，請使用 `set` 指令。下列範例會將 [區域] 設定為 `us-west-2`。

```
set AWS_REGION=us-west-2
```

設置區域 `Aws.config`

通過向 `Aws.config` 散列添加 `region` 值來設置區域。下列範例會更新 `Aws.config` 雜湊以使用 `us-west-1` Region。

```
Aws.config.update({region: 'us-west-1'})
```

您稍後建立的任何用戶端或資源都會繫結至此區域。

在客戶端或資源對象中設置區域

建立 AWS 用戶端或資源時設定 [區域]。下列範例會在該 `us-west-1` 區域中建立 Amazon S3 資源物件。為您的 AWS 資源選擇正確的區域。服務客戶端對象是不可變的，因此您必須為向其發出請求的每個服務創建一個新的客戶端，並使用不同的配置向同一服務發出請求。

```
s3 = Aws::S3::Resource.new(region: 'us-west-1')
```

設定非標準端點

該區域用於構建用於 AWS 請求的 SSL 端點。如果您需要在所選區域中使用非標準端點，請將 `endpoint` 項目新增至 `Aws.config`。或者，在建立服務用戶端或資源物件 `endpoint:` 時設定。下列範例會在 `other_endpoint` 端點中建立 Amazon S3 資源物件。

```
s3 = Aws::S3::Resource.new(endpoint: other_endpoint)
```

若要使用您選擇的端點來處理 API 要求，並保留該選項，請參閱 [AWSSDK 和工具參考指南中的服務特定端點設定選項](#)。

使用 AWS SDK for Ruby

本節提供使用 AWS SDK for Ruby 開發軟體的相關資訊，包括如何使用 SDK 的某些進階功能。

[AWSSDK 和工具參考指南](#)還包含許多 SDK 中常見的設置，功能和其他基礎概念。AWS

主題

- [使用 AWS SDK 的紅寶石 REPL 實用程序](#)
- [使用 SDK 與 Rails 上的紅寶石](#)
- [偵錯提示：從用戶端取得電線追蹤資訊](#)
- [存根客戶端響應和錯誤](#)
- [分頁](#)
- [服務員](#)
- [指定用戶端重試行為](#)
- [將第 1 版或第 2 版遷移到 AWS SDK for Ruby 3 版](#)

使用 AWS SDK 的紅寶石 REPL 實用程序

aws-sdk 寶石包括一個讀取-發射-打印循環 (REPL) 交互式命令行界面，您可以在其中測試 SDK for Ruby 並立即查看結果。紅寶石寶石的 SDK 可在 RubyGems.org 上找到。

先決條件

- [安裝 AWS SDK for Ruby](#).
- 位於 [aws-v3.rb](#) 於 [aws-sdk-resources](#) 寶石。aws-sdk-resources 寶石也包含在主要 [aws-sdk](#) 寶石中。
- 您將需要一個 xml 庫，例如 [rexml](#) 寶石。
- 雖然該程序確實與交互式 Ruby 外殼 (irb) 一起工作，但我們建議您安裝 pry gem，它提供了更強大的 REPL 環境。

捆綁器設置

如果您使用 [Bundler](#)，以下更新您的 Gemfile 意旨解決先決條件寶石：

1. 打開Gemfile您在安裝 AWS SDK for Ruby 時創建的。在檔案中新增下列各行：

```
gem "aws-sdk"  
gem "rexml"  
gem "pry"
```

2. 保存寶石文件。
3. 安裝在以下指定的依賴關係Gemfile：

```
$ bundle install
```

執行回應

您可以從命令列執行aws-v3.rb來存取 REPL。

```
aws-v3.rb
```

或者，您可以透過設定詳細旗標來啟用 HTTP 配線記錄。HTTP 線路記錄提供有關 AWS SDK 之間的通訊資訊，適用於 Ruby 和AWS。請注意，詳細標誌也會增加開銷，使您的代碼運行速度變慢。

```
aws-v3.rb -v
```

SDK for Ruby 包含提供介面的用戶端類別AWS 服務。每個客戶端類都支持一個特定的AWS 服務。在 REPL 中，每個服務類都有一個幫助程序，該幫助程序返回與該服務進行交互的新客戶端對象。助手的名稱將是轉換為小寫的服務的名稱。例如，Amazon S3 和亞馬 Amazon EC2 助手物件的名稱分別為s3和ec2。若要列出帳戶中的 Amazon S3 儲存貯體，您可以輸s3.list_buckets入提示。

您可以輸quit入 REPL 提示符退出。

使用 SDK 與 Rails 上的紅寶石

[Ruby on Rails](#) 提供了一個 Web 開發框架，可以很容易地使用 Ruby 創建網站。

AWS提供了aws-sdk-rails寶石，以便與 Rails 輕鬆集成。您可以使用AWS Elastic Beanstalk、AWS OpsWorksAWS CodeDeploy、或 [AWSRails 佈建](#)程式在AWS雲端部署和執行 Rails 應用程式。

有關安裝和使用 `aws-sdk-rails` gem 的信息，請參閱存GitHub儲庫 <https://github.com/aws/aws-sdk-rails>。

偵錯提示：從用戶端取得電線追蹤資訊

您可以設定 `http_wire_trace` 布林值，從AWS用戶端取得配線追蹤資訊。配線追蹤資訊有助於區分用戶端變更、服務問題和使用者錯誤。何時 `true`，設定會顯示電線上傳送的內容。下列範例會建立 Amazon S3 用戶端，並在用戶端建立時啟用電線追蹤。

```
s3 = Aws::S3::Client.new(http_wire_trace: true)
```

給定以下代碼和參數 `bucket_name`，輸出會顯示一條消息，指出具有該名稱的存儲桶是否存在。

```
require 'aws-sdk-s3'

s3 = Aws::S3::Resource.new(client: Aws::S3::Client.new(http_wire_trace: true))

if s3.bucket(ARGV[0]).exists?
  puts "Bucket #{ARGV[0]} exists"
else
  puts "Bucket #{ARGV[0]} does not exist"
end
```

如下。（返回已添加到該HEAD行以提高可讀性。）

```
opening connection to bucket_name.s3-us-west-1.amazonaws.com:443...
opened
starting SSL for bucket_name.s3-us-west-1.amazonaws.com:443...
SSL established, protocol: TLSv1.2, cipher: ECDHE-RSA-AES128-GCM-SHA256
-> "HEAD / HTTP/1.1
    Accept-Encoding:
    User-Agent: aws-sdk-ruby3/3.171.0 ruby/3.2.2 x86_64-linux aws-sdk-s3/1.120.0
    Host: bucket_name.s3-us-west-1.amazonaws.com
    X-Amz-Date: 20230427T143146Z
/* omitted */
Accept: */*\r\n\r\n"
-> "HTTP/1.1 200 OK\r\n"
-> "x-amz-id-2: XxB2J+kpHgTjmMUwpkUI1EjaFSPxAjWRgkn/+z7YwWc/
iAX5E30XRBzJ37cfc8T4D7ELC1KFELM=\r\n"
-> "x-amz-request-id: 5MD4APQS815QVBR\r\n"
```

```
-> "Date: Thu, 27 Apr 2023 14:31:47 GMT\r\n"  
-> "x-amz-bucket-region: us-east-1\r\n"  
-> "x-amz-access-point-alias: false\r\n"  
-> "Content-Type: application/xml\r\n"  
-> "Server: AmazonS3\r\n"  
-> "\r\n"  
Conn keep-alive  
Bucket bucket_name exists
```

您也可以在建​​立用戶端後開啟配線追蹤。

```
s3 = Aws::S3::Client.new  
s3.config.http_wire_trace = true
```

若要取得有關報告的配線追蹤資訊中欄位的詳細資訊，請參閱 [Transfer Family 必要的請求標頭](#)。

存根客戶端響應和錯誤

了解如何在 Ruby 應用程式的 AWS SDK 中存根用戶端響應和用戶端錯誤。

存根用戶端回應

當您存根響應時，Ruby 的 AWS SDK 會禁用網絡流量，並且客戶端返回存根（或假的）數據。如果沒有提供存根的數據，則客戶端返回：

- 列表為空數組
- 映射為空哈希
- 數值為零
- 日期為 now

以下範例傳回 Amazon S3 儲存貯體清單的清單名稱。

```
require 'aws-sdk'  
  
s3 = Aws::S3::Client.new(stub_responses: true)  
  
bucket_data = s3.stub_data(:list_buckets, :buckets => [{name:'aws-sdk'}, {name:'aws-  
sdk2'}])
```

```
s3.stub_responses(:list_buckets, bucket_data)
bucket_names = s3.list_buckets.buckets.map(&:name)

# List each bucket by name
bucket_names.each do |name|
  puts name
end
```

運行此代碼顯示以下內容。

```
aws-sdk
aws-sdk2
```

Note

提供任何已記錄的資料後，預設值將不再適用於任何剩餘的實體屬性。這表示在前面的範例中，其餘的實例屬性不再是 `nil`。 `creation_date`

AWSSDK for Ruby 會驗證您的存根資料。如果您傳入錯誤類型的資料，則會引發 `ArgumentError` 例外狀況。例如，如果您使用了下列項目 `bucket_data`，而不是先前的指派：

```
bucket_data = s3.stub_data(:list_buckets, buckets:['aws-sdk', 'aws-sdk2'])
```

AWSSDK for Ruby 引發了兩個 `ArgumentError` 異常。

```
expected params[:buckets][0] to be a hash
expected params[:buckets][1] to be a hash
```

存根用戶端錯誤

您也可以 AWS SDK for Ruby 引發特定方法的虛設常式錯誤。顯示以下範例 `Caught Timeout::Error error calling head_bucket on aws-sdk`。

```
require 'aws-sdk'

s3 = Aws::S3::Client.new(stub_responses: true)
s3.stub_responses(:head_bucket, Timeout::Error)
```

```
begin
  s3.head_bucket({bucket: 'aws-sdk'})
rescue Exception => ex
  puts "Caught #{ex.class} error calling 'head_bucket' on 'aws-sdk'"
end
```

分頁

有些AWS呼叫會提供分頁回應，以限制每個回應傳回的資料量。資料頁面最多可以擁有 1,000 個項目的頁面。

分頁響應可枚舉

處理分頁回應資料的最簡單方法是在回應物件中使用內建的列舉值，如下列範例所示。

```
s3 = Aws::S3::Client.new

s3.list_objects(bucket:'aws-sdk').each do |response|
  puts response.contents.map(&:key)
end
```

這會在每個 API 呼叫中產生一個回應物件，並列舉指定值區中的物件。SDK 會擷取其他頁面的資料以完成要求。

手動處理分頁回應

要自己處理分頁，請使用響應的 `next_page?` 方法來驗證要檢索的頁面更多，或者使用該 `last_page?` 方法來驗證沒有更多頁面可以檢索。

如果有更多頁面，請使用 `next_page` (注意沒有?) 方法來擷取下一頁的結果，如下列範例所示。

```
s3 = Aws::S3::Client.new

# Get the first page of data
response = s3.list_objects(bucket:'aws-sdk')

# Get additional pages
while response.next_page? do
  response = response.next_page
  # Use the response data here...
```

```
end
```

Note

如果您呼叫方 `next_page` 法，且沒有其他頁面可擷取，SDK 會引發 `Aws::PageableResponse: LastPageError` 例外狀況。

分頁數據類

Ruby AWS 開發套件中的分頁資料是由 `Aws::` 類別處理，這個 `PageableResponse` 類別包含在 `海馬::用戶端::回應`，以提供對分頁資料的存取。

服務員

服務員是輪詢在客戶端上發生的特定狀態的實用程序方法。在為服務用戶端定義的輪詢間隔進行多次嘗試之後，服務員可能會失敗。如需如何使用服務員的範例，請參閱程式碼範例儲存庫中 Amazon DynamoDB 加密用戶端的 `create_table` 方法。AWS

呼叫服務員呼叫一個

要調用服務員，請呼 `wait_until` 叫服務客戶端。在下列範例中，服務員會等待執行個體 `i-12345678` 行，然後再繼續執行。

```
ec2 = Aws::EC2::Client.new

begin
  ec2.wait_until(:instance_running, instance_ids:['i-12345678'])
  puts "instance running"
rescue Aws::Waiters::Errors::WaiterFailed => error
  puts "failed waiting for instance running: #{error.message}"
end
```

第一個參數是服務員名稱，這是特定於服務客戶端，並指示正在等待哪個操作。第二個參數是傳遞給服務員調用的客戶端方法的散列參數，該方法根據服務員姓名而變化。

如需可等待的作業清單，以及針對每項作業呼叫的用戶端方法，請參閱您所使用之用戶端的 `waiter_names` 和 `wait_until` 欄位文件。

等待失敗

服務員可能會失敗，但有以下任何例外情況。

[Aws:: 服務員:: 錯誤:: FailureStateError](#)

等待時遇到失敗狀態。

[Aws:: 服務員:: 錯誤:: NoSuchWaiterError](#)

指定的服務員名稱沒有為正在使用的客戶定義。

[Aws:: 服務員:: 錯誤:: TooManyAttemptsError](#)

嘗試次數超過服務員的max_attempts價值。

[Aws:: 服務員:: 錯誤:: UnexpectedError](#)

等待等待時發生未預期的錯誤誤差

[Aws:: 服務員:: 錯誤:: WaiterFailed](#)

其中一個等待狀態已超過，或在等待時發生另一個失敗。

所有這些錯誤（除了 NoSuchWaiterError-）都是基於 WaiterFailed 為了 catch 一個服務員，使用 WaiterFailed，如下列範圍所示。

```
rescue Aws::Waiters::Errors::WaiterFailed => error
  puts "failed waiting for instance running: #{error.message}"
end
```

設定一個服務員

每個服務員都有一個默認的輪詢間隔，並且在將控制權返回到您的程序之前進行的最大嘗試次數。若要設定這些值，請在wait_until呼叫中使用max_attempts和delay:參數。下列範例會等待最多 25 秒，每五秒輪詢一次。

```
# Poll for ~25 seconds
client.wait_until(...) do |w|
  w.max_attempts = 5
  w.delay = 5
end
```

若要停用等待失敗，請將這些參數的值設定為nil。

擴充一個服務員

若要修改等候者的行為，您可以註冊在每次輪詢嘗試之前和等待之前觸發的回呼。

以下示例通過將每次嘗試等待的時間加倍，在服務員中實現指數輪詢。

```
ec2 = Aws::EC2::Client.new

ec2.wait_until(:instance_running, instance_ids:['i-12345678']) do |w|
  w.interval = 0 # disable normal sleep
  w.before_wait do |n, resp|
    sleep(n ** 2)
  end
end
```

下列範例會停用嘗試次數上限，而是在失敗之前等待一小時 (3600 秒)。

```
started_at = Time.now
client.wait_until(...) do |w|
  # Disable max attempts
  w.max_attempts = nil

  # Poll for one hour, instead of a number of attempts
  w.before_wait do |attempts, response|
    throw :failure if Time.now - started_at > 3600
  end
end
```

指定用戶端重試行為

根據預設，AWSSDK for Ruby 最多會執行三次重試，重試間隔 15 秒，最多可嘗試四次。因此，作業最多可能需要 60 秒才能逾時。

下列範例會在該區域中建立 Amazon S3 用戶端us-west-2，並指定在每個用戶端操作上的兩次重試之間等待 5 秒鐘。因此，Amazon S3 用戶端操作最多可能需要 15 秒才能逾時。

```
s3 = Aws::S3::Client.new(
  region: region,
```

```
retry_limit: 2,  
retry_backoff: lambda { |c| sleep(5) }  
)
```

此範例顯示如何直接在程式碼中變更重試參數。不過，您也可以使用環境變數或共用AWSconfig檔案來為您的應用程式設定這些變數。如需這些設定的詳細資訊，請參閱 [AWSSDK 和工具參考指南中的重試行為](#)。在程式碼或服務用戶端上設定的任何明確設定，都會優先於環境變數或共用config檔案中設定的設定。

將第 1 版或第 2 版遷移到 AWS SDK for Ruby 3 版

本主題的目的在於協助您從適用 AWS SDK for Ruby 版本 1 或第 2 版遷移至第 3 版。

Side-by-side 用法

您不需要將 Ruby 的 AWS SDK 版本 1 或第 2 版取代為第 3 版。您可以在同一個應用程式中一起使用它們。如需詳細資訊，請參閱 [此部落格文章](#)。

下面是一個簡單的例子。

```
require 'aws-sdk-v1' # version 1  
require 'aws-sdk'   # version 2  
require 'aws-sdk-s3' # version 3  
  
s3 = AWS::S3::Client.new # version 1  
s3 = Aws::S3::Client.new # version 2 or 3
```

您無需重寫現有的工作版本 1 或 2 代碼即可開始使用第 3 版 SDK。有效的遷移策略是僅針對版本 3 SDK 編寫新代碼。

一般差異

版本 3 與版本 2 在一個重要的方式不同。

- 每個服務都可以作為單獨的寶石提供。

版本 2 與版本 1 在幾個重要方面不同。

- 不同的根命名空間-Aws 與AWS. 這會啟side-by-side用使用。

- `Aws.config`-現在是香草 Ruby 哈希值，而不是方法。
- 嚴格的構造函數選項-在版本 1 SDK 中構建客戶端或資源對象時，將忽略未知的構造函數選項。在版本 2 中，未知的構造函數選項會觸發 `ArgumentError`。例如：

```
# version 1
AWS::S3::Client.new(http_reed_timeout: 10)
# oops, typo'd option is ignored

# version 2
Aws::S3::Client.new(http_reed_timeout: 10)
# => raises ArgumentError
```

客戶差異

版本 2 和版本 3 中的客戶端類之間沒有區別。

第 1 版和第 2 版之間，用戶端類別的外部差異最少。許多服務客戶端在客戶端構建後將具有兼容的接口。部分重要說明：

- `Aws::S3::Client`-版本 1 的 Amazon S3 用戶端類別是手動編碼的。版本 2 是從服務模型產生的。方法名稱和輸入在版本 2 中有很大的不同。
- `Aws::EC2::Client`-版本 2 使用複數名稱作為輸出清單，版本 1 使用尾碼 `_set`。例如：

```
# version 1
resp = AWS::EC2::Client.new.describe_security_groups
resp.security_group_set
#=> [...]

# version 2
resp = Aws::EC2::Client.new.describe_security_groups
resp.security_groups
#=> [...]
```

- `Aws::SWF::Client`-第 2 版使用結構化響應，其中版本 1 使用香草紅寶石哈希值。
- 服務類別重新命名 — 第 2 版針對多個服務使用不同的名稱：
 - `AWS::SimpleWorkflow`已成為 `Aws::SWF`
 - `AWS::ELB`已成為 `Aws::ElasticLoadBalancing`
 - `AWS::SimpleEmailService`已成為 `Aws::SES`

- 用戶端組態選項 — 部分版本 1 組態選項已在版本 2 中重新命名。其他則會移除或取代。以下是主要變更：
 - `:use_ssl` 已移除。第 2 版在任何地方都使用 SSL。要禁用 SSL，您必須配置 `:endpoint` 使用 `http://`。
 - `:ssl_ca_file` 是現在 `:ssl_ca_bundle`
 - `:ssl_ca_path` 是現在 `:ssl_ca_directory`
 - 已新增 `:ssl_ca_store`。
 - `:endpoint` 現在必須是完全合格的 HTTP 或 HTTPS URI，而不是主機名稱。
 - 刪除每個服務的 `:*_port` 選項，現在由替換 `:endpoint`。
 - `:user_agent_prefix` 是現在 `:user_agent_suffix`

資源差異

版本 2 和版本 3 中的資源接口之間沒有差異。

第 1 版和第 2 版中的資源介面之間存在明顯差異。版本 1 完全是手工編碼的，其中版本 2 資源接口是從模型生成的。第 2 版資源介面明顯更一致。一些系統性差異包括：

- 單獨的資源類別 — 在版本 2 中，服務名稱是模組，而不是類別。在本模塊中，它是資源接口：

```
# version 1
s3 = AWS::S3.new

# version 2
s3 = Aws::S3::Resource.new
```

- 引用資源-版本 2 SDK 將集合和個別資源獲取者分為兩種不同的方法：

```
# version 1
s3.buckets['bucket-name'].objects['key'].delete

# version 2
s3.bucket('bucket-name').object('key').delete
```

- Batch 作業 — 在版本 1 中，所有批次作業都是手工編碼的公用程式。在版本 2 中，許多批次作業都是透過 API 自動產生的批次處理作業。第 2 版批次處理介面與版本 1 有很大的不同。

在AWS開發套件AWS 服務中使用紅寶石

以下各節包含討論和範例，說明如何使用 AWS SDK for Ruby 來使用AWS 服務。

如果您是 AWS SDK for Ruby 新手，您可能需要先閱讀這個[開始使用](#)主題。

- [帶有指導的代碼示例](#)— 提供了幾個指導示例AWS 服務。
- [程式碼範例](#)— 提供可用服務範例的完整清單 (但不提供程式碼以外的其他指引)。

所有這些示例的源代碼都可以在[AWS代碼示例存儲庫](#)中下載 GitHub。若要向 AWS 文件團隊提議可考慮產生的新程式碼範例，請建立新請求。該團隊想要產生比僅涵蓋個別 API 呼叫之簡易程式碼更為廣泛的程式碼範例，以涵蓋更為廣泛的案例和使用案例。如需指示，請參閱[上 GitHub的讀我檔案](#)中的「提出新程式碼範例」一節。

包含 Ruby AWS SDK 指引的程式碼範例

本節提供您可以使用 Ruby 的 AWS SDK AWS 服務 來存取的範例。

在上的程式碼範例[儲存庫中尋找這些範例和其他範例的原始AWS程式碼](#) GitHub。

主題

- [CloudTrail 使用 AWS SDK for Ruby 範例](#)
- [亞馬遜 CloudWatch 示例使用 AWS SDK for Ruby](#)
- [CodeBuild 使用 AWS SDK for Ruby 範例](#)
- [Amazon EC2 示例使用AWS開發套件的紅寶石](#)
- [AWS Elastic Beanstalk使用 AWS SDK for Ruby 範例](#)
- [AWS Identity and Access Management\(IAM\) 使用 AWS SDK for Ruby 範例](#)
- [AWS Key Management Service使用 AWS SDK for Ruby 範例](#)
- [AWS Lambda使用 AWS SDK for Ruby 範例](#)
- [Amazon Polly 示例使用 AWS SDK for Ruby](#)
- [亞馬遜 RDS 示例使用AWS開發套件的紅寶石](#)
- [Amazon SES 示例使用AWS開發套件的紅寶石](#)
- [使用 AWS SDK for Ruby 件的 Amazon SNS 示例](#)
- [使用 AWS SDK for Ruby 件的 Amazon SQS 示例](#)

- [亞馬遜 WorkDocs 示例](#)

CloudTrail 使用 AWS SDK for Ruby 範例

CloudTrail 是您AWS 服務可以通過獲取帳戶的 AWS API 調用歷史記錄來監視雲中的AWS部署。您可以使用下面的 AWS SDK 來訪問 Ruby 代碼示例AWS CloudTrail。如需詳細資訊 CloudTrail，請參閱 [AWS CloudTrail; 文件](#)。

主題

- [列出 CloudTrail 軌跡](#)
- [建立系 CloudTrail 統線](#)
- [列出 CloudTrail 路徑活動](#)
- [刪除系 CloudTrail 統線](#)

列出 CloudTrail 軌跡

此範例使用 [describe_trails](#) 方法來列出追蹤的名稱，以及 CloudTrail 儲存區域中資訊的值區。
CloudTrail us-west-2

選擇Copy在本機儲存程式碼。

使用下面的代碼創建文件描述。

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
#
# This file is licensed under the Apache License, Version 2.0 (the "License").
# You may not use this file except in compliance with the License. A copy of the
# License is located at
#
# http://aws.amazon.com/apache2.0/
#
# This file is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS
# OF ANY KIND, either express or implied. See the License for the specific
# language governing permissions and limitations under the License.

require 'aws-sdk-cloudtrail' # v2: require 'aws-sdk'

# Create client in us-west-2
client = Aws::CloudTrail::Client.new(region: 'us-west-2')
```

```
resp = client.describe_trails({})

puts
puts "Found #{resp.trail_list.count} trail(s) in us-west-2:"
puts

resp.trail_list.each do |trail|
  puts 'Name:          ' + trail.name
  puts 'S3 bucket name: ' + trail.s3_bucket_name
  puts
end
```

請參閱 (詳見) 的 [完整實例](#) GitHub。

建立系 CloudTrail 統線

此範例使用 [create_trail](#) 方法在區域中建立 CloudTrail 軌跡。us-west-2 它需要兩個輸入，即追蹤名稱和 CloudTrail 儲存資訊的值區名稱。如果值區沒有適當的政策，請包含 -p 旗標，以便將正確的政策附加到值區。

選擇 Copy 在本機儲存程式碼。

```
# Copyright 2010-2019 Amazon.com, Inc. or its affiliates. All Rights Reserved.
#
# This file is licensed under the Apache License, Version 2.0 (the "License").
# You may not use this file except in compliance with the License. A copy of the
# License is located at
#
# http://aws.amazon.com/apache2.0/
#
# This file is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS
# OF ANY KIND, either express or implied. See the License for the specific
# language governing permissions and limitations under the License.

require 'aws-sdk-cloudtrail' # v2: require 'aws-sdk'
require 'aws-sdk-s3'
require 'aws-sdk-sts'

# Attach IAM policy to bucket
def add_policy(bucket)
  # Get account ID using STS
  sts_client = Aws::STS::Client.new(region: 'us-west-2')
  resp = sts_client.get_caller_identity({})
```

```
account_id = resp.account

# Attach policy to S3 bucket
s3_client = Aws::S3::Client.new(region: 'us-west-2')

begin
  policy = {
    'Version' => '2012-10-17',
    'Statement' => [
      {
        'Sid' => 'AWSCloudTrailAclCheck20150319',
        'Effect' => 'Allow',
        'Principal' => {
          'Service' => 'cloudtrail.amazonaws.com',
        },
        'Action' => 's3:GetBucketAcl',
        'Resource' => 'arn:aws:s3:::' + bucket,
      },
      {
        'Sid' => 'AWSCloudTrailWrite20150319',
        'Effect' => 'Allow',
        'Principal' => {
          'Service' => 'cloudtrail.amazonaws.com',
        },
        'Action' => 's3:PutObject',
        'Resource' => 'arn:aws:s3:::' + bucket + '/AWSLogs/' + account_id + '/*',
        'Condition' => {
          'StringEquals' => {
            's3:x-amz-acl' => 'bucket-owner-full-control',
          },
        },
      },
    ],
  }
}.to_json

s3_client.put_bucket_policy(
  bucket: bucket,
  policy: policy
)

puts 'Successfully added policy to bucket ' + bucket
rescue StandardError => err
  puts 'Got error trying to add policy to bucket ' + bucket + ':'
  puts err
end
```

```
    exit 1
  end
end

# main
name = ''
bucket = ''
attach_policy = false

i = 0

while i < ARGV.length
  case ARGV[i]
    when '-b'
      i += 1
      bucket = ARGV[i]

    when '-p'
      attach_policy = true

    else
      name = ARGV[i]
    end

    i += 1
  end

  if name == '' || bucket == ''
    puts 'You must supply a trail name and bucket name'
    puts USAGE
    exit 1
  end

  if attach_policy
    add_policy(bucket)
  end

  # Create client in us-west-2
  client = Aws::CloudTrail::Client.new(region: 'us-west-2')

  begin
    client.create_trail({
      name: name, # required
      s3_bucket_name: bucket, # required
    })
  end
end
```

```
}))

puts 'Successfully created CloudTrail ' + name + ' in us-west-2'
rescue StandardError => err
  puts 'Got error trying to create trail ' + name + ':'
  puts err
  exit 1
end
```

請參閱 (詳見) 的[完整實例](#) GitHub。

列出 CloudTrail 路徑活動

此範例使用 [lookup_events](#) 方法來列出區域中的 CloudTrail 追蹤事件。us-west-2

選擇Copy在本機儲存程式碼。

```
# Copyright 2010-2019 Amazon.com, Inc. or its affiliates. All Rights Reserved.
#
# This file is licensed under the Apache License, Version 2.0 (the "License").
# You may not use this file except in compliance with the License. A copy of the
# License is located at
#
# http://aws.amazon.com/apache2.0/
#
# This file is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS
# OF ANY KIND, either express or implied. See the License for the specific
# language governing permissions and limitations under the License.

require 'aws-sdk-cloudtrail' # v2: require 'aws-sdk'

def show_event(event)
  puts 'Event name:   ' + event.event_name
  puts 'Event ID:     ' + event.event_id
  puts "Event time:    #{event.event_time}"
  puts 'User name:     ' + event.username

  puts 'Resources:'

  event.resources.each do |r|
    puts '  Name:       ' + r.resource_name
    puts '  Type:       ' + r.resource_type
    puts ''
  end
end
```



```
end

# Create client in us-west-2
client = Aws::CloudTrail::Client.new(region: 'us-west-2')

resp = client.lookup_events()

puts
puts "Found #{resp.events.count} events in us-west-2:"
puts

resp.events.each do |e|
  show_event(e)
end
```

請參閱 (詳見) 的 [完整實例](#) GitHub。

刪除系 CloudTrail 統線

此範例使用 [delete_trail](#) 方法來刪除區域中的 CloudTrail 軌跡。us-west-2它需要一個輸入，即軌跡的名稱。

選擇Copy在本機儲存程式碼。

```
# Copyright 2010-2019 Amazon.com, Inc. or its affiliates. All Rights Reserved.
#
# This file is licensed under the Apache License, Version 2.0 (the "License").
# You may not use this file except in compliance with the License. A copy of the
# License is located at
#
# http://aws.amazon.com/apache2.0/
#
# This file is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS
# OF ANY KIND, either express or implied. See the License for the specific
# language governing permissions and limitations under the License.

require 'aws-sdk-cloudtrail' # v2: require 'aws-sdk'

if ARGV.length != 1
  puts 'You must supply the name of the trail to delete'
  exit 1
end
```

```
name = ARGV[0]

# Create client in us-west-2
client = Aws::CloudTrail::Client.new(region: 'us-west-2')

begin
  client.delete_trail({
    name: name, # required
  })

  puts 'Successfully deleted CloudTrail ' + name + ' in us-west-2'
rescue StandardError => err
  puts 'Got error trying to delete trail ' + name + ':'
  puts err
  exit 1
end
```

請參閱 (詳見) 的 [完整實例](#) GitHub。

亞馬遜 CloudWatch 示例使用 AWS SDK for Ruby

Amazon CloudWatch (CloudWatch) 是 AWS 雲端資源和執行應用程式的監控服務 AWS。您可以使用下面的實例通 CloudWatch 過使用 AWS SDK for Ruby 訪問。如需有關的詳細資訊 CloudWatch，請參閱 [Amazon CloudWatch 文件](#)。

主題

- [獲取有關亞馬遜 CloudWatch 報警信息](#)
- [創建一個亞馬遜 CloudWatch 警報](#)
- [啟用和停用亞馬遜 CloudWatch 警示動作](#)
- [獲取有關亞馬遜自定義指標的信息 CloudWatch](#)
- [將事件發送到亞馬遜 CloudWatch 活動](#)

獲取有關亞馬遜 CloudWatch 報警信息

下列程式碼範例顯示 Amazon 中可用指標警示的相關資訊 CloudWatch。

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX-License-Identifier: Apache-2.0

require 'aws-sdk-cloudwatch'
```

```
# Displays information about available metric alarms in Amazon CloudWatch.
#
# @param cloudwatch_client [Aws::CloudWatch::Client]
#   An initialized CloudWatch client.
# @example
#   describe_metric_alarms(Aws::CloudWatch::Client.new(region: 'us-east-1'))
def describe_metric_alarms(cloudwatch_client)
  response = cloudwatch_client.describe_alarms

  if response.metric_alarms.count.positive?
    response.metric_alarms.each do |alarm|
      puts '-' * 16
      puts 'Name:           ' + alarm.alarm_name
      puts 'State value:      ' + alarm.state_value
      puts 'State reason:     ' + alarm.state_reason
      puts 'Metric:           ' + alarm.metric_name
      puts 'Namespace:        ' + alarm.namespace
      puts 'Statistic:         ' + alarm.statistic
      puts 'Period:           ' + alarm.period.to_s
      puts 'Unit:             ' + alarm.unit.to_s
      puts 'Eval. periods:    ' + alarm.evaluation_periods.to_s
      puts 'Threshold:        ' + alarm.threshold.to_s
      puts 'Comp. operator:   ' + alarm.comparison_operator

      if alarm.key?(:ok_actions) && alarm.ok_actions.count.positive?
        puts 'OK actions:'
        alarm.ok_actions.each do |a|
          puts '  ' + a
        end
      end

      if alarm.key?(:alarm_actions) && alarm.alarm_actions.count.positive?
        puts 'Alarm actions:'
        alarm.alarm_actions.each do |a|
          puts '  ' + a
        end
      end

      if alarm.key?(:insufficient_data_actions) &&
        alarm.insufficient_data_actions.count.positive?
        puts 'Insufficient data actions:'
        alarm.insufficient_data_actions.each do |a|
          puts '  ' + a
        end
      end
    end
  end
end
```

```
        end
      end

      puts 'Dimensions:'
      if alarm.key?(:dimensions) && alarm.dimensions.count.positive?
        alarm.dimensions.each do |d|
          puts '  Name: ' + d.name + ', Value: ' + d.value
        end
      else
        puts '  None for this alarm.'
      end
    end
  else
    puts 'No alarms found.'
  end
end

rescue StandardError => e
  puts "Error getting information about alarms: #{e.message}"
end

# Full example call:
def run_me
  region = ''

  # Print usage information and then stop.
  if ARGV[0] == '--help' || ARGV[0] == '-h'
    puts 'Usage:  ruby cw-ruby-example-show-alarms.rb REGION'
    puts 'Example: ruby cw-ruby-example-show-alarms.rb us-east-1'
    exit 1
  # If no values are specified at the command prompt, use these default values.
  elsif ARGV.count.zero?
    region = 'us-east-1'
  # Otherwise, use the values as specified at the command prompt.
  else
    region = ARGV[0]
  end

  cloudwatch_client = Aws::CloudWatch::Client.new(region: region)
  puts 'Available alarms:'
  describe_metric_alarms(cloudwatch_client)
end

run_me if $PROGRAM_NAME == __FILE__
```

創建一個亞馬遜 CloudWatch 警報

下列程式碼範例會建立新 CloudWatch 警示 (如果已存在具有指定名稱的警示，則更新現有警示)。

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX-License-Identifier: Apache-2.0

require 'aws-sdk-cloudwatch'

# Creates or updates an alarm in Amazon CloudWatch.
#
# @param cloudwatch_client [Aws::CloudWatch::Client]
#   An initialized CloudWatch client.
# @param alarm_name [String] The name of the alarm.
# @param alarm_description [String] A description about the alarm.
# @param metric_name [String] The name of the metric associated with the alarm.
# @param alarm_actions [Array] A list of Strings representing the
#   Amazon Resource Names (ARNs) to execute when the alarm transitions to the
#   ALARM state.
# @param namespace [String] The namespace for the metric to alarm on.
# @param statistic [String] The statistic for the metric.
# @param dimensions [Array] A list of dimensions for the metric, specified as
#   Aws::CloudWatch::Types::Dimension.
# @param period [Integer] The number of seconds before re-evaluating the metric.
# @param unit [String] The unit of measure for the statistic.
# @param evaluation_periods [Integer] The number of periods over which data is
#   compared to the specified threshold.
# @param threshold [Float] The value against which the specified statistic is compared.
# @param comparison_operator [String] The arithmetic operation to use when
#   comparing the specified statistic and threshold.
# @return [Boolean] true if the alarm was created or updated; otherwise, false.
# @example
#   exit 1 unless alarm_created_or_updated?(
#     Aws::CloudWatch::Client.new(region: 'us-east-1'),
#     'ObjectsInBucket',
#     'Objects exist in this bucket for more than 1 day.',
#     'NumberOfObjects',
#     ['arn:aws:sns:us-east-1:111111111111:Default_CloudWatch_Alarms_Topic'],
#     'AWS/S3',
#     'Average',
#     [
#       {
#         name: 'BucketName',
#         value: 'doc-example-bucket'
```

```
#     },
#     {
#       name: 'StorageType',
#       value: 'AllStorageTypes'
#     }
#   ],
#   86_400,
#   'Count',
#   1,
#   1,
#   'GreaterThanThreshold'
# )
def alarm_created_or_updated?(
  cloudwatch_client,
  alarm_name,
  alarm_description,
  metric_name,
  alarm_actions,
  namespace,
  statistic,
  dimensions,
  period,
  unit,
  evaluation_periods,
  threshold,
  comparison_operator
)
  cloudwatch_client.put_metric_alarm(
    alarm_name: alarm_name,
    alarm_description: alarm_description,
    metric_name: metric_name,
    alarm_actions: alarm_actions,
    namespace: namespace,
    statistic: statistic,
    dimensions: dimensions,
    period: period,
    unit: unit,
    evaluation_periods: evaluation_periods,
    threshold: threshold,
    comparison_operator: comparison_operator
  )
  return true
rescue StandardError => e
  puts "Error creating alarm: #{e.message}"
end
```

```
    return false
  end

  # Full example call:
  def run_me
    alarm_name = 'ObjectsInBucket'
    alarm_description = 'Objects exist in this bucket for more than 1 day.'
    metric_name = 'NumberOfObjects'
    # Notify this Amazon Simple Notification Service (Amazon SNS) topic when
    # the alarm transitions to the ALARM state.
    alarm_actions = ['arn:aws:sns:us-
east-1:111111111111:Default_CloudWatch_Alarms_Topic']
    namespace = 'AWS/S3'
    statistic = 'Average'
    dimensions = [
      {
        name: 'BucketName',
        value: 'doc-example-bucket'
      },
      {
        name: 'StorageType',
        value: 'AllStorageTypes'
      }
    ]
    period = 86_400 # Daily (24 hours * 60 minutes * 60 seconds = 86400 seconds).
    unit = 'Count'
    evaluation_periods = 1 # More than one day.
    threshold = 1 # One object.
    comparison_operator = 'GreaterThanThreshold' # More than one object.
    region = 'us-east-1'

    cloudwatch_client = Aws::CloudWatch::Client.new(region: region)

    if alarm_created_or_updated?(
      cloudwatch_client,
      alarm_name,
      alarm_description,
      metric_name,
      alarm_actions,
      namespace,
      statistic,
      dimensions,
      period,
      unit,
```

```

    evaluation_periods,
    threshold,
    comparison_operator
  )
  puts "Alarm '#{alarm_name}' created or updated."
else
  puts "Could not create or update alarm '#{alarm_name}'."
end
end
end

run_me if $PROGRAM_NAME == __FILE__

```

啟用和停用亞馬遜 CloudWatch 警示動作

下面的代碼示例：

1. 建立並啟用新 CloudWatch 警示 (如果已存在具有指定名稱的警示，則更新現有警示)。
2. 停用新的或現有的警報。若要再次啟用鬧鐘，請撥打 `enable_alarm_actions`。

```

# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX-License-Identifier: Apache-2.0

# The following code example shows how to:
# 1. Create or update an Amazon CloudWatch alarm.
# 2. Disable all actions for an alarm.

require 'aws-sdk-cloudwatch'

# Creates or updates an alarm in Amazon CloudWatch.
#
# @param cloudwatch_client [Aws::CloudWatch::Client]
#   An initialized CloudWatch client.
# @param alarm_name [String] The name of the alarm.
# @param alarm_description [String] A description about the alarm.
# @param metric_name [String] The name of the metric associated with the alarm.
# @param alarm_actions [Array] A list of Strings representing the
#   Amazon Resource Names (ARNs) to execute when the alarm transitions to the
#   ALARM state.
# @param namespace [String] The namespace for the metric to alarm on.
# @param statistic [String] The statistic for the metric.
# @param dimensions [Array] A list of dimensions for the metric, specified as
#   Aws::CloudWatch::Types::Dimension.

```



```
# @param period [Integer] The number of seconds before re-evaluating the metric.
# @param unit [String] The unit of measure for the statistic.
# @param evaluation_periods [Integer] The number of periods over which data is
#   compared to the specified threshold.
# @param theshold [Float] The value against which the specified statistic is compared.
# @param comparison_operator [String] The arithmetic operation to use when
#   comparing the specified statistic and threshold.
# @return [Boolean] true if the alarm was created or updated; otherwise, false.
# @example
#   exit 1 unless alarm_created_or_updated?(
#     Aws::CloudWatch::Client.new(region: 'us-east-1'),
#     'ObjectsInBucket',
#     'Objects exist in this bucket for more than 1 day.',
#     'NumberOfObjects',
#     ['arn:aws:sns:us-east-1:111111111111:Default_CloudWatch_Alarms_Topic'],
#     'AWS/S3',
#     'Average',
#     [
#       {
#         name: 'BucketName',
#         value: 'doc-example-bucket'
#       },
#       {
#         name: 'StorageType',
#         value: 'AllStorageTypes'
#       }
#     ],
#     86_400,
#     'Count',
#     1,
#     1,
#     'GreaterThanThreshold'
#   )
def alarm_created_or_updated?(
  cloudwatch_client,
  alarm_name,
  alarm_description,
  metric_name,
  alarm_actions,
  namespace,
  statistic,
  dimensions,
  period,
  unit,
```

```
    evaluation_periods,
    threshold,
    comparison_operator
  )
  cloudwatch_client.put_metric_alarm(
    alarm_name: alarm_name,
    alarm_description: alarm_description,
    metric_name: metric_name,
    alarm_actions: alarm_actions,
    namespace: namespace,
    statistic: statistic,
    dimensions: dimensions,
    period: period,
    unit: unit,
    evaluation_periods: evaluation_periods,
    threshold: threshold,
    comparison_operator: comparison_operator
  )
  return true
rescue StandardError => e
  puts "Error creating alarm: #{e.message}"
  return false
end

# Disables an alarm in Amazon CloudWatch.
#
# Prerequisites.
#
# - The alarm to disable.
#
# @param cloudwatch_client [Aws::CloudWatch::Client]
#   An initialized CloudWatch client.
# @param alarm_name [String] The name of the alarm to disable.
# @return [Boolean] true if the alarm was disabled; otherwise, false.
# @example
#   exit 1 unless alarm_actions_disabled?(
#     Aws::CloudWatch::Client.new(region: 'us-east-1'),
#     'ObjectsInBucket'
#   )
def alarm_actions_disabled?(cloudwatch_client, alarm_name)
  cloudwatch_client.disable_alarm_actions(alarm_names: [alarm_name])
  return true
rescue StandardError => e
  puts "Error disabling alarm actions: #{e.message}"
```

```
    return false
  end

  # Full example call:
  def run_me
    alarm_name = 'ObjectsInBucket'
    alarm_description = 'Objects exist in this bucket for more than 1 day.'
    metric_name = 'NumberOfObjects'
    # Notify this Amazon Simple Notification Service (Amazon SNS) topic when
    # the alarm transitions to the ALARM state.
    alarm_actions = ['arn:aws:sns:us-
east-1:111111111111:Default_CloudWatch_Alarms_Topic']
    namespace = 'AWS/S3'
    statistic = 'Average'
    dimensions = [
      {
        name: 'BucketName',
        value: 'doc-example-bucket'
      },
      {
        name: 'StorageType',
        value: 'AllStorageTypes'
      }
    ]
    period = 86_400 # Daily (24 hours * 60 minutes * 60 seconds = 86400 seconds).
    unit = 'Count'
    evaluation_periods = 1 # More than one day.
    threshold = 1 # One object.
    comparison_operator = 'GreaterThanThreshold' # More than one object.
    region = 'us-east-1'

    cloudwatch_client = Aws::CloudWatch::Client.new(region: region)

    if alarm_created_or_updated?(
      cloudwatch_client,
      alarm_name,
      alarm_description,
      metric_name,
      alarm_actions,
      namespace,
      statistic,
      dimensions,
      period,
      unit,
```

```

    evaluation_periods,
    threshold,
    comparison_operator
  )
  puts "Alarm '#{alarm_name}' created or updated."
else
  puts "Could not create or update alarm '#{alarm_name}'."
end

if alarm_actions_disabled?(cloudwatch_client, alarm_name)
  puts "Alarm '#{alarm_name}' disabled."
else
  puts "Could not disable alarm '#{alarm_name}'."
end
end
end

run_me if $PROGRAM_NAME == __FILE__

```

獲取有關亞馬遜自定義指標的信息 CloudWatch

下面的代碼示例：

1. 將資料點新增至中的自訂量度。 CloudWatch
2. 顯示中測量結果命名空間的可用測量結果清單 CloudWatch。

```

# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX-License-Identifier: Apache-2.0

# The following example shows how to:
# 1. Add a datapoint to a metric in Amazon CloudWatch.
# 2. List available metrics for a metric namespace in Amazon CloudWatch.

require 'aws-sdk-cloudwatch'

# Adds a datapoint to a metric in Amazon CloudWatch.
#
# @param cloudwatch_client [Aws::CloudWatch::Client]
#   An initialized CloudWatch client.
# @param metric_namespace [String] The namespace of the metric to add the
#   datapoint to.
# @param metric_name [String] The name of the metric to add the datapoint to.
# @param dimension_name [String] The name of the dimension to add the

```

```
# datapoint to.
# @param dimension_value [String] The value of the dimension to add the
# datapoint to.
# @param metric_value [Float] The value of the datapoint.
# @param metric_unit [String] The unit of measurement for the datapoint.
# @return [Boolean]
# @example
# exit 1 unless datapoint_added_to_metric?(
#   Aws::CloudWatch::Client.new(region: 'us-east-1'),
#   'SITE/TRAFFIC',
#   'UniqueVisitors',
#   'SiteName',
#   'example.com',
#   5_885.0,
#   'Count'
# )
def datapoint_added_to_metric?(
  cloudwatch_client,
  metric_namespace,
  metric_name,
  dimension_name,
  dimension_value,
  metric_value,
  metric_unit
)
  cloudwatch_client.put_metric_data(
    namespace: metric_namespace,
    metric_data: [
      {
        metric_name: metric_name,
        dimensions: [
          {
            name: dimension_name,
            value: dimension_value
          }
        ],
        value: metric_value,
        unit: metric_unit
      }
    ]
  )
  puts "Added data about '#{metric_name}' to namespace " \
    "'#{metric_namespace}'."
  return true
end
```

```
rescue StandardError => e
  puts "Error adding data about '#{metric_name}' to namespace " \
    "'#{metric_namespace}': #{e.message}"
  return false
end

# Lists available metrics for a metric namespace in Amazon CloudWatch.
#
# @param cloudwatch_client [Aws::CloudWatch::Client]
#   An initialized CloudWatch client.
# @param metric_namespace [String] The namespace of the metric.
# @example
#   list_metrics_for_namespace(
#     Aws::CloudWatch::Client.new(region: 'us-east-1'),
#     'SITE/TRAFFIC'
#   )
def list_metrics_for_namespace(cloudwatch_client, metric_namespace)
  response = cloudwatch_client.list_metrics(namespace: metric_namespace)

  if response.metrics.count.positive?
    response.metrics.each do |metric|
      puts " Metric name: #{metric.metric_name}"
      if metric.dimensions.count.positive?
        puts '   Dimensions:'
        metric.dimensions.each do |dimension|
          puts "     Name: #{dimension.name}, Value: #{dimension.value}"
        end
      else
        puts 'No dimensions found.'
      end
    end
  else
    puts "No metrics found for namespace '#{metric_namespace}'. " \
      'Note that it could take up to 15 minutes for recently-added metrics ' \
      'to become available.'
  end
end

# Full example call:
def run_me
  metric_namespace = 'SITE/TRAFFIC'
  region = 'us-east-1'

  cloudwatch_client = Aws::CloudWatch::Client.new(region: region)
```

```
# Add three datapoints.
puts 'Continuing...' unless datapoint_added_to_metric?(
  cloudwatch_client,
  metric_namespace,
  'UniqueVisitors',
  'SiteName',
  'example.com',
  5_885.0,
  'Count'
)

puts 'Continuing...' unless datapoint_added_to_metric?(
  cloudwatch_client,
  metric_namespace,
  'UniqueVisits',
  'SiteName',
  'example.com',
  8_628.0,
  'Count'
)

puts 'Continuing...' unless datapoint_added_to_metric?(
  cloudwatch_client,
  metric_namespace,
  'PageViews',
  'PageURL',
  'example.html',
  18_057.0,
  'Count'
)

puts "Metrics for namespace '#{metric_namespace}':"
list_metrics_for_namespace(cloudwatch_client, metric_namespace)
end

run_me if $PROGRAM_NAME == __FILE__
```

將事件發送到亞馬遜 CloudWatch 活動

下列程式碼範例顯示如何在 Amazon CloudWatch 事件中建立和觸發規則。只要 Amazon 彈性運算雲端 (Amazon EC2) 中的可用執行個體變更為執行中狀態，此規則就會向 Amazon 簡單通知服務

(Amazon SNS) 中的指定主題傳送通知。此外，相關的事件資訊也會記錄到 CloudWatch 事件中的記錄群組。

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX-License-Identifier: Apache-2.0

# The following code example shows how to create and trigger a rule in
# Amazon CloudWatch Events. This rule sends a notification to the specified
# topic in Amazon Simple Notification Service (Amazon SNS) whenever an
# available instance in Amazon Elastic Compute Cloud (Amazon EC2) changes
# to a running state. Also, related event information is logged to a log group
# in Amazon CloudWatch Logs.
#
# This code example works with the following AWS resources through the
# following functions:
#
# - A rule in Amazon CloudWatch Events. See the rule_exists?, rule_found?,
#   create_rule, and display_rule_activity functions.
# - A role in AWS Identity and Access Management (IAM) to allow the rule
#   to work with Amazon CloudWatch Events. See role_exists?, role_found?,
#   and create_role.
# - An Amazon EC2 instance, which triggers the rule whenever it is restarted.
#   See instance_restarted?.
# - A topic and topic subscription in Amazon SNS for the rule to send event
#   notifications to. See topic_exists?, topic_found?, and create_topic.
# - A log group in Amazon CloudWatch Logs to capture related event information.
#   See log_group_exists?, log_group_created?, log_event, and display_log_data.
#
# This code example requires the following AWS resources to exist in advance:
#
# - An Amazon EC2 instance to restart, which triggers the rule.
#
# The run_me function toward the end of this code example calls the
# preceding functions in the correct order.

require 'aws-sdk-sns'
require 'aws-sdk-iam'
require 'aws-sdk-cloudwatchevents'
require 'aws-sdk-ec2'
require 'aws-sdk-cloudwatch'
require 'aws-sdk-cloudwatchlogs'
require 'securerandom'
```



```
# Checks whether the specified Amazon Simple Notification Service
# (Amazon SNS) topic exists among those provided to this function.
# This is a helper function that is called by the topic_exists? function.
#
# @param topics [Array] An array of Aws::SNS::Types::Topic objects.
# @param topic_arn [String] The Amazon Resource Name (ARN) of the
#   topic to find.
# @return [Boolean] true if the topic ARN was found; otherwise, false.
# @example
#   sns_client = Aws::SNS::Client.new(region: 'us-east-1')
#   response = sns_client.list_topics
#   if topic_found?(
#     response.topics,
#     'arn:aws:sns:us-east-1:111111111111:aws-doc-sdk-examples-topic'
#   )
#     puts 'Topic found.'
#   end
def topic_found?(topics, topic_arn)
  topics.each do |topic|
    return true if topic.topic_arn == topic_arn
  end
  return false
end

# Checks whether the specified topic exists among those available to the
# caller in Amazon Simple Notification Service (Amazon SNS).
#
# @param sns_client [Aws::SNS::Client] An initialized Amazon SNS client.
# @param topic_arn [String] The Amazon Resource Name (ARN) of the
#   topic to find.
# @return [Boolean] true if the topic ARN was found; otherwise, false.
# @example
#   exit 1 unless topic_exists?(
#     Aws::SNS::Client.new(region: 'us-east-1'),
#     'arn:aws:sns:us-east-1:111111111111:aws-doc-sdk-examples-topic'
#   )
def topic_exists?(sns_client, topic_arn)
  puts "Searching for topic with ARN '#{topic_arn}'..."
  response = sns_client.list_topics
  if response.topics.count.positive?
    if topic_found?(response.topics, topic_arn)
      puts 'Topic found.'
      return true
    end
  end
end
```

```

while response.next_page? do
  response = response.next_page
  if response.topics.count.positive?
    if topic_found?(response.topics, topic_arn)
      puts 'Topic found.'
      return true
    end
  end
end
puts 'Topic not found.'
return false
rescue StandardError => e
  puts "Topic not found: #{e.message}"
  return false
end

# Creates a topic in Amazon Simple Notification Service (Amazon SNS)
# and then subscribes an email address to receive notifications to that topic.
#
# @param sns_client [Aws::SNS::Client] An initialized Amazon SNS client.
# @param topic_name [String] The name of the topic to create.
# @param email_address [String] The email address of the recipient to notify.
# @return [String] The Amazon Resource Name (ARN) of the topic that
#   was created.
# @example
#   puts create_topic(
#     Aws::SNS::Client.new(region: 'us-east-1'),
#     'aws-doc-sdk-examples-topic',
#     'mary@example.com'
#   )
def create_topic(sns_client, topic_name, email_address)
  puts "Creating the topic named '#{topic_name}'..."
  topic_response = sns_client.create_topic(name: topic_name)
  puts "Topic created with ARN '#{topic_response.topic_arn}'."
  subscription_response = sns_client.subscribe(
    topic_arn: topic_response.topic_arn,
    protocol: 'email',
    endpoint: email_address,
    return_subscription_arn: true
  )
  puts 'Subscription created with ARN ' \
    "'#{subscription_response.subscription_arn}'. Have the owner of the " \
    "email address '#{email_address}' check their inbox in a few minutes " \

```

```
    'and confirm the subscription to start receiving notification emails.'
    return topic_response.topic_arn
  rescue StandardError => e
    puts "Error creating or subscribing to topic: #{e.message}"
    return 'Error'
  end

# Checks whether the specified AWS Identity and Access Management (IAM)
# role exists among those provided to this function.
# This is a helper function that is called by the role_exists? function.
#
# @param roles [Array] An array of Aws::IAM::Role objects.
# @param role_arn [String] The Amazon Resource Name (ARN) of the
#   role to find.
# @return [Boolean] true if the role ARN was found; otherwise, false.
# @example
#   iam_client = Aws::IAM::Client.new(region: 'us-east-1')
#   response = iam_client.list_roles
#   if role_found?(
#     response.roles,
#     'arn:aws:iam::111111111111:role/aws-doc-sdk-examples-ec2-state-change'
#   )
#     puts 'Role found.'
#   end
def role_found?(roles, role_arn)
  roles.each do |role|
    return true if role.arn == role_arn
  end
  return false
end

# Checks whether the specified role exists among those available to the
# caller in AWS Identity and Access Management (IAM).
#
# @param iam_client [Aws::IAM::Client] An initialized IAM client.
# @param role_arn [String] The Amazon Resource Name (ARN) of the
#   role to find.
# @return [Boolean] true if the role ARN was found; otherwise, false.
# @example
#   exit 1 unless role_exists?(
#     Aws::IAM::Client.new(region: 'us-east-1'),
#     'arn:aws:iam::111111111111:role/aws-doc-sdk-examples-ec2-state-change'
#   )
def role_exists?(iam_client, role_arn)
```

```
puts "Searching for role with ARN '#{role_arn}'..."
response = iam_client.list_roles
if response.roles.count.positive?
  if role_found?(response.roles, role_arn)
    puts 'Role found.'
    return true
  end
while response.next_page? do
  response = response.next_page
  if response.roles.count.positive?
    if role_found?(response.roles, role_arn)
      puts 'Role found.'
      return true
    end
  end
end
end
puts 'Role not found.'
return false
rescue StandardError => e
  puts "Role not found: #{e.message}"
  return false
end

# Creates a role in AWS Identity and Access Management (IAM).
# This role is used by a rule in Amazon CloudWatch Events to allow
# that rule to operate within the caller's account.
# This role is designed to be used specifically by this code example.
#
# @param iam_client [Aws::IAM::Client] An initialized IAM client.
# @param role_name [String] The name of the role to create.
# @return [String] The Amazon Resource Name (ARN) of the role that
#   was created.
# @example
#   puts create_role(
#     Aws::IAM::Client.new(region: 'us-east-1'),
#     'aws-doc-sdk-examples-ec2-state-change'
#   )
def create_role(iam_client, role_name)
  puts "Creating the role named '#{role_name}'..."
  response = iam_client.create_role(
    assume_role_policy_document: {
      'Version': '2012-10-17',
      'Statement': [
```

```
    {
      'Sid': '',
      'Effect': 'Allow',
      'Principal': {
        'Service': 'events.amazonaws.com'
      },
      'Action': 'sts:AssumeRole'
    }
  ]
}.to_json,
path: '/',
role_name: role_name
)
puts "Role created with ARN '#{response.role.arn}'."
puts 'Adding access policy to role...'
iam_client.put_role_policy(
  policy_document: {
    'Version': '2012-10-17',
    'Statement': [
      {
        'Sid': 'CloudWatchEventsFullAccess',
        'Effect': 'Allow',
        'Resource': '*',
        'Action': 'events:*'
      },
      {
        'Sid': 'IAMPassRoleForCloudWatchEvents',
        'Effect': 'Allow',
        'Resource': 'arn:aws:iam::*:role/AWS_Events_Invoke_Targets',
        'Action': 'iam:PassRole'
      }
    ]
  }.to_json,
  policy_name: 'CloudWatchEventsPolicy',
  role_name: role_name
)
puts 'Access policy added to role.'
return response.role.arn
rescue StandardError => e
  puts "Error creating role or adding policy to it: #{e.message}"
  puts 'If the role was created, you must add the access policy ' \
    'to the role yourself, or delete the role yourself and try again.'
  return 'Error'
end
```

```
# Checks whether the specified AWS CloudWatch Events rule exists among
# those provided to this function.
# This is a helper function that is called by the rule_exists? function.
#
# @param rules [Array] An array of Aws::CloudWatchEvents::Types::Rule objects.
# @param rule_arn [String] The name of the rule to find.
# @return [Boolean] true if the name of the rule was found; otherwise, false.
# @example
#   cloudwatchevents_client = Aws::CloudWatch::Client.new(region: 'us-east-1')
#   response = cloudwatchevents_client.list_rules
#   if rule_found?(response.rules, 'aws-doc-sdk-examples-ec2-state-change')
#     puts 'Rule found.'
#   end
def rule_found?(rules, rule_name)
  rules.each do |rule|
    return true if rule.name == rule_name
  end
  return false
end

# Checks whether the specified rule exists among those available to the
# caller in AWS CloudWatch Events.
#
# @param cloudwatchevents_client [Aws::CloudWatchEvents::Client]
#   An initialized AWS CloudWatch Events client.
# @param rule_name [String] The name of the rule to find.
# @return [Boolean] true if the rule name was found; otherwise, false.
# @example
#   exit 1 unless rule_exists?(
#     Aws::CloudWatch::Client.new(region: 'us-east-1')
#     'aws-doc-sdk-examples-ec2-state-change'
#   )
def rule_exists?(cloudwatchevents_client, rule_name)
  puts "Searching for rule with name '#{rule_name}'..."
  response = cloudwatchevents_client.list_rules
  if response.rules.count.positive?
    if rule_found?(response.rules, rule_name)
      puts 'Rule found.'
      return true
    end
  end
  while response.next_page? do
    response = response.next_page
    if response.rules.count.positive?
```

```
        if rule_found?(response.rules, rule_name)
          puts 'Rule found.'
          return true
        end
      end
    end
  end
  puts 'Rule not found.'
  return false
rescue StandardError => e
  puts "Rule not found: #{e.message}"
  return false
end

# Creates a rule in AWS CloudWatch Events.
# This rule is triggered whenever an available instance in
# Amazon Elastic Compute Cloud (Amazon EC2) changes to the specified state.
# This rule is designed to be used specifically by this code example.
#
# Prerequisites:
#
# - A role in AWS Identity and Access Management (IAM) that is designed
#   to be used specifically by this code example.
# - A topic in Amazon Simple Notification Service (Amazon SNS).
#
# @param cloudwatchevents_client [Aws::CloudWatchEvents::Client]
#   An initialized AWS CloudWatch Events client.
# @param rule_name [String] The name of the rule to create.
# @param rule_description [String] Some description for this rule.
# @param instance_state [String] The state that available instances in
#   Amazon Elastic Compute Cloud (Amazon EC2) must change to, to
#   trigger this rule.
# @param role_arn [String] The Amazon Resource Name (ARN) of the IAM role.
# @param target_id [String] Some identifying string for the rule's target.
# @param topic_arn [String] The ARN of the Amazon SNS topic.
# @return [Boolean] true if the rule was created; otherwise, false.
# @example
#   exit 1 unless rule_created?(
#     Aws::CloudWatch::Client.new(region: 'us-east-1'),
#     'aws-doc-sdk-examples-ec2-state-change',
#     'Triggers when any available EC2 instance starts.',
#     'running',
#     'arn:aws:iam::111111111111:role/aws-doc-sdk-examples-ec2-state-change',
#     'sns-topic',
```

```
# 'arn:aws:sns:us-east-1:111111111111:aws-doc-sdk-examples-topic'
# )
def rule_created?(
  cloudwatchevents_client,
  rule_name,
  rule_description,
  instance_state,
  role_arn,
  target_id,
  topic_arn
)
  puts "Creating rule with name '#{rule_name}'..."
  put_rule_response = cloudwatchevents_client.put_rule(
    name: rule_name,
    description: rule_description,
    event_pattern: {
      'source': [
        'aws.ec2'
      ],
      'detail-type': [
        'EC2 Instance State-change Notification'
      ],
      'detail': {
        'state': [
          instance_state
        ]
      }
    }.to_json,
    state: 'ENABLED',
    role_arn: role_arn
  )
  puts "Rule created with ARN '#{put_rule_response.rule_arn}'."

  put_targets_response = cloudwatchevents_client.put_targets(
    rule: rule_name,
    targets: [
      {
        id: target_id,
        arn: topic_arn
      }
    ]
  )
  if put_targets_response.key?(:failed_entry_count) &&
    put_targets_response.failed_entry_count > 0
```



```

    puts 'Error(s) adding target to rule:'
    put_targets_response.failed_entries.each do |failure|
      puts failure.error_message
    end
    return false
  else
    return true
  end
rescue StandardError => e
  puts "Error creating rule or adding target to rule: #{e.message}"
  puts 'If the rule was created, you must add the target ' \
    'to the rule yourself, or delete the rule yourself and try again.'
  return false
end

# Checks to see whether the specified log group exists among those available
# to the caller in Amazon CloudWatch Logs.
#
# @param cloudwatchlogs_client [Aws::CloudWatchLogs::Client] An initialized
#   Amazon CloudWatch Logs client.
# @param log_group_name [String] The name of the log group to find.
# @return [Boolean] true if the log group name was found; otherwise, false.
# @example
#   exit 1 unless log_group_exists?(
#     Aws::CloudWatchLogs::Client.new(region: 'us-east-1'),
#     'aws-doc-sdk-examples-cloudwatch-log'
#   )
def log_group_exists?(cloudwatchlogs_client, log_group_name)
  puts "Searching for log group with name '#{log_group_name}'..."
  response = cloudwatchlogs_client.describe_log_groups(
    log_group_name_prefix: log_group_name
  )
  if response.log_groups.count.positive?
    response.log_groups.each do |log_group|
      if log_group.log_group_name == log_group_name
        puts 'Log group found.'
        return true
      end
    end
  end
  puts 'Log group not found.'
  return false
rescue StandardError => e
  puts "Log group not found: #{e.message}"

```

```
    return false
  end

  # Creates a log group in Amazon CloudWatch Logs.
  #
  # @param cloudwatchlogs_client [Aws::CloudWatchLogs::Client] An initialized
  #   Amazon CloudWatch Logs client.
  # @param log_group_name [String] The name of the log group to create.
  # @return [Boolean] true if the log group name was created; otherwise, false.
  # @example
  #   exit 1 unless log_group_created?(
  #     Aws::CloudWatchLogs::Client.new(region: 'us-east-1'),
  #     'aws-doc-sdk-examples-cloudwatch-log'
  #   )
  def log_group_created?(cloudwatchlogs_client, log_group_name)
    puts "Attempting to create log group with the name '#{log_group_name}'..."
    cloudwatchlogs_client.create_log_group(log_group_name: log_group_name)
    puts 'Log group created.'
    return true
  rescue StandardError => e
    puts "Error creating log group: #{e.message}"
    return false
  end

  # Writes an event to a log stream in Amazon CloudWatch Logs.
  #
  # Prerequisites:
  #
  # - A log group in Amazon CloudWatch Logs.
  # - A log stream within the log group.
  #
  # @param cloudwatchlogs_client [Aws::CloudWatchLogs::Client] An initialized
  #   Amazon CloudWatch Logs client.
  # @param log_group_name [String] The name of the log group.
  # @param log_stream_name [String] The name of the log stream within
  #   the log group.
  # @param message [String] The message to write to the log stream.
  # @param sequence_token [String] If available, the sequence token from the
  #   message that was written immediately before this message. This sequence
  #   token is returned by Amazon CloudWatch Logs whenever you programmatically
  #   write a message to the log stream.
  # @return [String] The sequence token that is returned by
  #   Amazon CloudWatch Logs after successfully writing the message to the
  #   log stream.
```

```
# @example
# puts log_event(
#   Aws::EC2::Client.new(region: 'us-east-1'),
#   'aws-doc-sdk-examples-cloudwatch-log'
#   '2020/11/19/53f985be-199f-408e-9a45-fc242df41fEX',
#   "Instance 'i-033c48ef067af3dEX' restarted.",
#   '495426724868310740095796045676567882148068632824696073EX'
# )
def log_event(
  cloudwatchlogs_client,
  log_group_name,
  log_stream_name,
  message,
  sequence_token
)
  puts "Attempting to log '#{message}' to log stream '#{log_stream_name}'..."
  event = {
    log_group_name: log_group_name,
    log_stream_name: log_stream_name,
    log_events: [
      {
        timestamp: (Time.now.utc.to_f.round(3) * 1_000).to_i,
        message: message
      }
    ]
  }
  unless sequence_token.empty?
    event[:sequence_token] = sequence_token
  end

  response = cloudwatchlogs_client.put_log_events(event)
  puts 'Message logged.'
  return response.next_sequence_token
rescue StandardError => e
  puts "Message not logged: #{e.message}"
end

# Restarts an Amazon Elastic Compute Cloud (Amazon EC2) instance
# and adds information about the related activity to a log stream
# in Amazon CloudWatch Logs.
#
# Prerequisites:
#
# - The Amazon EC2 instance to restart.
```

```

# - The log group in Amazon CloudWatch Logs to add related activity
#   information to.
#
# @param ec2_client [Aws::EC2::Client] An initialized Amazon EC2 client.
# @param cloudwatchlogs_client [Aws::CloudWatchLogs::Client]
#   An initialized Amazon CloudWatch Logs client.
# @param instance_id [String] The ID of the instance.
# @param log_group_name [String] The name of the log group.
# @return [Boolean] true if the instance was restarted and the information
#   was written to the log stream; otherwise, false.
# @example
#   exit 1 unless instance_restarted?(
#     Aws::EC2::Client.new(region: 'us-east-1'),
#     Aws::CloudWatchLogs::Client.new(region: 'us-east-1'),
#     'i-033c48ef067af3dEX',
#     'aws-doc-sdk-examples-cloudwatch-log'
#   )
def instance_restarted?(
  ec2_client,
  cloudwatchlogs_client,
  instance_id,
  log_group_name
)
  log_stream_name = "#{Time.now.year}/#{Time.now.month}/#{Time.now.day}/" \
    "#{SecureRandom.uuid}"
  cloudwatchlogs_client.create_log_stream(
    log_group_name: log_group_name,
    log_stream_name: log_stream_name
  )
  sequence_token = ''

  puts "Attempting to stop the instance with the ID '#{instance_id}'. " \
    'This might take a few minutes...'
  ec2_client.stop_instances(instance_ids: [instance_id])
  ec2_client.wait_until(:instance_stopped, instance_ids: [instance_id])
  puts 'Instance stopped.'
  sequence_token = log_event(
    cloudwatchlogs_client,
    log_group_name,
    log_stream_name,
    "Instance '#{instance_id}' stopped.",
    sequence_token
  )
)

```

```

puts 'Attempting to restart the instance. This might take a few minutes...'
ec2_client.start_instances(instance_ids: [instance_id])
ec2_client.wait_until(:instance_running, instance_ids: [instance_id])
puts 'Instance restarted.'
sequence_token = log_event(
  cloudwatchlogs_client,
  log_group_name,
  log_stream_name,
  "Instance '#{instance_id}' restarted.",
  sequence_token
)

return true
rescue StandardError => e
puts 'Error creating log stream or stopping or restarting the instance: ' \
  "#{e.message}"
log_event(
  cloudwatchlogs_client,
  log_group_name,
  log_stream_name,
  "Error stopping or starting instance '#{instance_id}': #{e.message}",
  sequence_token
)
return false
end

# Displays information about activity for a rule in Amazon CloudWatch Events.
#
# Prerequisites:
#
# - A rule in Amazon CloudWatch Events.
#
# @param cloudwatch_client [Amazon::CloudWatch::Client] An initialized
#   Amazon CloudWatch client.
# @param rule_name [String] The name of the rule.
# @param start_time [Time] The timestamp that determines the first datapoint
#   to return. Can also be expressed as DateTime, Date, Integer, or String.
# @param end_time [Time] The timestamp that determines the last datapoint
#   to return. Can also be expressed as DateTime, Date, Integer, or String.
# @param period [Integer] The interval, in seconds, to check for activity.
# @example
#   display_rule_activity(
#     Aws::CloudWatch::Client.new(region: 'us-east-1'),
#     'aws-doc-sdk-examples-ec2-state-change',

```

```
# Time.now - 600, # Start checking from 10 minutes ago.
# Time.now, # Check up until now.
# 60 # Check every minute during those 10 minutes.
# )
def display_rule_activity(
  cloudwatch_client,
  rule_name,
  start_time,
  end_time,
  period
)
  puts 'Attempting to display rule activity...'
  response = cloudwatch_client.get_metric_statistics(
    namespace: 'AWS/Events',
    metric_name: 'Invocations',
    dimensions: [
      {
        name: 'RuleName',
        value: rule_name
      }
    ],
    start_time: start_time,
    end_time: end_time,
    period: period,
    statistics: ['Sum'],
    unit: 'Count'
  )

  if response.key?(:datapoints) && response.datapoints.count.positive?
    puts "The event rule '#{rule_name}' was triggered:"
    response.datapoints.each do |datapoint|
      puts "  #{datapoint.sum} time(s) at #{datapoint.timestamp}"
    end
  else
    puts "The event rule '#{rule_name}' was not triggered during the " \
      'specified time period.'
  end
rescue StandardError => e
  puts "Error getting information about event rule activity: #{e.message}"
end

# Displays log information for all of the log streams in a log group in
# Amazon CloudWatch Logs.
#
```

```
# Prerequisites:
#
# - A log group in Amazon CloudWatch Logs.
#
# @param cloudwatchlogs_client [Amazon::CloudWatchLogs::Client] An initialized
#   Amazon CloudWatch Logs client.
# @param log_group_name [String] The name of the log group.
# @example
#   display_log_data(
#     Amazon::CloudWatchLogs::Client.new(region: 'us-east-1'),
#     'aws-doc-sdk-examples-cloudwatch-log'
#   )
def display_log_data(cloudwatchlogs_client, log_group_name)
  puts 'Attempting to display log stream data for the log group ' \
    "named '#{log_group_name}'..."
  describe_log_streams_response = cloudwatchlogs_client.describe_log_streams(
    log_group_name: log_group_name,
    order_by: 'LastEventTime',
    descending: true
  )
  if describe_log_streams_response.key?(:log_streams) &&
    describe_log_streams_response.log_streams.count.positive?
    describe_log_streams_response.log_streams.each do |log_stream|
      get_log_events_response = cloudwatchlogs_client.get_log_events(
        log_group_name: log_group_name,
        log_stream_name: log_stream.log_stream_name
      )
      puts "\nLog messages for '#{log_stream.log_stream_name}':"
      puts '-' * (log_stream.log_stream_name.length + 20)
      if get_log_events_response.key?(:events) &&
        get_log_events_response.events.count.positive?
        get_log_events_response.events.each do |event|
          puts event.message
        end
      else
        puts 'No log messages for this log stream.'
      end
    end
  end
end
rescue StandardError => e
  puts 'Error getting information about the log streams or their messages: ' \
    "#{e.message}"
end
```

```
# Displays a reminder to the caller to manually clean up any associated
# AWS resources that they no longer need.
#
# @param topic_name [String] The name of the Amazon SNS topic.
# @param role_name [String] The name of the IAM role.
# @param rule_name [String] The name of the Amazon CloudWatch Events rule.
# @param log_group_name [String] The name of the Amazon CloudWatch Logs log group.
# @param instance_id [String] The ID of the Amazon EC2 instance.
# @example
#   manual_cleanup_notice(
#     'aws-doc-sdk-examples-topic',
#     'aws-doc-sdk-examples-cloudwatch-events-rule-role',
#     'aws-doc-sdk-examples-ec2-state-change',
#     'aws-doc-sdk-examples-cloudwatch-log',
#     'i-033c48ef067af3dEX'
#   )
def manual_cleanup_notice(
  topic_name, role_name, rule_name, log_group_name, instance_id
)
  puts '-' * 10
  puts 'Some of the following AWS resources might still exist in your account.'
  puts 'If you no longer want to use this code example, then to clean up'
  puts 'your AWS account and avoid unexpected costs, you might want to'
  puts 'manually delete any of the following resources if they exist:'
  puts "- The Amazon SNS topic named '#{topic_name}'."
  puts "- The IAM role named '#{role_name}'."
  puts "- The Amazon CloudWatch Events rule named '#{rule_name}'."
  puts "- The Amazon CloudWatch Logs log group named '#{log_group_name}'."
  puts "- The Amazon EC2 instance with the ID '#{instance_id}'."
end

# Full example call:
def run_me
  # Properties for the Amazon SNS topic.
  topic_name = 'aws-doc-sdk-examples-topic'
  email_address = 'mary@example.com'
  # Properties for the IAM role.
  role_name = 'aws-doc-sdk-examples-cloudwatch-events-rule-role'
  # Properties for the Amazon CloudWatch Events rule.
  rule_name = 'aws-doc-sdk-examples-ec2-state-change'
  rule_description = 'Triggers when any available EC2 instance starts.'
  instance_state = 'running'
  target_id = 'sns-topic'
  # Properties for the Amazon EC2 instance.
```



```
instance_id = 'i-033c48ef067af3dEX'
# Properties for displaying the event rule's activity.
start_time = Time.now - 600 # Go back over the past 10 minutes
                        # (10 minutes * 60 seconds = 600 seconds).

end_time = Time.now
period = 60 # Look back every 60 seconds over the past 10 minutes.
# Properties for the Amazon CloudWatch Logs log group.
log_group_name = 'aws-doc-sdk-examples-cloudwatch-log'
# AWS service clients for this code example.
region = 'us-east-1'
sts_client = Aws::STS::Client.new(region: region)
sns_client = Aws::SNS::Client.new(region: region)
iam_client = Aws::IAM::Client.new(region: region)
cloudwatchevents_client = Aws::CloudWatchEvents::Client.new(region: region)
ec2_client = Aws::EC2::Client.new(region: region)
cloudwatch_client = Aws::CloudWatch::Client.new(region: region)
cloudwatchlogs_client = Aws::CloudWatchLogs::Client.new(region: region)

# Get the caller's account ID for use in forming
# Amazon Resource Names (ARNs) that this code relies on later.
account_id = sts_client.get_caller_identity.account

# If the Amazon SNS topic doesn't exist, create it.
topic_arn = "arn:aws:sns:#{region}:#{account_id}:#{topic_name}"
unless topic_exists?(sns_client, topic_arn)
  topic_arn = create_topic(sns_client, topic_name, email_address)
  if topic_arn == 'Error'
    puts 'Could not create the Amazon SNS topic correctly. Program stopped.'
    manual_cleanup_notice(
      topic_name, role_name, rule_name, log_group_name, instance_id
    )
    exit 1
  end
end

# If the IAM role doesn't exist, create it.
role_arn = "arn:aws:iam:#{account_id}:role/#{role_name}"
unless role_exists?(iam_client, role_arn)
  role_arn = create_role(iam_client, role_name)
  if role_arn == 'Error'
    puts 'Could not create the IAM role correctly. Program stopped.'
    manual_cleanup_notice(
      topic_name, role_name, rule_name, log_group_name, instance_id
    )
  end
end
```

```
end
end

# If the Amazon CloudWatch Events rule doesn't exist, create it.
unless rule_exists?(cloudwatchevents_client, rule_name)
  unless rule_created?(
    cloudwatchevents_client,
    rule_name,
    rule_description,
    instance_state,
    role_arn,
    target_id,
    topic_arn
  )
    puts 'Could not create the Amazon CloudWatch Events rule correctly. ' \
        'Program stopped.'
    manual_cleanup_notice(
      topic_name, role_name, rule_name, log_group_name, instance_id
    )
  end
end

# If the Amazon CloudWatch Logs log group doesn't exist, create it.
unless log_group_exists?(cloudwatchlogs_client, log_group_name)
  unless log_group_created?(cloudwatchlogs_client, log_group_name)
    puts 'Could not create the Amazon CloudWatch Logs log group ' \
        'correctly. Program stopped.'
    manual_cleanup_notice(
      topic_name, role_name, rule_name, log_group_name, instance_id
    )
  end
end

# Restart the Amazon EC2 instance, which triggers the rule.
unless instance_restarted?(
  ec2_client,
  cloudwatchlogs_client,
  instance_id,
  log_group_name
)
  puts 'Could not restart the instance to trigger the rule. ' \
      'Continuing anyway to show information about the rule and logs...'
end
```

```
# Display how many times the rule was triggered over the past 10 minutes.
display_rule_activity(
  cloudwatch_client,
  rule_name,
  start_time,
  end_time,
  period
)

# Display related log data in Amazon CloudWatch Logs.
display_log_data(cloudwatchlogs_client, log_group_name)

# Reminder the caller to clean up any AWS resources that are used
# by this code example and are no longer needed.
manual_cleanup_notice(
  topic_name, role_name, rule_name, log_group_name, instance_id
)
end

run_me if $PROGRAM_NAME == __FILE__
```

CodeBuild 使用 AWS SDK for Ruby 範例

CodeBuild 是完全受控的建置服務，可編譯原始程式碼、執行測試，以及產生可供部署的軟體套件。您可以使用下面的 AWS SDK 來訪問 Ruby 代碼示例AWS CodeBuild。如需相關資訊 CodeBuild，請參閱[AWS CodeBuild文件](#)。

主題

- [取得所有AWS CodeBuild專案的相關資訊](#)
- [建立AWS CodeBuild專案](#)
- [列出AWS CodeBuild項目構建](#)

取得所有AWS CodeBuild專案的相關資訊

下列範例會列出最多 100 個AWS CodeBuild專案的名稱。

```
# Copyright 2010-2019 Amazon.com, Inc. or its affiliates. All Rights Reserved.
#
# This file is licensed under the Apache License, Version 2.0 (the "License").
# You may not use this file except in compliance with the License. A copy of the
```

```
# License is located at
#
# http://aws.amazon.com/apache2.0/
#
# This file is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS
# OF ANY KIND, either express or implied. See the License for the specific
# language governing permissions and limitations under the License.

require 'aws-sdk-codebuild' # v2: require 'aws-sdk'

client = Aws::CodeBuild::Client.new(region: 'us-west-2')

resp = client.list_projects({
  sort_by: 'NAME', # accepts NAME, CREATED_TIME, LAST_MODIFIED_TIME
  sort_order: 'ASCENDING' # accepts ASCENDING, DESCENDING
})

resp.projects.each { |p| puts p }

puts
```

選擇Copy在本機儲存程式碼。請參閱 (詳見) 的[完整實例](#) GitHub。

建立AWS CodeBuild專案

下列範例會建置在命令列上指定的AWS CodeBuild專案。如果沒有提供命令行參數，它會發出錯誤並退出。

```
# Copyright 2010-2019 Amazon.com, Inc. or its affiliates. All Rights Reserved.
#
# This file is licensed under the Apache License, Version 2.0 (the "License").
# You may not use this file except in compliance with the License. A copy of the
# License is located at
#
# http://aws.amazon.com/apache2.0/
#
# This file is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS
# OF ANY KIND, either express or implied. See the License for the specific
# language governing permissions and limitations under the License.

require 'aws-sdk-codebuild' # v2: require 'aws-sdk'

project_name = ''
```

```
if ARGV.length != 1
  puts 'You must supply the name of the project to build'
  exit 1
else
  project_name = ARGV[0]
end

client = Aws::CodeBuild::Client.new(region: 'us-west-2')

begin
  client.start_build(project_name: project_name)
  puts 'Building project ' + project_name
rescue StandardError => ex
  puts 'Error building project: ' + ex.message
end
```

選擇Copy在本機儲存程式碼。請參閱 (詳見) 的[完整實例](#) GitHub。

列出AWS CodeBuild項目構建

下列範例會顯示AWS CodeBuild專案組建的相關資訊。這項資訊包括專案名稱、建置開始的時間，以及每個建置階段所花費的時間 (以秒為單位)。

```
# Copyright 2010-2019 Amazon.com, Inc. or its affiliates. All Rights Reserved.
#
# This file is licensed under the Apache License, Version 2.0 (the "License").
# You may not use this file except in compliance with the License. A copy of the
# License is located at
#
# http://aws.amazon.com/apache2.0/
#
# This file is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS
# OF ANY KIND, either express or implied. See the License for the specific
# language governing permissions and limitations under the License.

require 'aws-sdk-codebuild' # v2: require 'aws-sdk'

client = Aws::CodeBuild::Client.new(region: 'us-west-2')

build_list = client.list_builds({sort_order: 'ASCENDING', })

builds = client.batch_get_builds({ids: build_list.ids})
```

```
builds.builds.each do |build|
  puts 'Project:      ' + build.project_name
  puts 'Phase:       ' + build.current_phase
  puts 'Status:      ' + build.build_status
end
```

選擇Copy在本機儲存程式碼。請參閱 (詳見) 的[完整實例](#) GitHub。

Amazon EC2 示例使用AWS開發套件的紅寶石

Amazon Elastic Compute Cloud (Amazon EC2) 是一種網路服務，提供可調整大小的運算容量 (實際上是 Amazon 資料中心中的伺服器)，可用來建立和託管軟體系統。您可以使用下列範例，使用 AWS SDK for Ruby 存取 Amazon EC2。如需有關 Amazon EC2 的詳細資訊，請參閱 [Amazon EC2 文件](#)。

主題

- [創建一個 Amazon EC2 VPC](#)
- [建立 Internet Gateway 並將其連接至 Amazon EC2 中的 VPC](#)
- [為 Amazon EC2 創建公共子網](#)
- [建立 Amazon EC2 路由表並將其與子網路產生關聯](#)
- [在 Amazon EC2 中使用彈性 IP 地址](#)
- [創建一個 Amazon EC2 安全組](#)
- [使用 Amazon EC2 安全群組](#)
- [在 Amazon EC2 中使用密鑰對](#)
- [取得所有 Amazon EC2 執行個體的相關資訊](#)
- [取得具有特定標籤值之所有 Amazon EC2 執行個體的相關資訊](#)
- [取得特定 Amazon EC2 執行個體的相關資訊](#)
- [創建一個 Amazon EC2 實例](#)
- [停止亞 Amazon EC2 實例](#)
- [啟動一個 Amazon EC2 實例](#)
- [重新啟動 Amazon EC2 實例](#)
- [管理 Amazon EC2 實例](#)
- [終止亞馬遜 EC2 實例](#)
- [取得 Amazon EC2 區域和可用區域的相關資訊](#)

創建一個 Amazon EC2 VPC

以下程式碼範例會在 Amazon Virtual Private Cloud (Amazon VPC) 中建立虛擬私有雲 (VPC)，然後為 VPC 加上標籤。

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX - License - Identifier: Apache - 2.0

require 'aws-sdk-ec2'

# Creates a virtual private cloud (VPC) in
# Amazon Virtual Private Cloud (Amazon VPC) and then tags
# the VPC.
#
# @param ec2_resource [Aws::EC2::Resource] An initialized
#   Amazon Elastic Compute Cloud (Amazon EC2) resource object.
# @param cidr_block [String] The IPv4 CIDR block for the subnet.
# @param tag_key [String] The key portion of the tag for the VPC.
# @param tag_value [String] The value portion of the tag for the VPC.
# @return [Boolean] true if the VPC was created and tagged;
#   otherwise, false.
# @example
#   exit 1 unless vpc_created_and_tagged?(
#     Aws::EC2::Resource.new(region: 'us-east-1'),
#     '10.0.0.0/24',
#     'my-key',
#     'my-value'
#   )
def vpc_created_and_tagged?(
  ec2_resource,
  cidr_block,
  tag_key,
  tag_value
)
  vpc = ec2_resource.create_vpc(cidr_block: cidr_block)

  # Create a public DNS by enabling DNS support and DNS hostnames.
  vpc.modify_attribute(enable_dns_support: { value: true })
  vpc.modify_attribute(enable_dns_hostnames: { value: true })

  vpc.create_tags(tags: [{ key: tag_key, value: tag_value }])

  puts "Created VPC with ID '#{vpc.id}' and tagged with key " \
```

```
    "#{tag_key}' and value '#{tag_value}'."
  return true
rescue StandardError => e
  puts "#{e.message}"
  return false
end

# Full example call:
def run_me
  cidr_block = ''
  tag_key = ''
  tag_value = ''
  region = ''
  # Print usage information and then stop.
  if ARGV[0] == '--help' || ARGV[0] == '-h'
    puts 'Usage:  ruby ec2-ruby-example-create-vpc.rb ' \
      'CIDR_BLOCK TAG_KEY TAG_VALUE REGION'
    puts 'Example: ruby ec2-ruby-example-create-vpc.rb ' \
      '10.0.0.0/24 my-key my-value us-east-1'
    exit 1
  # If no values are specified at the command prompt, use these default values.
  elsif ARGV.count.zero?
    cidr_block = '10.0.0.0/24'
    tag_key = 'my-key'
    tag_value = 'my-value'
    region = 'us-east-1'
  # Otherwise, use the values as specified at the command prompt.
  else
    cidr_block = ARGV[0]
    tag_key = ARGV[1]
    tag_value = ARGV[2]
    region = ARGV[3]
  end

  ec2_resource = Aws::EC2::Resource.new(region: region)

  if vpc_created_and_tagged?(
    ec2_resource,
    cidr_block,
    tag_key,
    tag_value
  )
    puts 'VPC created and tagged.'
  else
```



```
    puts 'VPC not created or not tagged.'
  end
end

run_me if $PROGRAM_NAME == __FILE__
```

建立 Internet Gateway 並將其連接至 Amazon EC2 中的 VPC

下列程式碼範例會建立網際網路閘道，然後將其附加至 Amazon 虛擬私有雲 (Amazon VPC) 中的虛擬私有雲 (VPC)。

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX - License - Identifier: Apache - 2.0

require 'aws-sdk-ec2'

# Creates an internet gateway and then attaches it to a virtual private cloud
# (VPC) in Amazon Virtual Private Cloud (Amazon VPC).
#
# Prerequisites:
#
# - A VPC in Amazon VPC.
#
# @param ec2_resource [Aws::EC2::Resource] An initialized
#   Amazon Elastic Compute Cloud (Amazon EC2) resource object.
# @param vpc_id [String] The ID of the VPC to attach the internet gateway.
# @param tag_key [String] The key of the tag to attach to the internet gateway.
# @param tag_value [String] The value of the tag to attach to the
#   internet gateway.
# @return [Boolean] true if the internet gateway was created and attached;
#   otherwise, false.
# @example
#   exit 1 unless internet_gateway_created_and_attached?(
#     Aws::EC2::Resource.new(region: 'us-east-1'),
#     'vpc-6713dfEX'
#   )
def internet_gateway_created_and_attached?(
  ec2_resource,
  vpc_id,
  tag_key,
  tag_value
)
  igw = ec2_resource.create_internet_gateway
```

```
puts "The internet gateway's ID is '#{igw.id}'."
igw.attach_to_vpc(vpc_id: vpc_id)
igw.create_tags(
  tags: [
    {
      key: tag_key,
      value: tag_value
    }
  ]
)
return true
rescue StandardError => e
  puts "Error creating or attaching internet gateway: #{e.message}"
  puts 'If the internet gateway was created but not attached, you should ' \
    'clean up by deleting the internet gateway.'
  return false
end

# Full example call:
def run_me
  vpc_id = ''
  tag_key = ''
  tag_value = ''
  region = ''
  # Print usage information and then stop.
  if ARGV[0] == '--help' || ARGV[0] == '-h'
    puts 'Usage: ruby ec2-ruby-example-attach-igw-vpc.rb ' \
      'VPC_ID TAG_KEY TAG_VALUE REGION'
    puts 'Example: ruby ec2-ruby-example-attach-igw-vpc.rb ' \
      'vpc-6713dfEX my-key my-value us-east-1'
    exit 1
  # If no values are specified at the command prompt, use these default values.
  elsif ARGV.count.zero?
    vpc_id = 'vpc-6713dfEX'
    tag_key = 'my-key'
    tag_value = 'my-value'
    region = 'us-east-1'
  # Otherwise, use the values as specified at the command prompt.
  else
    vpc_id = ARGV[0]
    tag_key = ARGV[1]
    tag_value = ARGV[2]
    region = ARGV[3]
  end
end
```

```
ec2_resource = Aws::EC2::Resource.new(region: region)

if internet_gateway_created_and_attached?(
  ec2_resource,
  vpc_id,
  tag_key,
  tag_value
)
  puts "Created and attached internet gateway to VPC '#{vpc_id}'."
else
  puts "Could not create or attach internet gateway to VPC '#{vpc_id}'."
end
end

run_me if $PROGRAM_NAME == __FILE__
```

為 Amazon EC2 創建公共子網

下列程式碼範例會在 Amazon 虛擬私有雲端 (Amazon VPC) 的虛擬私有雲端 (VPC) 內建立子網路，然後標記子網路。

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX - License - Identifier: Apache - 2.0

require 'aws-sdk-ec2'

# Creates a subnet within a virtual private cloud (VPC) in
# Amazon Virtual Private Cloud (Amazon VPC) and then tags
# the subnet.
#
# Prerequisites:
#
# - A VPC in Amazon VPC.
#
# @param ec2_resource [Aws::EC2::Resource] An initialized
#   Amazon Elastic Compute Cloud (Amazon EC2) resource object.
# @param vpc_id [String] The ID of the VPC for the subnet.
# @param cidr_block [String] The IPv4 CIDR block for the subnet.
# @param availability_zone [String] The ID of the Availability Zone
#   for the subnet.
# @param tag_key [String] The key portion of the tag for the subnet.
# @param tag_value [String] The value portion of the tag for the subnet.
```

```
# @return [Boolean] true if the subnet was created and tagged;
# otherwise, false.
# @example
# exit 1 unless subnet_created_and_tagged?(
#   Aws::EC2::Resource.new(region: 'us-east-1'),
#   'vpc-6713dfEX',
#   '10.0.0.0/24',
#   'us-east-1a',
#   'my-key',
#   'my-value'
# )
def subnet_created_and_tagged?(
  ec2_resource,
  vpc_id,
  cidr_block,
  availability_zone,
  tag_key,
  tag_value
)
  subnet = ec2_resource.create_subnet(
    vpc_id: vpc_id,
    cidr_block: cidr_block,
    availability_zone: availability_zone
  )
  subnet.create_tags(
    tags: [
      {
        key: tag_key,
        value: tag_value
      }
    ]
  )
  puts "Subnet created with ID '#{subnet.id}' in VPC with ID '#{vpc_id}' " \
    "and CIDR block '#{cidr_block}' in availability zone " \
    "'#{availability_zone}' and tagged with key '#{tag_key}' and " \
    "value '#{tag_value}'."
  return true
rescue StandardError => e
  puts "Error creating or tagging subnet: #{e.message}"
  return false
end

# Full example call:
def run_me
```

```
vpc_id = ''
cidr_block = ''
availability_zone = ''
tag_key = ''
tag_value = ''
region = ''
# Print usage information and then stop.
if ARGV[0] == '--help' || ARGV[0] == '-h'
  puts 'Usage: ruby ec2-ruby-example-create-subnet.rb ' \
    'VPC_ID CIDR_BLOCK AVAILABILITY_ZONE TAG_KEY TAG_VALUE REGION'
  puts 'Example: ruby ec2-ruby-example-create-subnet.rb ' \
    'vpc-6713dfEX 10.0.0.0/24 us-east-1a my-key my-value us-east-1'
  exit 1
# If no values are specified at the command prompt, use these default values.
elsif ARGV.count.zero?
  vpc_id = 'vpc-6713dfEX'
  cidr_block = '10.0.0.0/24'
  availability_zone = 'us-east-1a'
  tag_key = 'my-key'
  tag_value = 'my-value'
  region = 'us-east-1'
# Otherwise, use the values as specified at the command prompt.
else
  vpc_id = ARGV[0]
  cidr_block = ARGV[1]
  availability_zone = ARGV[2]
  tag_key = ARGV[3]
  tag_value = ARGV[4]
  region = ARGV[5]
end

ec2_resource = Aws::EC2::Resource.new(region: region)

if subnet_created_and_tagged?(
  ec2_resource,
  vpc_id,
  cidr_block,
  availability_zone,
  tag_key,
  tag_value
)
  puts 'Subnet created and tagged.'
else
  puts 'Subnet not created or not tagged.'
```

```
end
end

run_me if $PROGRAM_NAME == __FILE__
```

建立 Amazon EC2 路由表並將其與子網路產生關聯

下列程式碼範例會在 Amazon 虛擬私有雲端 (Amazon VPC) 中建立路由表，然後將路由表與 Amazon VPC 中的子網路產生關聯。

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX - License - Identifier: Apache - 2.0

require 'aws-sdk-ec2'

# Creates a route table in Amazon Virtual Private Cloud (Amazon VPC)
# and then associates the route table with a subnet in Amazon VPC.
#
# Prerequisites:
#
# - A VPC in Amazon VPC.
# - A subnet in that VPC.
# - A gateway attached to that subnet.
#
# @param ec2_resource [Aws::EC2::Resource] An initialized
#   Amazon Elastic Compute Cloud (Amazon EC2) resource object.
# @param vpc_id [String] The ID of the VPC for the route table.
# @param subnet_id [String] The ID of the subnet for the route table.
# @param gateway_id [String] The ID of the gateway for the route.
# @param destination_cidr_block [String] The destination CIDR block
#   for the route.
# @param tag_key [String] The key portion of the tag for the route table.
# @param tag_value [String] The value portion of the tag for the route table.
# @return [Boolean] true if the route table was created and associated;
#   otherwise, false.
# @example
#   exit 1 unless route_table_created_and_associated?(
#     Aws::EC2::Resource.new(region: 'us-east-1'),
#     'vpc-0b6f769731EXAMPLE',
#     'subnet-03d9303b57EXAMPLE',
#     'igw-06ca90c011EXAMPLE',
#     '0.0.0.0/0',
#     'my-key',
```

```
#   'my-value'
#   )
def route_table_created_and_associated?(
  ec2_resource,
  vpc_id,
  subnet_id,
  gateway_id,
  destination_cidr_block,
  tag_key,
  tag_value
)
  route_table = ec2_resource.create_route_table(vpc_id: vpc_id)
  puts "Created route table with ID '#{route_table.id}'."
  route_table.create_tags(
    tags: [
      {
        key: tag_key,
        value: tag_value
      }
    ]
  )
  puts 'Added tags to route table.'
  route_table.create_route(
    destination_cidr_block: destination_cidr_block,
    gateway_id: gateway_id
  )
  puts 'Created route with destination CIDR block ' \
    "'#{destination_cidr_block}' and associated with gateway " \
    "with ID '#{gateway_id}'."
  route_table.associate_with_subnet(subnet_id: subnet_id)
  puts "Associated route table with subnet with ID '#{subnet_id}'."
  return true
rescue StandardError => e
  puts "Error creating or associating route table: #{e.message}"
  puts 'If the route table was created but not associated, you should ' \
    'clean up by deleting the route table.'
  return false
end

# Full example call:
def run_me
  vpc_id = ''
  subnet_id = ''
  gateway_id = ''
```

```
destination_cidr_block = ''
tag_key = ''
tag_value = ''
region = ''
# Print usage information and then stop.
if ARGV[0] == '--help' || ARGV[0] == '-h'
  puts 'Usage: ruby ec2-ruby-example-create-route-table.rb ' \
    'VPC_ID SUBNET_ID GATEWAY_ID DESTINATION_CIDR_BLOCK ' \
    'TAG_KEY TAG_VALUE REGION'
  puts 'Example: ruby ec2-ruby-example-create-route-table.rb ' \
    'vpc-0b6f769731EXAMPLE subnet-03d9303b57EXAMPLE igw-06ca90c011EXAMPLE ' \
    '\0.0.0.0/0\ my-key my-value us-east-1'
  exit 1
# If no values are specified at the command prompt, use these default values.
elsif ARGV.count.zero?
  vpc_id = 'vpc-0b6f769731EXAMPLE'
  subnet_id = 'subnet-03d9303b57EXAMPLE'
  gateway_id = 'igw-06ca90c011EXAMPLE'
  destination_cidr_block = '0.0.0.0/0'
  tag_key = 'my-key'
  tag_value = 'my-value'
  region = 'us-east-1'
# Otherwise, use the values as specified at the command prompt.
else
  vpc_id = ARGV[0]
  subnet_id = ARGV[1]
  gateway_id = ARGV[2]
  destination_cidr_block = ARGV[3]
  tag_key = ARGV[4]
  tag_value = ARGV[5]
  region = ARGV[6]
end

ec2_resource = Aws::EC2::Resource.new(region: region)

if route_table_created_and_associated?(
  ec2_resource,
  vpc_id,
  subnet_id,
  gateway_id,
  destination_cidr_block,
  tag_key,
  tag_value
)
```



```
puts 'Route table created and associated.'
else
  puts 'Route table not created or not associated.'
end
end
end

run_me if $PROGRAM_NAME == __FILE__
```

在 Amazon EC2 中使用彈性 IP 地址

下面的代碼示例：

1. 顯示與 Amazon Elastic Compute Cloud (Amazon EC2) 執行個體相關聯的任何地址的相關資訊。
2. 在 Amazon Virtual Private Cloud (亞馬遜 VPC) 中創建一個彈性 IP 地址。
3. 將位址與執行個體建立關聯。
4. 再次顯示與執行個體相關聯位址的相關資訊。這次，應該會顯示新的位址關聯。
5. 釋放地址。
6. 再次顯示與執行個體相關聯位址的相關資訊。這次，不應顯示已發行的地址。

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX - License - Identifier: Apache - 2.0

# This code example does the following:
# 1. Displays information about any addresses associated with an
#   Amazon Elastic Compute Cloud (Amazon EC2) instance.
# 2. Creates an Elastic IP address in Amazon Virtual Private Cloud (Amazon VPC).
# 3. Associates the address with the instance.
# 4. Displays information again about addresses associated with the instance.
#   This time, the new address association should display.
# 5. Releases the address.
# 6. Displays information again about addresses associated with the instance.
#   This time, the released address should not display.

require 'aws-sdk-ec2'

# Checks whether the specified Amazon Elastic Compute Cloud
# (Amazon EC2) instance exists.
#
# Prerequisites:
#
```

```
# - The Amazon EC2 instance.
#
# @param ec2_client [Aws::EC2::Client] An initialized EC2 client.
# @param instance_id [String] The ID of the instance.
# @return [Boolean] true if the instance exists; otherwise, false.
# @example
#   exit 1 unless instance_exists?(
#     Aws::EC2::Client.new(region: 'us-east-1'),
#     'i-033c48ef067af3dEX'
#   )
def instance_exists?(ec2_client, instance_id)
  ec2_client.describe_instances(instance_ids: [instance_id])
  return true
rescue StandardError
  return false
end

# Creates an Elastic IP address in Amazon Virtual Private Cloud (Amazon VPC).
#
# @param ec2_client [Aws::EC2::Client] An initialized EC2 client.
# @return [String] The allocation ID corresponding to the Elastic IP address.
# @example
#   puts allocate_elastic_ip_address(Aws::EC2::Client.new(region: 'us-east-1'))
def allocate_elastic_ip_address(ec2_client)
  response = ec2_client.allocate_address(domain: 'vpc')
  return response.allocation_id
rescue StandardError => e
  puts "Error allocating Elastic IP address: #{e.message}"
  return 'Error'
end

# Associates an Elastic IP address with an Amazon Elastic Compute Cloud
# (Amazon EC2) instance.
#
# Prerequisites:
#
# - The allocation ID corresponding to the Elastic IP address.
# - The Amazon EC2 instance.
#
# @param ec2_client [Aws::EC2::Client] An initialized EC2 client.
# @param allocation_id [String] The ID of the allocation corresponding to
#   the Elastic IP address.
# @param instance_id [String] The ID of the instance.
# @return [String] The association ID corresponding to the association of the
```

```
# Elastic IP address to the instance.
# @example
# puts allocate_elastic_ip_address(
#   Aws::EC2::Client.new(region: 'us-east-1'),
#   'eipalloc-04452e528a66279EX',
#   'i-033c48ef067af3dEX')
def associate_elastic_ip_address_with_instance(
  ec2_client,
  allocation_id,
  instance_id
)
  response = ec2_client.associate_address(
    allocation_id: allocation_id,
    instance_id: instance_id,
  )
  return response.association_id
rescue StandardError => e
  puts "Error associating Elastic IP address with instance: #{e.message}"
  return 'Error'
end

# Gets information about addresses associated with an
# Amazon Elastic Compute Cloud (Amazon EC2) instance.
#
# Prerequisites:
#
# - The Amazon EC2 instance.
#
# @param ec2_client [Aws::EC2::Client] An initialized EC2 client.
# @param instance_id [String] The ID of the instance.
# @example
# describe_addresses_for_instance(
#   Aws::EC2::Client.new(region: 'us-east-1'),
#   'i-033c48ef067af3dEX'
# )
def describe_addresses_for_instance(ec2_client, instance_id)
  response = ec2_client.describe_addresses(
    filters: [
      {
        name: 'instance-id',
        values: [instance_id]
      }
    ]
  )
end
```

```
addresses = response.addresses
if addresses.count.zero?
  puts 'No addresses.'
else
  addresses.each do |address|
    puts '-' * 20
    puts "Public IP: #{address.public_ip}"
    puts "Private IP: #{address.private_ip_address}"
  end
end
rescue StandardError => e
  puts "Error getting address information for instance: #{e.message}"
end

# Releases an Elastic IP address from an
# Amazon Elastic Compute Cloud (Amazon EC2) instance.
#
# Prerequisites:
#
# - An Amazon EC2 instance with an associated Elastic IP address.
#
# @param ec2_client [Aws::EC2::Client] An initialized EC2 client.
# @param allocation_id [String] The ID of the allocation corresponding to
# the Elastic IP address.
# @return [Boolean] true if the Elastic IP address was released;
# otherwise, false.
# @example
# exit 1 unless elastic_ip_address_released?(
#   Aws::EC2::Client.new(region: 'us-east-1'),
#   'eipalloc-04452e528a66279EX'
# )
def elastic_ip_address_released?(ec2_client, allocation_id)
  ec2_client.release_address(allocation_id: allocation_id)
  return true
rescue StandardError => e
  return "Error releasing Elastic IP address: #{e.message}"
  return false
end

# Full example call:
def run_me
  instance_id = ''
  region = ''
  # Print usage information and then stop.
```

```
if ARGV[0] == '--help' || ARGV[0] == '-h'
  puts 'Usage:  ruby ec2-ruby-example-elastic-ips.rb ' \
        'INSTANCE_ID REGION'
  puts 'Example: ruby ec2-ruby-example-elastic-ips.rb ' \
        'i-033c48ef067af3dEX us-east-1'
  exit 1
# If no values are specified at the command prompt, use these default values.
elsif ARGV.count.zero?
  instance_id = 'i-033c48ef067af3dEX'
  region = 'us-east-1'
# Otherwise, use the values as specified at the command prompt.
else
  instance_id = ARGV[0]
  region = ARGV[1]
end

ec2_client = Aws::EC2::Client.new(region: region)

unless instance_exists?(ec2_client, instance_id)
  puts "Cannot find instance with ID '#{instance_id}'. Stopping program."
  exit 1
end

puts "Addresses for instance with ID '#{instance_id}' before allocating " \
      'Elastic IP address:'
describe_addresses_for_instance(ec2_client, instance_id)

puts 'Allocating Elastic IP address...'
allocation_id = allocate_elastic_ip_address(ec2_client)
if allocation_id.start_with?('Error')
  puts 'Stopping program.'
  exit 1
else
  puts "Elastic IP address created with allocation ID '#{allocation_id}'."
end

puts 'Associating Elastic IP address with instance...'
association_id = associate_elastic_ip_address_with_instance(
  ec2_client,
  allocation_id,
  instance_id
)
if association_id.start_with?('Error')
  puts 'Stopping program. You must associate the Elastic IP address yourself.'
```

```

    exit 1
  else
    puts 'Elastic IP address associated with instance with association ID ' \
        "'#{association_id}'."
  end

  puts 'Addresses for instance after allocating Elastic IP address:'
  describe_addresses_for_instance(ec2_client, instance_id)

  puts 'Releasing the Elastic IP address from the instance...'
  if elastic_ip_address_released?(ec2_client, allocation_id) == false
    puts 'Stopping program. You must release the Elastic IP address yourself.'
    exit 1
  else
    puts 'Address released.'
  end

  puts 'Addresses for instance after releasing Elastic IP address:'
  describe_addresses_for_instance(ec2_client, instance_id)
end

run_me if $PROGRAM_NAME == __FILE__

```

創建一個 Amazon EC2 安全組

下列程式碼範例會建立 Amazon EC2 安全群組，然後將輸出規則新增至該安全群組。

```

# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX - License - Identifier: Apache - 2.0

require 'aws-sdk-ec2'

# Creates an Amazon Elastic Compute Cloud (Amazon EC2) security group and
# then adds an outbound rule to that security group.
#
# Prerequisites:
#
# - A VPC in Amazon Virtual Private Cloud (Amazon VPC).
#
# @param ec2_resource [Aws::EC2::Resource] An initialized
#   Amazon EC2 resource object.
# @param group_name [String] A name for the security group.
# @param description [String] A description for the security group.

```

```
# @param vpc_id [String] The ID of the VPC for the security group.
# @param protocol [String] The network protocol for the outbound rule.
# @param from_port [String] The originating port for the outbound rule.
# @param to_port [String] The destination port for the outbound rule.
# @param cidr_ip_range [String] The CIDR IP range for the outbound rule.
# @return [Boolean] true if the security group was created and the outbound
#   rule was added; otherwise, false.
# @example
#   exit 1 unless security_group_created_with_egress?(
#     Aws::EC2::Resource.new(region: 'us-east-1'),
#     'my-security-group',
#     'This is my security group.',
#     'vpc-6713dfEX',
#     'tcp',
#     '22',
#     '22',
#     '0.0.0.0/0'
#   )
def security_group_created_with_egress?(
  ec2_resource,
  group_name,
  description,
  vpc_id,
  ip_protocol,
  from_port,
  to_port,
  cidr_ip_range
)
  security_group = ec2_resource.create_security_group(
    group_name: group_name,
    description: description,
    vpc_id: vpc_id
  )
  puts "Created security group '#{group_name}' with ID " \
    "'#{security_group.id}' in VPC with ID '#{vpc_id}'."
  security_group.authorize_egress(
    ip_permissions: [
      {
        ip_protocol: ip_protocol,
        from_port: from_port,
        to_port: to_port,
        ip_ranges: [
          {
            cidr_ip: cidr_ip_range
```

```
    }
  ]
}
]
)
puts "Granted egress to security group '#{group_name}' for protocol " \
    "'#{ip_protocol}' from port '#{from_port}' to port '#{to_port}' " \
    "with CIDR IP range '#{cidr_ip_range}'."
return true
rescue StandardError => e
  puts "Error creating security group or granting egress: #{e.message}"
  return false
end

# Full example call:
def run_me
  group_name = ''
  description = ''
  vpc_id = ''
  ip_protocol = ''
  from_port = ''
  to_port = ''
  cidr_ip_range = ''
  region = ''
  # Print usage information and then stop.
  if ARGV[0] == '--help' || ARGV[0] == '-h'
    puts 'Usage: ruby ec2-ruby-example-create-security-group.rb ' \
        'GROUP_NAME DESCRIPTION VPC_ID IP_PROTOCOL FROM_PORT TO_PORT ' \
        'CIDR_IP_RANGE REGION'
    puts 'Example: ruby ec2-ruby-example-create-security-group.rb ' \
        'my-security-group \'This is my security group.\' vpc-6713dfEX ' \
        'tcp 22 22 \'0.0.0.0/0\' us-east-1'
    exit 1
  # If no values are specified at the command prompt, use these default values.
  elsif ARGV.count.zero?
    group_name = 'my-security-group'
    description = 'This is my security group.'
    vpc_id = 'vpc-6713dfEX'
    ip_protocol = 'tcp'
    from_port = '22'
    to_port = '22'
    cidr_ip_range = '0.0.0.0/0'
    region = 'us-east-1'
  # Otherwise, use the values as specified at the command prompt.
```



```
else
  group_name = ARGV[0]
  description = ARGV[1]
  vpc_id = ARGV[2]
  ip_protocol = ARGV[3]
  from_port = ARGV[4]
  to_port = ARGV[5]
  cidr_ip_range = ARGV[6]
  region = ARGV[7]
end

ec2_resource = Aws::EC2::Resource.new(region: region)

if security_group_created_with_egress?(
  ec2_resource,
  group_name,
  description,
  vpc_id,
  ip_protocol,
  from_port,
  to_port,
  cidr_ip_range
)
  puts 'Security group created and egress granted.'
else
  puts 'Security group not created or egress not granted.'
end
end

run_me if $PROGRAM_NAME == __FILE__
```

使用 Amazon EC2 安全群組

以下為範例示範：

1. 建立 Amazon EC2 安全群組。
2. 將輸入規則新增至安全性群組。
3. 顯示可用安全性群組的相關資訊。
4. 刪除安全性群組。

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
```

```
# SPDX - License - Identifier: Apache - 2.0

# This code example does the following:
# 1. Creates an Amazon Elastic Compute Cloud (Amazon EC2) security group.
# 2. Adds inbound rules to the security group.
# 3. Displays information about available security groups.
# 4. Deletes the security group.

require 'aws-sdk-ec2'

# Creates an Amazon Elastic Compute Cloud (Amazon EC2) security group.
#
# Prerequisites:
#
# - A VPC in Amazon Virtual Private Cloud (Amazon VPC).
#
# @param ec2_client [Aws::EC2::Client] An initialized
#   Amazon EC2 client.
# @param group_name [String] A name for the security group.
# @param description [String] A description for the security group.
# @param vpc_id [String] The ID of the VPC for the security group.
# @return [String] The ID of security group that was created.
# @example
#   puts create_security_group(
#     Aws::EC2::Client.new(region: 'us-east-1'),
#     'my-security-group',
#     'This is my security group.',
#     'vpc-6713dfEX'
#   )
def create_security_group(
  ec2_client,
  group_name,
  description,
  vpc_id
)
  security_group = ec2_client.create_security_group(
    group_name: group_name,
    description: description,
    vpc_id: vpc_id
  )
  puts "Created security group '#{group_name}' with ID " \
    "'#{security_group.group_id}' in VPC with ID '#{vpc_id}'."
  return security_group.group_id
rescue StandardError => e
```

```
puts "Error creating security group: #{e.message}"
return 'Error'
end

# Adds an inbound rule to an Amazon Elastic Compute Cloud (Amazon EC2)
# security group.
#
# Prerequisites:
#
# - The security group.
#
# @param ec2_client [Aws::EC2::Client] An initialized Amazon EC2 client.
# @param security_group_id [String] The ID of the security group.
# @param ip_protocol [String] The network protocol for the inbound rule.
# @param from_port [String] The originating port for the inbound rule.
# @param to_port [String] The destination port for the inbound rule.
# @param cidr_ip_range [String] The CIDR IP range for the inbound rule.
# @return
# @example
#   exit 1 unless security_group_ingress_authorized?(
#     Aws::EC2::Client.new(region: 'us-east-1'),
#     'sg-030a858e078f1b9EX',
#     'tcp',
#     '80',
#     '80',
#     '0.0.0.0/0'
#   )
def security_group_ingress_authorized?(
  ec2_client,
  security_group_id,
  ip_protocol,
  from_port,
  to_port,
  cidr_ip_range
)
  ec2_client.authorize_security_group_ingress(
    group_id: security_group_id,
    ip_permissions: [
      {
        ip_protocol: ip_protocol,
        from_port: from_port,
        to_port: to_port,
        ip_ranges: [
          {
```

```

        cidr_ip: cidr_ip_range
      }
    ]
  }
]
)
puts "Added inbound rule to security group '#{security_group_id}' for protocol " \
    "'#{ip_protocol}' from port '#{from_port}' to port '#{to_port}' " \
    "with CIDR IP range '#{cidr_ip_range}'."
return true
rescue StandardError => e
  puts "Error adding inbound rule to security group: #{e.message}"
  return false
end

# Displays information about a security group's IP permissions set in
# Amazon Elastic Compute Cloud (Amazon EC2).
#
# Prerequisites:
#
# - A security group with inbound rules, outbound rules, or both.
#
# @param p [Aws::EC2::Types::IpPermission] The IP permissions set.
# @example
#   ec2_client = Aws::EC2::Client.new(region: 'us-east-1')
#   response = ec2_client.describe_security_groups
#   unless sg.ip_permissions.empty?
#     describe_security_group_permissions(
#       response.security_groups[0].ip_permissions[0]
#     )
#   end
def describe_security_group_permissions(perm)
  print " Protocol: #{perm.ip_protocol == '-1' ? 'All' : perm.ip_protocol}"

  unless perm.from_port.nil?
    if perm.from_port == '-1' || perm.from_port == -1
      print ', From: All'
    else
      print ", From: #{perm.from_port}"
    end
  end
end

unless perm.to_port.nil?
  if perm.to_port == '-1' || perm.to_port == -1

```

```
    print ', To: All'
  else
    print ", To: #{perm.to_port}"
  end
end

if perm.key?(:ipv_6_ranges) && perm.ipv_6_ranges.count.positive?
  print ", CIDR IPv6: #{perm.ipv_6_ranges[0].cidr_ipv_6}"
end

if perm.key?(:ip_ranges) && perm.ip_ranges.count.positive?
  print ", CIDR IPv4: #{perm.ip_ranges[0].cidr_ip}"
end

print "\n"
end

# Displays information about available security groups in
# Amazon Elastic Compute Cloud (Amazon EC2).
#
# @param ec2_client [Aws::EC2::Client] An initialized Amazon EC2 client.
# @example
#   describe_security_groups(Aws::EC2::Client.new(region: 'us-east-1'))
def describe_security_groups(ec2_client)
  response = ec2_client.describe_security_groups

  if response.security_groups.count.positive?
    response.security_groups.each do |sg|
      puts '-' * (sg.group_name.length + 13)
      puts "Name:      #{sg.group_name}"
      puts "Description: #{sg.description}"
      puts "Group ID:    #{sg.group_id}"
      puts "Owner ID:    #{sg.owner_id}"
      puts "VPC ID:     #{sg.vpc_id}"

      if sg.tags.count.positive?
        puts 'Tags:'
        sg.tags.each do |tag|
          puts "  Key: #{tag.key}, Value: #{tag.value}"
        end
      end
    end

    unless sg.ip_permissions.empty?
      puts 'Inbound rules:' if sg.ip_permissions.count.positive?
    end
  end
end
```

```

    sg.ip_permissions.each do |p|
      describe_security_group_permissions(p)
    end
  end

  unless sg.ip_permissions_egress.empty?
    puts 'Outbound rules:' if sg.ip_permissions.count.positive?
    sg.ip_permissions_egress.each do |p|
      describe_security_group_permissions(p)
    end
  end
end
else
  puts 'No security groups found.'
end
rescue StandardError => e
  puts "Error getting information about security groups: #{e.message}"
end

# Deletes an Amazon Elastic Compute Cloud (Amazon EC2)
# security group.
#
# Prerequisites:
#
# - The security group.
#
# @param ec2_client [Aws::EC2::Client] An initialized
#   Amazon EC2 client.
# @param security_group_id [String] The ID of the security group to delete.
# @return [Boolean] true if the security group was deleted; otherwise, false.
# @example
#   exit 1 unless security_group_deleted?(
#     Aws::EC2::Client.new(region: 'us-east-1'),
#     'sg-030a858e078f1b9EX'
#   )
def security_group_deleted?(ec2_client, security_group_id)
  ec2_client.delete_security_group(group_id: security_group_id)
  puts "Deleted security group '#{security_group_id}'."
  return true
rescue StandardError => e
  puts "Error deleting security group: #{e.message}"
  return false
end
end

```

```
# Full example call:
def run_me
  group_name = ''
  description = ''
  vpc_id = ''
  ip_protocol_http = ''
  from_port_http = ''
  to_port_http = ''
  cidr_ip_range_http = ''
  ip_protocol_ssh = ''
  from_port_ssh = ''
  to_port_ssh = ''
  cidr_ip_range_ssh = ''
  region = ''
  # Print usage information and then stop.
  if ARGV[0] == '--help' || ARGV[0] == '-h'
    puts 'Usage:  ruby ec2-ruby-example-security-group.rb ' \
        'GROUP_NAME DESCRIPTION VPC_ID IP_PROTOCOL_1 FROM_PORT_1 TO_PORT_1 ' \
        'CIDR_IP_RANGE_1 IP_PROTOCOL_2 FROM_PORT_2 TO_PORT_2 ' \
        'CIDR_IP_RANGE_2 REGION'
    puts 'Example: ruby ec2-ruby-example-security-group.rb ' \
        'my-security-group \'This is my security group.\' vpc-6713dfEX ' \
        'tcp 80 80 \'0.0.0.0/0\' tcp 22 22 \'0.0.0.0/0\' us-east-1'
    exit 1
  # If no values are specified at the command prompt, use these default values.
  elsif ARGV.count.zero?
    group_name = 'my-security-group'
    description = 'This is my security group.'
    vpc_id = 'vpc-6713dfEX'
    ip_protocol_http = 'tcp'
    from_port_http = '80'
    to_port_http = '80'
    cidr_ip_range_http = '0.0.0.0/0'
    ip_protocol_ssh = 'tcp'
    from_port_ssh = '22'
    to_port_ssh = '22'
    cidr_ip_range_ssh = '0.0.0.0/0'
    region = 'us-east-1'
  # Otherwise, use the values as specified at the command prompt.
  else
    group_name = ARGV[0]
    description = ARGV[1]
    vpc_id = ARGV[2]
    ip_protocol_http = ARGV[3]
```

```
from_port_http = ARGV[4]
to_port_http = ARGV[5]
cidr_ip_range_http = ARGV[6]
ip_protocol_ssh = ARGV[7]
from_port_ssh = ARGV[8]
to_port_ssh = ARGV[9]
cidr_ip_range_ssh = ARGV[10]
region = ARGV[11]
end

security_group_id = ''
security_group_exists = false
ec2_client = Aws::EC2::Client.new(region: region)

puts 'Attempting to create security group...'
security_group_id = create_security_group(
  ec2_client,
  group_name,
  description,
  vpc_id
)
if security_group_id == 'Error'
  puts 'Could not create security group. Skipping this step.'
else
  security_group_exists = true
end

if security_group_exists
  puts 'Attempting to add inbound rules to security group...'
  unless security_group_ingress_authorized?(
    ec2_client,
    security_group_id,
    ip_protocol_http,
    from_port_http,
    to_port_http,
    cidr_ip_range_http
  )
    puts 'Could not add inbound HTTP rule to security group. ' \
      'Skipping this step.'
  end

  unless security_group_ingress_authorized?(
    ec2_client,
    security_group_id,
```



```
    ip_protocol_ssh,
    from_port_ssh,
    to_port_ssh,
    cidr_ip_range_ssh
  )
  puts 'Could not add inbound SSH rule to security group. ' \
    'Skipping this step.'
end
end

puts "\nInformation about available security groups:"
describe_security_groups(ec2_client)

if security_group_exists
  puts "\nAttempting to delete security group..."
  unless security_group_deleted?(ec2_client, security_group_id)
    puts 'Could not delete security group. You must delete it yourself.'
  end
end
end

run_me if $PROGRAM_NAME == __FILE__
```

在 Amazon EC2 中使用密鑰對

下面的代碼示例：

1. 在 Amazon EC2 中創建一個 key pair。
2. 顯示有關可用金鑰配對的資訊。
3. 刪除 key pair。

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX - License - Identifier: Apache - 2.0

# This code example does the following:
# 1. Creates a key pair in Amazon Elastic Compute Cloud (Amazon EC2).
# 2. Displays information about available key pairs.
# 3. Deletes the key pair.

require 'aws-sdk-ec2'
```

```

# Creates a key pair in Amazon Elastic Compute Cloud (Amazon EC2) and
# saves the resulting RSA private key file locally in the calling
# user's home directory.
#
# @param ec2_client [Aws::EC2::Client] An initialized EC2 client.
# @param key_pair_name [String] The name for the key pair and private
#   key file.
# @return [Boolean] true if the key pair and private key file were
#   created; otherwise, false.
# @example
#   exit 1 unless key_pair_created?(
#     Aws::EC2::Client.new(region: 'us-east-1'),
#     'my-key-pair'
#   )
def key_pair_created?(ec2_client, key_pair_name)
  key_pair = ec2_client.create_key_pair(key_name: key_pair_name)
  puts "Created key pair '#{key_pair.key_name}' with fingerprint " \
    "'#{key_pair.key_fingerprint}' and ID '#{key_pair.key_pair_id}'."
  filename = File.join(Dir.home, key_pair_name + '.pem')
  File.open(filename, 'w') { |file| file.write(key_pair.key_material) }
  puts "Private key file saved locally as '#{filename}'."
  return true
rescue Aws::EC2::Errors::InvalidKeyPairDuplicate
  puts "Error creating key pair: a key pair named '#{key_pair_name}' " \
    'already exists.'
  return false
rescue StandardError => e
  puts "Error creating key pair or saving private key file: #{e.message}"
  return false
end

# Displays information about available key pairs in
# Amazon Elastic Compute Cloud (Amazon EC2).
#
# @param ec2_client [Aws::EC2::Client] An initialized EC2 client.
# @example
#   describe_key_pairs(Aws::EC2::Client.new(region: 'us-east-1'))
def describe_key_pairs(ec2_client)
  result = ec2_client.describe_key_pairs
  if result.key_pairs.count.zero?
    puts 'No key pairs found.'
  else
    puts 'Key pair names:'
    result.key_pairs.each do |key_pair|

```

```
        puts key_pair.key_name
      end
    end
  end
rescue StandardError => e
  puts "Error getting information about key pairs: #{e.message}"
end

# Deletes a key pair in Amazon Elastic Compute Cloud (Amazon EC2).
#
# Prerequisites:
#
# - The key pair to delete.
#
# @param ec2_client [Aws::EC2::Client] An initialized EC2 client.
# @param key_pair_name [String] The name of the key pair to delete.
# @return [Boolean] true if the key pair was deleted; otherwise, false.
# @example
#   exit 1 unless key_pair_deleted?(
#     Aws::EC2::Client.new(region: 'us-east-1'),
#     'my-key-pair'
#   )
def key_pair_deleted?(ec2_client, key_pair_name)
  ec2_client.delete_key_pair(key_name: key_pair_name)
  return true
rescue StandardError => e
  puts "Error deleting key pair: #{e.message}"
  return false
end

# Full example call:
def run_me
  key_pair_name = ''
  region = ''
  # Print usage information and then stop.
  if ARGV[0] == '--help' || ARGV[0] == '-h'
    puts 'Usage:  ruby ec2-ruby-example-key-pairs.rb KEY_PAIR_NAME REGION'
    puts 'Example: ruby ec2-ruby-example-key-pairs.rb my-key-pair us-east-1'
    exit 1
  # If no values are specified at the command prompt, use these default values.
  elsif ARGV.count.zero?
    key_pair_name = 'my-key-pair'
    region = 'us-east-1'
  # Otherwise, use the values as specified at the command prompt.
  else
```

```
    key_pair_name = ARGV[0]
    region = ARGV[1]
end

ec2_client = Aws::EC2::Client.new(region: region)

puts 'Displaying existing key pair names before creating this key pair...'
describe_key_pairs(ec2_client)

puts '-' * 10
puts 'Creating key pair...'
unless key_pair_created?(ec2_client, key_pair_name)
  puts 'Stopping program.'
  exit 1
end

puts '-' * 10
puts 'Displaying existing key pair names after creating this key pair...'
describe_key_pairs(ec2_client)

puts '-' * 10
puts 'Deleting key pair...'
unless key_pair_deleted?(ec2_client, key_pair_name)
  puts 'Stopping program. You must delete the key pair yourself.'
  exit 1
end
puts 'Key pair deleted.'

puts '-' * 10
puts 'Now that the key pair is deleted, ' \
      'also deleting the related private key pair file...'
filename = File.join(Dir.home, key_pair_name + '.pem')
File.delete(filename)
if File.exist?(filename)
  puts "Could not delete file at '#{filename}'. You must delete it yourself."
else
  puts 'File deleted.'
end

puts '-' * 10
puts 'Displaying existing key pair names after deleting this key pair...'
describe_key_pairs(ec2_client)
end
```

```
run_me if $PROGRAM_NAME == __FILE__
```

取得所有 Amazon EC2 執行個體的相關資訊

下列程式碼範例會列出可用 Amazon EC2 執行個體的 ID 和目前狀態。

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX - License - Identifier: Apache - 2.0

require 'aws-sdk-ec2'

# Lists the IDs and current states of available
# Amazon Elastic Compute Cloud (Amazon EC2) instances.
#
# @param ec2_resource [Aws::EC2::Resource] An initialized EC2 resource object.
# @example
# list_instance_ids_states(Aws::EC2::Resource.new(region: 'us-east-1'))
def list_instance_ids_states(ec2_resource)
  response = ec2_resource.instances
  if response.count.zero?
    puts 'No instances found.'
  else
    puts 'Instances -- ID, state:'
    response.each do |instance|
      puts "#{instance.id}, #{instance.state.name}"
    end
  end
end

rescue StandardError => e
  puts "Error getting information about instances: #{e.message}"
end

#Full example call:
def run_me
  region = ''
  # Print usage information and then stop.
  if ARGV[0] == '--help' || ARGV[0] == '-h'
    puts 'Usage: ruby ec2-ruby-example-get-all-instance-info.rb REGION'
    puts 'Example: ruby ec2-ruby-example-get-all-instance-info.rb us-east-1'
    exit 1
  # If no values are specified at the command prompt, use these default values.
  elsif ARGV.count.zero?
    region = 'us-east-1'
  # Otherwise, use the values as specified at the command prompt.
```

```
else
  region = ARGV[0]
end
ec2_resource = Aws::EC2::Resource.new(region: region)
list_instance_ids_states(ec2_resource)
end

run_me if $PROGRAM_NAME == __FILE__
```

取得具有特定標籤值之所有 Amazon EC2 執行個體的相關資訊

下列程式碼範例列出符合指定標籤金鑰和值的可用 Amazon EC2 執行個體的 ID 和目前狀態。

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX - License - Identifier: Apache - 2.0

require 'aws-sdk-ec2'

# Lists the IDs, current states, and tag keys/values of matching
# available Amazon Elastic Compute Cloud (Amazon EC2) instances.
#
# @param ec2_resource [Aws::EC2::Resource] An initialized EC2 resource object.
# @param tag_key [String] The key portion of the tag to search on.
# @param tag_value [String] The value portion of the tag to search on.
# @example
#   list_instance_ids_states_by_tag(
#     Aws::EC2::Resource.new(region: 'us-east-1'),
#     'my-key',
#     'my-value'
#   )
def list_instance_ids_states_by_tag(ec2_resource, tag_key, tag_value)
  response = ec2_resource.instances(
    filters: [
      {
        name: "tag:#{tag_key}",
        values: [tag_value]
      }
    ]
  )
  if response.count.zero?
    puts 'No matching instances found.'
  else
    puts 'Matching instances -- ID, state, tag key/value:'
```

```
response.each do |instance|
  print "#{instance.id}, #{instance.state.name}"
  instance.tags.each do |tag|
    print ", #{tag.key}/#{tag.value}"
  end
  print "\n"
end
end
rescue StandardError => e
  puts "Error getting information about instances: #{e.message}"
end

#Full example call:
def run_me
  tag_key = ''
  tag_value = ''
  region = ''
  # Print usage information and then stop.
  if ARGV[0] == '--help' || ARGV[0] == '-h'
    puts 'Usage:  ruby ec2-ruby-example-get-instance-info-by-tag.rb ' \
      'TAG_KEY TAG_VALUE REGION'
    puts 'Example: ruby ec2-ruby-example-get-instance-info-by-tag.rb ' \
      'my-key my-value us-east-1'
    exit 1
  # If no values are specified at the command prompt, use these default values.
  elsif ARGV.count.zero?
    tag_key = 'my-key'
    tag_value = 'my-value'
    region = 'us-east-1'
  # Otherwise, use the values as specified at the command prompt.
  else
    tag_key = ARGV[0]
    tag_value = ARGV[1]
    region = ARGV[2]
  end
  ec2_resource = Aws::EC2::Resource.new(region: region)
  list_instance_ids_states_by_tag(ec2_resource, tag_key, tag_value)
end

run_me if $PROGRAM_NAME == __FILE__
```

取得特定 Amazon EC2 執行個體的相關資訊

下列範例列出指定之 Amazon EC2 執行個體的狀態。

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX - License - Identifier: Apache - 2.0

require 'aws-sdk-ec2'

# Lists the state of an Amazon Elastic Compute Cloud (Amazon EC2) instance.
#
# Prerequisites:
#
# - An Amazon EC2 instance.
#
# @param ec2_client [Aws::EC2::Client] An initialized EC2 client.
# @param instance_id [String] The ID of the instance.
# @example
#   list_instance_state(
#     Aws::EC2::Resource.new(region: 'us-east-1'),
#     'i-123abc'
#   )
def list_instance_state(ec2_client, instance_id)
  response = ec2_client.describe_instances(
    instance_ids: [instance_id]
  )
  if response.count.zero?
    puts 'No matching instance found.'
  else
    instance = response.reservations[0].instances[0]
    puts "The instance with ID '#{instance_id}' is '#{instance.state.name}'."
  end
end

rescue StandardError => e
  puts "Error getting information about instance: #{e.message}"
end

# Full example call:
def run_me
  instance_id = ''
  region = ''
  # Print usage information and then stop.
  if ARGV[0] == '--help' || ARGV[0] == '-h'
    puts 'Usage:  ruby ec2-ruby-example-list-state-instance-i-123abc.rb ' \
      'INSTANCE_ID REGION'
```



```
puts 'Example: ruby ec2-ruby-example-list-state-instance-i-123abc.rb ' \
     'i-123abc us-east-1'
exit 1
# If no values are specified at the command prompt, use these default values.
elsif ARGV.count.zero?
  instance_id = 'i-123abc'
  region = 'us-east-1'
# Otherwise, use the values as specified at the command prompt.
else
  instance_id = ARGV[0]
  region = ARGV[1]
end

ec2_client = Aws::EC2::Client.new(region: region)
list_instance_state(ec2_client, instance_id)
end

run_me if $PROGRAM_NAME == __FILE__
```

創建一個 Amazon EC2 實例

下列範例會建立 Amazon EC2 執行個體並加上標籤。

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX - License - Identifier: Apache - 2.0

require 'aws-sdk-ec2'
require 'base64'

# Creates and tags an Amazon Elastic Compute Cloud (Amazon EC2) instance.
#
# Prerequisites:
#
# - An EC2 key pair.
# - If you want to run any commands on the instance after it starts, a
#   file containing those commands.
#
# @param ec2_resource [Aws::EC2::Resource] An initialized EC2 resource object.
# @param image_id [String] The ID of the target Amazon Machine Image (AMI).
# @param key_pair_name [String] The name of the existing EC2 key pair.
# @param tag_key [String] The key portion of the tag for the instance.
# @param tag_value [String] The value portion of the tag for the instance.
# @param instance_type [String] The ID of the type of instance to create.
```

```
# If not specified, the default value is 't2.micro'.
# @param user_data_file [String] The path to the file containing any commands
# to run on the instance after it starts. If not specified, the default
# value is an empty string.
# @return [Boolean] true if the instance was created and tagged;
# otherwise, false.
# @example
# exit 1 unless instance_created?(
#   Aws::EC2::Resource.new(region: 'us-east-1'),
#   'ami-0947d2ba12EXAMPLE',
#   'my-key-pair',
#   'my-key',
#   'my-value',
#   't2.micro',
#   'my-user-data.txt'
# )
def instance_created?(
  ec2_resource,
  image_id,
  key_pair_name,
  tag_key,
  tag_value,
  instance_type = 't2.micro',
  user_data_file = ''
)
  encoded_script = ''

  unless user_data_file == ''
    script = File.read(user_data_file)
    encoded_script = Base64.encode64(script)
  end

  instance = ec2_resource.create_instances(
    image_id: image_id,
    min_count: 1,
    max_count: 1,
    key_name: key_pair_name,
    instance_type: instance_type,
    user_data: encoded_script
  )

  puts 'Creating instance...'

  # Check whether the new instance is in the "running" state.
```

```
polls = 0
loop do
  polls += 1
  response = ec2_resource.client.describe_instances(
    instance_ids: [
      instance.first.id
    ]
  )
  # Stop polling after 10 minutes (40 polls * 15 seconds per poll) if not running.
  break if response.reservations[0].instances[0].state.name == 'running' || polls >
40

  sleep(15)
end

puts "Instance created with ID '#{instance.first.id}'."

instance.batch_create_tags(
  tags: [
    {
      key: tag_key,
      value: tag_value
    }
  ]
)
puts 'Instance tagged.'

return true
rescue StandardError => e
  puts "Error creating or tagging instance: #{e.message}"
  return false
end

# Full example call:
def run_me
  image_id = ''
  key_pair_name = ''
  tag_key = ''
  tag_value = ''
  instance_type = ''
  region = ''
  user_data_file = ''
  # Print usage information and then stop.
  if ARGV[0] == '--help' || ARGV[0] == '-h'
```

```
puts 'Usage: ruby ec2-ruby-example-create-instance.rb ' \
      'IMAGE_ID KEY_PAIR_NAME TAG_KEY TAG_VALUE INSTANCE_TYPE ' \
      'REGION [USER_DATA_FILE]'
puts 'Example: ruby ec2-ruby-example-create-instance.rb ' \
      'ami-0947d2ba12EXAMPLE my-key-pair my-key my-value t2.micro ' \
      'us-east-1 my-user-data.txt'
exit 1
# If no values are specified at the command prompt, use these default values.
elsif ARGV.count.zero?
  image_id = 'ami-0947d2ba12EXAMPLE'
  key_pair_name = 'my-key-pair'
  tag_key = 'my-key'
  tag_value = 'my-value'
  instance_type = 't2.micro'
  region = 'us-east-1'
  user_data_file = 'my-user-data.txt'
# Otherwise, use the values as specified at the command prompt.
else
  image_id = ARGV[0]
  key_pair_name = ARGV[1]
  tag_key = ARGV[2]
  tag_value = ARGV[3]
  instance_type = ARGV[4]
  region = ARGV[5]
  user_data_file = ARGV[6] if ARGV.count == 7 # If user data file specified.
end

ec2_resource = Aws::EC2::Resource.new(region: region)

if instance_created?(
  ec2_resource,
  image_id,
  key_pair_name,
  tag_key,
  tag_value,
  instance_type,
  user_data_file
)
  puts 'Created and tagged instance.'
else
  puts 'Could not create or tag instance.'
end
end
```

```
run_me if $PROGRAM_NAME == __FILE__
```

停止亞 Amazon EC2 實例

下列範例會嘗試停止指定的 Amazon EC2 執行個體。

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX - License - Identifier: Apache - 2.0

require 'aws-sdk-ec2'

# Attempts to stop an Amazon Elastic Compute Cloud (Amazon EC2) instance.
#
# Prerequisites:
#
# - The Amazon EC2 instance.
#
# @param ec2_client [Aws::EC2::Client] An initialized EC2 client.
# @param instance_id [String] The ID of the instance.
# @return [Boolean] true if the instance was stopped; otherwise, false.
# @example
#   exit 1 unless instance_stopped?(
#     Aws::EC2::Client.new(region: 'us-east-1'),
#     'i-123abc'
#   )
def instance_stopped?(ec2_client, instance_id)
  response = ec2_client.describe_instance_status(instance_ids: [instance_id])

  if response.instance_statuses.count.positive?
    state = response.instance_statuses[0].instance_state.name
    case state
    when 'stopping'
      puts 'The instance is already stopping.'
      return true
    when 'stopped'
      puts 'The instance is already stopped.'
      return true
    when 'terminated'
      puts 'Error stopping instance: ' \
        'the instance is terminated, so you cannot stop it.'
      return false
    end
  end
end
```

```
ec2_client.stop_instances(instance_ids: [instance_id])
ec2_client.wait_until(:instance_stopped, instance_ids: [instance_id])
puts 'Instance stopped.'
return true
rescue StandardError => e
  puts "Error stopping instance: #{e.message}"
  return false
end

# Full example call:
def run_me
  instance_id = ''
  region = ''
  # Print usage information and then stop.
  if ARGV[0] == '--help' || ARGV[0] == '-h'
    puts 'Usage:  ruby ec2-ruby-example-stop-instance-i-123abc.rb ' \
      'INSTANCE_ID REGION '
    puts 'Example: ruby ec2-ruby-example-start-instance-i-123abc.rb ' \
      'i-123abc us-east-1'
    exit 1
  # If no values are specified at the command prompt, use these default values.
  elsif ARGV.count.zero?
    instance_id = 'i-123abc'
    region = 'us-east-1'
  # Otherwise, use the values as specified at the command prompt.
  else
    instance_id = ARGV[0]
    region = ARGV[1]
  end

  ec2_client = Aws::EC2::Client.new(region: region)

  puts "Attempting to stop instance '#{instance_id}' " \
    '(this might take a few minutes)...'
  unless instance_stopped?(ec2_client, instance_id)
    puts 'Could not stop instance.'
  end
end

run_me if $PROGRAM_NAME == __FILE__
```

啟動一個 Amazon EC2 實例

下列範例會嘗試啟動指定的 Amazon EC2 執行個體。

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX - License - Identifier: Apache - 2.0

require 'aws-sdk-ec2'

# Attempts to start an Amazon Elastic Compute Cloud (Amazon EC2) instance.
#
# Prerequisites:
#
# - The Amazon EC2 instance.
#
# @param ec2_client [Aws::EC2::Client] An initialized EC2 client.
# @param instance_id [String] The ID of the instance.
# @return [Boolean] true if the instance was started; otherwise, false.
# @example
#   exit 1 unless instance_started?(
#     Aws::EC2::Client.new(region: 'us-east-1'),
#     'i-123abc'
#   )
def instance_started?(ec2_client, instance_id)
  response = ec2_client.describe_instance_status(instance_ids: [instance_id])

  if response.instance_statuses.count.positive?
    state = response.instance_statuses[0].instance_state.name
    case state
    when 'pending'
      puts 'Error starting instance: the instance is pending. Try again later.'
      return false
    when 'running'
      puts 'The instance is already running.'
      return true
    when 'terminated'
      puts 'Error starting instance: ' \
        'the instance is terminated, so you cannot start it.'
      return false
    end
  end
end

ec2_client.start_instances(instance_ids: [instance_id])
ec2_client.wait_until(:instance_running, instance_ids: [instance_id])
```

```
puts 'Instance started.'
return true
rescue StandardError => e
  puts "Error starting instance: #{e.message}"
  return false
end

# Full example call:
def run_me
  instance_id = ''
  region = ''
  # Print usage information and then stop.
  if ARGV[0] == '--help' || ARGV[0] == '-h'
    puts 'Usage:  ruby ec2-ruby-example-start-instance-i-123abc.rb ' \
        'INSTANCE_ID REGION '
    puts 'Example: ruby ec2-ruby-example-start-instance-i-123abc.rb ' \
        'i-123abc us-east-1'
    exit 1
  # If no values are specified at the command prompt, use these default values.
  elsif ARGV.count.zero?
    instance_id = 'i-123abc'
    region = 'us-east-1'
  # Otherwise, use the values as specified at the command prompt.
  else
    instance_id = ARGV[0]
    region = ARGV[1]
  end

  ec2_client = Aws::EC2::Client.new(region: region)

  puts "Attempting to start instance '#{instance_id}' " \
      '(this might take a few minutes)... '
  unless instance_started?(ec2_client, instance_id)
    puts 'Could not start instance.'
  end
end

run_me if $PROGRAM_NAME == __FILE__
```

重新啟動 Amazon EC2 實例

下列範例會嘗試重新啟動指定的 Amazon EC2 執行個體。


```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX - License - Identifier: Apache - 2.0

require 'aws-sdk-ec2'

# Reboots an Amazon Elastic Compute Cloud (Amazon EC2) instance.
#
# Prerequisites:
#
# - An Amazon EC2 instance.
#
# @param ec2_client [Aws::EC2::Client] An initialized EC2 client.
# @param instance_id [String] The ID of the instance.
# @example
#   request_instance_reboot(
#     Aws::EC2::Resource.new(region: 'us-east-1'),
#     'i-123abc'
#   )
def request_instance_reboot(ec2_client, instance_id)
  response = ec2_client.describe_instances(instance_ids: [instance_id])
  if response.count.zero?
    puts 'Error requesting reboot: no matching instance found.'
  else
    instance = response.reservations[0].instances[0]
    if instance.state.name == 'terminated'
      puts 'Error requesting reboot: the instance is already terminated.'
    else
      ec2_client.reboot_instances(instance_ids: [instance_id])
      puts 'Reboot request sent.'
    end
  end
end

rescue StandardError => e
  puts "Error requesting reboot: #{e.message}"
end

# Full example call:
def run_me
  instance_id = ''
  region = ''
  # Print usage information and then stop.
  if ARGV[0] == '--help' || ARGV[0] == '-h'
    puts 'Usage:  ruby ec2-ruby-example-reboot-instance-i-123abc.rb ' \
      'INSTANCE_ID REGION'
```

```
puts 'Example: ruby ec2-ruby-example-reboot-instance-i-123abc.rb ' \
     'i-123abc us-east-1'
exit 1
# If no values are specified at the command prompt, use these default values.
elsif ARGV.count.zero?
  instance_id = 'i-123abc'
  region = 'us-east-1'
# Otherwise, use the values as specified at the command prompt.
else
  instance_id = ARGV[0]
  region = ARGV[1]
end

ec2_client = Aws::EC2::Client.new(region: region)
request_instance_reboot(ec2_client, instance_id)
end

run_me if $PROGRAM_NAME == __FILE__
```

管理 Amazon EC2 實例

下面的代碼示例：

1. 停止亞 Amazon EC2 實例。
2. 重新啟動執行個體。
3. 重新啟動執行個體。
4. 啟用執行個體的詳細監視。
5. 顯示可用執行處理的資訊。

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX - License - Identifier: Apache - 2.0

# This code example does the following:
# 1. Stops an Amazon Elastic Compute Cloud (Amazon EC2) instance.
# 2. Restarts the instance.
# 3. Reboots the instance.
# 4. Enables detailed monitoring for the instance.
# 5. Displays information about available instances.

require 'aws-sdk-ec2'
```

```
# Waits for an Amazon Elastic Compute Cloud (Amazon EC2) instance
# to reach the specified state.
#
# Prerequisites:
#
# - The Amazon EC2 instance.
#
# @param ec2_client [Aws::EC2::Client] An initialized EC2 client.
# @param instance_state [Symbol] The desired instance state.
# @param instance_id [String] The ID of the instance.
# @example
#   wait_for_instance(
#     Aws::EC2::Client.new(region: 'us-east-1'),
#     :instance_stopped,
#     'i-033c48ef067af3dEX'
#   )
def wait_for_instance(ec2_client, instance_state, instance_id)
  ec2_client.wait_until(instance_state, instance_ids: [instance_id])
  puts "Success: #{instance_state}."
rescue Aws::Waiters::Errors::WaiterFailed => e
  puts "Failed: #{e.message}"
end

# Attempts to stop an Amazon Elastic Compute Cloud (Amazon EC2) instance.
#
# Prerequisites:
#
# - The Amazon EC2 instance.
#
# @param ec2_client [Aws::EC2::Client] An initialized EC2 client.
# @param instance_id [String] The ID of the instance.
# @return [Boolean] true if the instance was stopped; otherwise, false.
# @example
#   exit 1 unless instance_stopped?(
#     Aws::EC2::Client.new(region: 'us-east-1'),
#     'i-033c48ef067af3dEX'
#   )
def instance_stopped?(ec2_client, instance_id)
  ec2_client.stop_instances(instance_ids: [instance_id])
  wait_for_instance(ec2_client, :instance_stopped, instance_id)
  return true
rescue StandardError => e
  puts "Error stopping instance: #{e.message}"
  return false
end
```

```
end

# Attempts to restart an Amazon Elastic Compute Cloud (Amazon EC2) instance.
#
# Prerequisites:
#
# - The Amazon EC2 instance.
#
# @param ec2_client [Aws::EC2::Client] An initialized EC2 client.
# @param instance_id [String] The ID of the instance.
# @return [Boolean] true if the instance was restarted; otherwise, false.
# @example
#   exit 1 unless instance_restarted?(
#     Aws::EC2::Client.new(region: 'us-east-1'),
#     'i-033c48ef067af3dEX'
#   )
def instance_restarted?(ec2_client, instance_id)
  ec2_client.start_instances(instance_ids: [instance_id])
  wait_for_instance(ec2_client, :instance_running, instance_id)
  return true
rescue StandardError => e
  puts "Error restarting instance: #{e.message}"
  return false
end

# Attempts to reboot an Amazon Elastic Compute Cloud (Amazon EC2) instance.
#
# Prerequisites:
#
# - The Amazon EC2 instance.
#
# @param ec2_client [Aws::EC2::Client] An initialized EC2 client.
# @param instance_id [String] The ID of the instance.
# @return [Boolean] true if the instance was rebooted; otherwise, false.
# @example
#   exit 1 unless instance_rebooted?(
#     Aws::EC2::Client.new(region: 'us-east-1'),
#     'i-033c48ef067af3dEX'
#   )
def instance_rebooted?(ec2_client, instance_id)
  ec2_client.reboot_instances(instance_ids: [instance_id])
  wait_for_instance(ec2_client, :instance_status_ok, instance_id)
  return true
rescue StandardError => e
```

```

    puts "Error rebooting instance: #{e.message}"
    return false
  end

  # Attempts to enabled detailed monitoring for an
  # Amazon Elastic Compute Cloud (Amazon EC2) instance.
  #
  # Prerequisites:
  #
  # - The Amazon EC2 instance.
  #
  # @param ec2_client [Aws::EC2::Client] An initialized EC2 client.
  # @param instance_id [String] The ID of the instance.
  # @return [Boolean] true if detailed monitoring was enabled; otherwise, false.
  # @example
  #   exit 1 unless instance_detailed_monitoring_enabled?(
  #     Aws::EC2::Client.new(region: 'us-east-1'),
  #     'i-033c48ef067af3dEX'
  #   )
  def instance_detailed_monitoring_enabled?(ec2_client, instance_id)
    result = ec2_client.monitor_instances(instance_ids: [instance_id])
    puts "Detailed monitoring state: #{result.instance_monitorings[0].monitoring.state}"
    return true
  rescue Aws::EC2::Errors::InvalidState
    puts "The instance is not in a monitorable state. Skipping this step."
    return false
  rescue StandardError => e
    puts "Error enabling detailed monitoring: #{e.message}"
    return false
  end

  # Displays information about available
  # Amazon Elastic Compute Cloud (Amazon EC2) instances.
  #
  # @param ec2_client [Aws::EC2::Client] An initialized EC2 client.
  # @example
  #   list_instances_information(Aws::EC2::Client.new(region: 'us-east-1'))
  def list_instances_information(ec2_client)
    result = ec2_client.describe_instances
    result.reservations.each do |reservation|
      if reservation.instances.count.positive?
        reservation.instances.each do |instance|
          puts '-' * 12
          puts "Instance ID:           #{instance.instance_id}"
        end
      end
    end
  end

```

```

    puts "State:                #{instance.state.name}"
    puts "Image ID:              #{instance.image_id}"
    puts "Instance type:         #{instance.instance_type}"
    puts "Architecture:          #{instance.architecture}"
    puts "IAM instance profile ARN: #{instance.iam_instance_profile.arn}"
    puts "Key name:                 #{instance.key_name}"
    puts "Launch time:              #{instance.launch_time}"
    puts "Detailed monitoring state: #{instance.monitoring.state}"
    puts "Public IP address:         #{instance.public_ip_address}"
    puts "Public DNS name:           #{instance.public_dns_name}"
    puts "VPC ID:                    #{instance.vpc_id}"
    puts "Subnet ID:                 #{instance.subnet_id}"
    if instance.tags.count.positive?
      puts 'Tags:'
      instance.tags.each do |tag|
        puts "                #{tag.key}/#{tag.value}"
      end
    end
  end
end
end
end
end

# Full example call:
def run_me
  instance_id = ''
  region = ''
  # Print usage information and then stop.
  if ARGV[0] == '--help' || ARGV[0] == '-h'
    puts 'Usage:  ruby ec2-ruby-example-manage-instances.rb ' \
      'INSTANCE_ID REGION'
    puts 'Example: ruby ec2-ruby-example-manage-instances.rb ' \
      'i-033c48ef067af3dEX us-east-1'
    exit 1
  # If no values are specified at the command prompt, use these default values.
  elsif ARGV.count.zero?
    instance_id = 'i-033c48ef067af3dEX'
    region = 'us-east-1'
  # Otherwise, use the values as specified at the command prompt.
  else
    instance_id = ARGV[0]
    region = ARGV[1]
  end
end

```

```
ec2_client = Aws::EC2::Client.new(region: region)

puts 'Attempting to stop the instance. ' \
     'This might take a few minutes...'
unless instance_stopped?(ec2_client, instance_id)
  puts 'Cannot stop the instance. Skipping this step.'
end

puts "\nAttempting to restart the instance. " \
     'This might take a few minutes...'
unless instance_restarted?(ec2_client, instance_id)
  puts 'Cannot restart the instance. Skipping this step.'
end

puts "\nAttempting to reboot the instance. " \
     'This might take a few minutes...'
unless instance_rebooted?(ec2_client, instance_id)
  puts 'Cannot reboot the instance. Skipping this step.'
end

puts "\nAttempting to enable detailed monitoring for the instance..."
unless instance_detailed_monitoring_enabled?(ec2_client, instance_id)
  puts 'Cannot enable detailed monitoring for the instance. ' \
       'Skipping this step.'
end

puts "\nInformation about available instances:"
list_instances_information(ec2_client)
end

run_me if $PROGRAM_NAME == __FILE__
```

終止亞馬遜 EC2 實例

下列範例會嘗試終止指定的 Amazon EC2 執行個體。

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX - License - Identifier: Apache - 2.0

require 'aws-sdk-ec2'

# Attempts to terminate an Amazon Elastic Compute Cloud (Amazon EC2) instance.
#
```

```
# Prerequisites:
#
# - The Amazon EC2 instance.
#
# @param ec2_client [Aws::EC2::Client] An initialized EC2 client.
# @param instance_id [String] The ID of the instance.
# @return [Boolean] true if the instance was terminated; otherwise, false.
# @example
#   exit 1 unless instance_terminated?(
#     Aws::EC2::Client.new(region: 'us-east-1'),
#     'i-123abc'
#   )
def instance_terminated?(ec2_client, instance_id)
  response = ec2_client.describe_instance_status(instance_ids: [instance_id])

  if response.instance_statuses.count.positive? &&
    response.instance_statuses[0].instance_state.name == 'terminated'

    puts 'The instance is already terminated.'
    return true
  end

  ec2_client.terminate_instances(instance_ids: [instance_id])
  ec2_client.wait_until(:instance_terminated, instance_ids: [instance_id])
  puts 'Instance terminated.'
  return true
rescue StandardError => e
  puts "Error terminating instance: #{e.message}"
  return false
end

# Full example call:
def run_me
  instance_id = ''
  region = ''
  # Print usage information and then stop.
  if ARGV[0] == '--help' || ARGV[0] == '-h'
    puts 'Usage:  ruby ec2-ruby-example-terminate-instance-i-123abc.rb ' \
      'INSTANCE_ID REGION '
    puts 'Example: ruby ec2-ruby-example-terminate-instance-i-123abc.rb ' \
      'i-123abc us-east-1'
    exit 1
  # If no values are specified at the command prompt, use these default values.
  elsif ARGV.count.zero?
```



```
instance_id = 'i-123abc'
region = 'us-east-1'
# Otherwise, use the values as specified at the command prompt.
else
  instance_id = ARGV[0]
  region = ARGV[1]
end

ec2_client = Aws::EC2::Client.new(region: region)

puts "Attempting to terminate instance '#{instance_id}' " \
      '(this might take a few minutes)... '
unless instance_terminated?(ec2_client, instance_id)
  puts 'Could not terminate instance.'
end
end
end

run_me if $PROGRAM_NAME == __FILE__
```

取得 Amazon EC2 區域和可用區域的相關資訊

以下為範例示範：

1. 顯示可AWS 區域供您使用的 Amazon EC2 清單。
2. 根據 Amazon EC2 用戶端的情況，顯示可供您使用AWS 區域的 Amazon EC2 可用區域清單。

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX - License - Identifier: Apache - 2.0

require 'aws-sdk-ec2'

# Displays a list of AWS Regions for Amazon Elastic Compute Cloud (Amazon EC2)
# that are available to you.
#
# @param ec2_client [Aws::EC2::Client] An initialized EC2 client.
# @example
# list_regions_endpoints(Aws::EC2::Client.new(region: 'us-east-1'))
def list_regions_endpoints(ec2_client)
  result = ec2_client.describe_regions
  # Enable pretty printing.
  max_region_string_length = 16
  max_endpoint_string_length = 33
```

```
# Print header.
print 'Region'
print ' ' * (max_region_string_length - 'Region'.length)
print "  Endpoint\n"
print '-' * max_region_string_length
print ' '
print '-' * max_endpoint_string_length
print "\n"
# Print Regions and their endpoints.
result.regions.each do |region|
  print region.region_name.to_s
  print ' ' * (max_region_string_length - region.region_name.length)
  print ' '
  print region.endpoint.to_s
  print "\n"
end
end

# Displays a list of Amazon Elastic Compute Cloud (Amazon EC2)
# Availability Zones available to you depending on the AWS Region
# of the Amazon EC2 client.
#
# @param ec2_client [Aws::EC2::Client] An initialized EC2 client.
# @example
#   list_availability_zones(Aws::EC2::Client.new(region: 'us-east-1'))
def list_availability_zones(ec2_client)
  result = ec2_client.describe_availability_zones
  # Enable pretty printing.
  max_region_string_length = 16
  max_zone_string_length = 18
  max_state_string_length = 9
  # Print header.
  print 'Region'
  print ' ' * (max_region_string_length - 'Region'.length)
  print '  Zone'
  print ' ' * (max_zone_string_length - 'Zone'.length)
  print "  State\n"
  print '-' * max_region_string_length
  print ' '
  print '-' * max_zone_string_length
  print ' '
  print '-' * max_state_string_length
  print "\n"
  # Print Regions, Availability Zones, and their states.
```

```
result.availability_zones.each do |zone|
  print zone.region_name
  print ' ' * (max_region_string_length - zone.region_name.length)
  print ' '
  print zone.zone_name
  print ' ' * (max_zone_string_length - zone.zone_name.length)
  print ' '
  print zone.state
  # Print any messages for this Availability Zone.
  if zone.messages.count.positive?
    print "\n"
    puts ' Messages for this zone:'
    zone.messages.each do |message|
      print "   #{message.message}\n"
    end
  end
  print "\n"
end
end

# Full example call:
def run_me
  region = ''
  # Print usage information and then stop.
  if ARGV[0] == '--help' || ARGV[0] == '-h'
    puts 'Usage: ruby ec2-ruby-example-regions-availability-zones.rb REGION'
    puts 'Example: ruby ec2-ruby-example-regions-availability-zones.rb us-east-1'
    exit 1
  # If no values are specified at the command prompt, use these default values.
  elsif ARGV.count.zero?
    region = 'us-east-1'
  # Otherwise, use the values as specified at the command prompt.
  else
    region = ARGV[0]
  end

  ec2_client = Aws::EC2::Client.new(region: region)

  puts 'AWS Regions for Amazon EC2 that are available to you:'
  list_regions_endpoints(ec2_client)
  puts "\n\nAmazon EC2 Availability Zones that are available to you for AWS Region
'#{region}':"
  list_availability_zones(ec2_client)
end
```

```
run_me if $PROGRAM_NAME == __FILE__
```

AWS Elastic Beanstalk使用 AWS SDK for Ruby 範例

AWS Elastic Beanstalk可讓您在AWS雲端快速部署和管理應用程式，而不必擔心執行這些應用程式的基礎結構。您可以使用下面的例子來訪問 Elastic Beanstalk 使用 AWS SDK for Ruby。[如需有關 Elastic Beanstalk 的詳細資訊](#)，請參閱文件[AWS Elastic Beanstalk](#)。

主題

- [取得所有應用程式的相關資訊 AWS Elastic Beanstalk](#)
- [取得中特定應用程式的相關資訊 AWS Elastic Beanstalk](#)
- [更新紅寶石應用程序 AWS Elastic Beanstalk](#)

取得所有應用程式的相關資訊 AWS Elastic Beanstalk

下列範例會列出區域中所有 Elastic Beanstalk 應用程式的名稱、說明和 URL。us-west-2

```
# Copyright 2010-2019 Amazon.com, Inc. or its affiliates. All Rights Reserved.
#
# This file is licensed under the Apache License, Version 2.0 (the "License").
# You may not use this file except in compliance with the License. A copy of the
# License is located at
#
# http://aws.amazon.com/apache2.0/
#
# This file is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS
# OF ANY KIND, either express or implied. See the License for the specific
# language governing permissions and limitations under the License.

require 'aws-sdk-elasticbeanstalk' # v2: require 'aws-sdk'

eb = Aws::ElasticBeanstalk::Client.new(region: 'us-west-2')

eb.describe_applications.applications.each do |a|
  puts "Name:          #{a.application_name}"
  puts "Description:   #{a.description}"

  eb.describe_environments({application_name: a.application_name}).environments.each do
|env|
    puts " Environment:  #{env.environment_name}"
```

```
puts "    URL:      #{env.cname}"
puts "    Health:   #{env.health}"
end
end
```

取得中特定應用程式的相關資訊 AWS Elastic Beanstalk

下列範例會列出us-west-2區域中MyRailsApp應用程式的名稱、說明和 URL。

```
# Copyright 2010-2019 Amazon.com, Inc. or its affiliates. All Rights Reserved.
#
# This file is licensed under the Apache License, Version 2.0 (the "License").
# You may not use this file except in compliance with the License. A copy of the
# License is located at
#
# http://aws.amazon.com/apache2.0/
#
# This file is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS
# OF ANY KIND, either express or implied. See the License for the specific
# language governing permissions and limitations under the License.

require 'aws-sdk-elasticbeanstalk' # v2: require 'aws-sdk'

eb = Aws::ElasticBeanstalk::Client.new(region: 'us-west-2')

app = eb.describe_applications({application_names: [args[0]]})

if app.exists?
  puts "Name:      #{app.application_name}"
  puts "Description: #{app.description}"

  envs = eb.describe_environments({application_name: app.application_name})
  puts "URL:      #{envs.environments[0].cname}"
end
```

更新紅寶石應用程式 AWS Elastic Beanstalk

下列範例會更新us-west-2區域MyRailsApp中的 Ruby on Rails 應用程式。

Note

您必須位於 Rails 應用程式的根目錄才能成功運行腳本。

```
# Copyright 2010-2019 Amazon.com, Inc. or its affiliates. All Rights Reserved.
#
# This file is licensed under the Apache License, Version 2.0 (the "License").
# You may not use this file except in compliance with the License. A copy of the
# License is located at
#
# http://aws.amazon.com/apache2.0/
#
# This file is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS
# OF ANY KIND, either express or implied. See the License for the specific
# language governing permissions and limitations under the License.

require 'aws-sdk-elasticbeanstalk' # v2: require 'aws-sdk'

Aws.config.update({region: 'us-west-2'})

eb = Aws::ElasticBeanstalk::Client.new
s3 = Aws::S3::Client.new

app_name = 'MyRailsApp'

# Get S3 bucket containing app
app_versions = eb.describe_application_versions({ application_name: app_name })
av = app_versions.application_versions[0]
bucket = av.source_bundle.s3_bucket
s3_key = av.source_bundle.s3_key

# Get info on environment
envs = eb.describe_environments({ application_name: app_name })
env = envs.environments[0]
env_name = env.environment_name

# Create new storage location
resp = eb.create_storage_location()

puts "Created storage location in bucket #{resp.s3_bucket}"

s3.list_objects({
  prefix: s3_key,
  bucket: bucket
})

# Create ZIP file
```

```
zip_file_basename = SecureRandom.urlsafe_base64.to_s
zip_file_name = zip_file_basename + '.zip'

# Call out to OS to produce ZIP file
cmd = "git archive --format=zip -o #{zip_file_name} HEAD"
%x[ #{cmd} ]

# Get ZIP file contents
zip_contents = File.read(zip_file_name)

key = app_name + "\\\" + zip_file_name

s3.put_object({
  body: zip_contents,
  bucket: bucket,
  key: key
})

date = Time.new
today = date.day.to_s + "/" + date.month.to_s + "/" + date.year.to_s

eb.create_application_version({
  process: false,
  application_name: app_name,
  version_label: zip_file_basename,
  source_bundle: {
    s3_bucket: bucket,
    s3_key: key
  },
  description: "Updated #{today}"
})

eb.update_environment({
  environment_name: env_name,
  version_label: zip_file_basename
})
```

AWS Identity and Access Management(IAM) 使用 AWS SDK for Ruby 範例

AWS Identity and Access Management(IAM) 是一種 Web 服務，用於安全地控制AWS 服務。您可以使用下列範例，使用 Ruby 的 AWS SDK 存取 IAM。如需 IAM 的詳細資訊，請參閱 [IAM 文件](#)。

主題

- [取得 IAM 使用者的相關資訊](#)
- [列出身為管理員的 IAM 使用者](#)
- [新增新的 IAM 使用者](#)
- [建立 IAM 使用者的使用者存取金鑰](#)
- [將受管政策新增至 IAM 使用者](#)
- [建立 IAM 角色](#)
- [管理 IAM 使用者](#)
- [使用 IAM 政策](#)
- [管理 IAM 存取金鑰](#)
- [使用 IAM 伺服器憑證](#)
- [管理 IAM 帳戶別名](#)

取得 IAM 使用者的相關資訊

下列範例列出us-west-2區域中 IAM 使用者的群組、政策和存取金鑰 ID。如果有超過 100 個使用者，則iam.list_users.IsTruncated為 true，且iam.list_users.Marker包含可用來取得其他使用者相關資訊的值。如需詳細資訊，請參閱 [AWS:: IAM:: 用戶端主題](#)。

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX - License - Identifier: Apache - 2.0

require 'aws-sdk-iam'

# Displays information about available users in
# AWS Identity and Access Management (IAM) including users'
# names, associated group names, inline embedded user policy names,
# and access key IDs.
#
# @param iam_client [Aws::IAM::Client] An initialized IAM client.
# @example
#   get_user_details(Aws::IAM::Client.new)
def get_user_details(iam_client)
  users_response = iam_client.list_users

  if users_response.key?('users') && users_response.users.count.positive?

    # Are there more users available than can be displayed?
    if users_response.key?('is_truncated') && users_response.is_truncated
```



```
puts '(Note: not all users are displayed here, ' \
      "only the first #{users_response.users.count}.)"
else
  puts "Found #{users_response.users.count} user(s):"
end

users_response.users.each do |user|
  name = user.user_name
  puts '-' * 30
  puts "User name: #{name}"

  puts "Groups:"
  groups_response = iam_client.list_groups_for_user(user_name: name)
  if groups_response.key?('groups') &&
    groups_response.groups.count.positive?

    groups_response.groups.each do |group|
      puts "  #{group.group_name}"
    end
  else
    puts '  None'
  end

  puts 'Inline embedded user policies:'
  policies_response = iam_client.list_user_policies(user_name: name)
  if policies_response.key?('policy_names') &&
    policies_response.policy_names.count.positive?

    policies_response.policy_names.each do |policy_name|
      puts "  #{policy_name}"
    end
  else
    puts '  None'
  end

  puts 'Access keys:'
  access_keys_response = iam_client.list_access_keys(user_name: name)

  if access_keys_response.key?('access_key_metadata') &&
    access_keys_response.access_key_metadata.count.positive?

    access_keys_response.access_key_metadata.each do |access_key|
      puts "  #{access_key.access_key_id}"
    end
  end
end
```

```

    else
      puts ' None'
    end
  end
else
  puts 'No users found.'
end
rescue StandardError => e
  puts "Error getting user details: #{e.message}"
end

# Full example call:
def run_me
  iam_client = Aws::IAM::Client.new
  puts 'Attempting to get details for available users...'
  get_user_details(iam_client)
end

run_me if $PROGRAM_NAME == __FILE__

```

列出身為管理員的 IAM 使用者

下面的例子使用了 [get_account_授權IZON_詳細信息](#)，方法來獲取當前帳戶的用戶列表。

選擇Copy在本機儲存程式碼。

創建文件獲取管理員 .rb。

添加所需的 IAM 寶石和 OS 寶石，如果您在 Microsoft Windows 上運行，請使用後者使用捆綁的證書。

Note

Ruby 的 AWS SDK 版本 2 沒有特定於服務的寶石。

```

require 'aws-sdk-iam' # v2: require 'aws-sdk'
require 'os'

if OS.windows?
  Aws.use_bundled_cert!
end

```

建立方法以判斷使用者是否具有具有管理員權限的策略。

```
def user_has_admin_policy(user, admin_access)
  policies = user.user_policy_list

  policies.each do |p|
    if p.policy_name == admin_access
      return true
    end
  end

  false
end
```

建立方法以判斷使用者是否具有具有管理員權限的附加原則。

```
def user_has_attached_policy(user, admin_access)
  attached_policies = user.attached_managed_policies

  attached_policies.each do |p|
    if p.policy_name == admin_access
      return true
    end
  end

  false
end
```

建立方法，以判斷使用者所屬的群組是否具有具有管理員權限的原則。

建立方法，以判斷使用者所屬的群組是否具有具有管理員權限的附加原則。

```
def group_has_admin_policy(client, group, admin_access)
  resp = client.list_group_policies(
    group_name: group.group_name
  )

  resp.policy_names.each do |name|
    if name == admin_access
      return true
    end
  end
end
```

```
end

false
end
```

建立方法，以判斷使用者所屬的群組是否具有管理員權限。

```
def user_has_admin_from_group(client, user, admin_access)
  resp = client.list_groups_for_user(
    user_name: user.user_name
  )

  resp.groups.each do |group|
    has_admin_policy = group_has_admin_policy(client, group, admin_access)
    if has_admin_policy
      return true
    end

    has_attached_policy = group_has_attached_policy(client, group, admin_access)
    if has_attached_policy
      return true
    end
  end

  false
end
```

建立方法以判斷使用者是否具有管理員權限。

```
def is_user_admin(client, user, admin_access)
  has_admin_policy = user_has_admin_policy(user, admin_access)
  if has_admin_policy
    return true
  end

  has_attached_admin_policy = user_has_attached_policy(user, admin_access)
  if has_attached_admin_policy
    return true
  end

  has_admin_from_group = user_has_admin_from_group(client, user, admin_access)
  if has_admin_from_group
    return true
  end
end
```

```
end

false
end
```

創建一個方法來遍歷一個用戶列表，並返回有多少這些用戶具有管理員權限。

```
<code>
```

主要例程序從這裡開始。建立 IAM 用戶端和變數來儲存使用者數量、具有系統管理員權限的使用者數量，以及識別提供系統管理員權限之政策的字串。

```
def get_admin_count(client, users, admin_access)
  num_admins = 0

  users.each do |user|
    is_admin = is_user_admin(client, user, admin_access)
    if is_admin
      puts user.user_name
      num_admins += 1
    end
  end

  num_admins
end
```

呼叫 `get_account_authorization_details` 以獲取帳戶的詳細信息，並從中獲取該帳戶的用戶 `user_detail_list`。跟踪我們獲得的用戶數量，打電話 `get_admin_count` 以獲取具有管理員權限的用戶數量，並跟踪這些用戶的數量。

```
details = client.get_account_authorization_details(
  filter: ['User']
)

users = details.user_detail_list
num_users += users.count
more_admins = get_admin_count(client, users, access_admin)
num_admins += more_admins
```

如果第一次呼叫 `get_account_authorization_details` 未取得所有詳細資料，請再次呼叫它，然後重複確定有多少具有管理員權限的程序。

```
<code>
```

最後，顯示有多少使用者擁有管理員權限。

更多用戶 = 已截斷的詳細信息

而更多使用者

詳細資料 = 用戶端帳戶授權詳細資訊 (

過濾器：[用戶]，標記：詳細信息。標記

)

用戶 = 詳細信息

用戶 += 用戶。計數更多管理員 = 獲取管理員 (客戶端，用戶，訪問管理員) 數量管理員 += 更多管理員

更多用戶 = 已截斷的詳細信息

結束

請參閱 (詳見) 的[完整實例](#) GitHub。

新增新的 IAM 使用者

下列範例會使用密碼my_groovy_user在us-west-2區域中建立 IAM 使用者REPLACE_ME，並顯示使用者的帳戶 ID。如果具有該名稱的使用者已存在，則會顯示訊息且不會建立新使用者。

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX - License - Identifier: Apache - 2.0

require 'aws-sdk-iam'

# Creates a user in AWS Identity and Access Management (IAM).
#
# @param iam [Aws::IAM::Client] An initialized IAM client.
# @param user_name [String] The name of the user.
# @param initial_password [String] The initial password for the user.
# @return [String] The ID of the user if the user was created, otherwise;
#   the string 'Error'.
```

```
# @example
# puts create_user(Aws::IAM::Client.new, 'my-user', 'my-!p@55w0rd!')
def create_user(iam_client, user_name, initial_password)
  response = iam_client.create_user(user_name: user_name)
  iam_client.wait_until(:user_exists, user_name: user_name)
  iam_client.create_login_profile(
    password: initial_password,
    password_reset_required: true,
    user_name: user_name
  )
  return response.user.user_id
rescue Aws::IAM::Errors::EntityAlreadyExists
  puts "Error creating user '#{user_name}': user already exists."
  return 'Error'
rescue StandardError => e
  puts "Error creating user '#{user_name}': #{e.message}"
  return 'Error'
end

# Full example call:
def run_me
  user_name = 'my-user'
  initial_password = 'my-!p@55w0rd!'
  iam_client = Aws::IAM::Client.new

  puts "Attempting to create user '#{user_name}'..."
  user_id = create_user(iam_client, user_name, initial_password)

  if user_id == 'Error'
    puts 'User not created.'
  else
    puts "User '#{user_name}' created with ID '#{user_id}' and initial " \
      "sign-in password '#{initial_password}'."
  end
end

run_me if $PROGRAM_NAME == __FILE__
```

建立 IAM 使用者的使用者存取金鑰

下列範例會為us-west-2區域my_groovy_user中的 IAM 使用者建立存取金鑰和秘密金鑰。

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
```

```
# SPDX - License - Identifier: Apache - 2.0

require 'aws-sdk-iam'

# Creates an access key for a user in AWS Identity and Access Management (IAM).
#
# Prerequisites:
# - The user in IAM.
#
# @param iam [Aws::IAM::Client] An initialized IAM client.
# @param user_name [String] The name of the user.
# @example
#   create_access_key(Aws::IAM::Client.new, 'my-user')
def create_access_key(iam, user_name)
  response = iam.create_access_key(user_name: user_name)
  access_key = response.access_key
  puts 'Access key created:'
  puts "  Access key ID: #{access_key.access_key_id}"
  puts "  Secret access key: #{access_key.secret_access_key}"
  puts 'Keep a record of this information in a secure location. ' \
    'This will be the only time you will be able to view the ' \
    'secret access key.'
rescue Aws::IAM::Errors::LimitExceeded
  puts 'Error creating access key: limit exceeded. Cannot create any more. ' \
    'To create more, delete an existing access key, and then try again.'
rescue StandardError => e
  puts "Error creating access key: #{e.message}"
end

# Full example call:
def run_me
  iam = Aws::IAM::Client.new
  user_name = 'my-user'

  puts 'Attempting to create an access key...'
  create_access_key(iam, user_name)
end

run_me if $PROGRAM_NAME == __FILE__
```


將受管政策新增至 IAM 使用者

下列範例會將受管政策新增AmazonS3FullAccess至us-west-2區域my_groovy_user中的 IAM 使用者。

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX - License - Identifier: Apache - 2.0

require 'aws-sdk-iam'

# Attaches a policy to a user in AWS Identity and Access Management (IAM).
#
# Prerequisites:
# - The user in IAM.
#
# @param iam [Aws::IAM::Client] An initialized IAM client.
# @param user_name [String] The name of the user.
# @param policy_arn [String] The Amazon Resource Name (ARN) of the policy.
# @return [Boolean] true if the policy was attached; otherwise, false.
# @example
#   exit 1 unless alias_created?(
#     Aws::IAM::Client.new,
#     'my-user',
#     'arn:aws:iam::aws:policy/AmazonS3FullAccess'
#   )
def policy_attached_to_user?(iam_client, user_name, policy_arn)
  iam_client.attach_user_policy(
    user_name: user_name,
    policy_arn: policy_arn
  )
  return true
rescue StandardError => e
  puts "Error attaching policy to user: #{e.message}"
  return false
end

# Full example call:
def run_me
  user_name = 'my-user'
  arn_prefix = 'arn:aws:iam::aws:policy/'
  policy_arn = arn_prefix + 'AmazonS3FullAccess'
  iam_client = Aws::IAM::Client.new
```

```
puts "Attempting to attach policy with ARN '#{policy_arn}' to " \
     "user '#{user_name}'..."

if policy_attached_to_user?(iam_client, user_name, policy_arn)
  puts 'Policy attached.'
else
  puts 'Policy not attached.'
end
end

run_me if $PROGRAM_NAME == __FILE__
```

建立 IAM 角色

下列範例會建立角色，`my_groovy_role`讓 Amazon EC2 可以存取該區域中的 Amazon S3 和亞馬遜 DynamoDB。us-west-2

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX - License - Identifier: Apache - 2.0

require 'aws-sdk-iam'

# Creates a role in AWS Access and Identity Management (IAM).
#
# @param iam_client [Aws::IAM::Client] An initialized IAM client.
# @param role_name [String] A name for the role.
# @param assume_role_policy_document [String]
# @param policy_arns [Array] An array of type String representing
#   Amazon Resource Names (ARNs) corresponding to available
#   IAM managed policies.
# @return [String] The ARN of the new role; otherwise, the string 'Error'.
# @example
#   puts create_role(
#     Aws::IAM::Client.new,
#     'my-ec2-s3-dynamodb-full-access-role',
#     {
#       Version: '2012-10-17',
#       Statement: [
#         {
#           Effect: 'Allow',
#           Principal: {
#             Service: 'ec2.amazonaws.com'
#           },
```

```
#       Action: 'sts:AssumeRole'
#     }
#   ]
# },
# [
#   'arn:aws:iam::aws:policy/AmazonS3FullAccess',
#   'arn:aws:iam::aws:policy/AmazonDynamoDBFullAccess'
# ]
# )
def create_role(
  iam_client,
  role_name,
  assume_role_policy_document,
  policy_arns
)
  iam_client.create_role(
    role_name: role_name,
    assume_role_policy_document: assume_role_policy_document.to_json
  )
  policy_arns.each do |policy_arn|
    iam_client.attach_role_policy(
      policy_arn: policy_arn,
      role_name: role_name,
    )
  end
  return iam_client.get_role(role_name: role_name).role.arn
rescue StandardError => e
  puts "Error creating role: #{e.message}"
  return 'Error'
end

# Full example call:
def run_me
  role_name = 'my-ec2-s3-dynamodb-full-access-role'

  # Allow the role to trust Amazon Elastic Compute Cloud (Amazon EC2)
  # within the AWS account.
  assume_role_policy_document = {
    Version: '2012-10-17',
    Statement: [
      {
        Effect: 'Allow',
        Principal: {
          Service: 'ec2.amazonaws.com'
```

```
    },
    Action: 'sts:AssumeRole'
  }
]
}

# Allow the role to take all actions within
# Amazon Simple Storage Service (Amazon S3)
# and Amazon DynamoDB across the AWS account.
policy_arns = [
  'arn:aws:iam::aws:policy/AmazonS3FullAccess',
  'arn:aws:iam::aws:policy/AmazonDynamoDBFullAccess'
]

iam_client = Aws::IAM::Client.new

puts "Attempting to create the role named '#{role_name}'..."

role_arn = create_role(
  iam_client,
  role_name,
  assume_role_policy_document,
  policy_arns
)

if role_arn == 'Error'
  puts 'Could not create role.'
else
  puts "Role created with ARN '#{role_arn}'."
end
end

run_me if $PROGRAM_NAME == __FILE__
```

管理 IAM 使用者

IAM 使用者代表與AWS之互動的個人或服務。如需 IAM 使用者的詳細資訊，請參閱 [IAM 使用者](#)。

在此範例中，您可以使用適用 AWS SDK for Ruby 搭配 IAM：

1. 使用 `Aws::IAM::Client#list_users` 取得有關可用 IAM 使用者的資訊。
2. 使用 `Aws::IAM::Client#create_user` 建立使用者。
3. 使用 `Aws::IAM::Client#update_user` 更新使用者名稱。

4. 使用 `Aws::IAM::Client#delete_user` 刪除該用戶。

先決條件

在執行範例程式碼之前，您必須先安裝並設定 AWS SDK for Ruby，如下所述：

- [安裝 AWS SDK for Ruby](#)
- [為紅寶石配置 AWS SDK](#)

範例

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX - License - Identifier: Apache - 2.0

# The following code example shows how to to:
# 1. Get a list of user names in AWS Identity and Access Management (IAM).
# 2. Create a user.
# 3. Update the user's name.
# 4. Delete the user.

require 'aws-sdk-iam'

# Gets a list of available user names in
# AWS Identity and Access Management (IAM).
#
# @param iam_client [Aws::IAM::Client] An initialized IAM client.
# @example
#   list_user_names(Aws::IAM::Client.new)
def list_user_names(iam_client)
  response = iam_client.list_users
  if response.key?('users') && response.users.count.positive?
    response.users.each do |user|
      puts user.user_name
    end
  else
    puts 'No users found.'
  end
end
rescue StandardError => e
  puts "Error listing user names: #{e.message}"
end

# Creates a user in AWS Identity and Access Management (IAM).
```

```
#
# @param iam_client [Aws::IAM::Client] An initialized IAM client.
# @param user_name [String] The name of the new user.
# @return [Boolean] true if the user was created; otherwise, false.
# @example
#   exit 1 unless user_created?(Aws::IAM::Client.new, 'my-user')
def user_created?(iam_client, user_name)
  iam_client.create_user(user_name: user_name)
  return true
rescue Aws::IAM::Errors::EntityAlreadyExists
  puts "Error creating user: user '#{user_name}' already exists."
  return false
rescue StandardError => e
  puts "Error creating user: #{e.message}"
  return false
end

# Changes the name of a user in AWS Identity and Access Management (IAM).
#
# Prerequisites:
# - The user in IAM.
#
# @param iam_client [Aws::IAM::Client] An initialized IAM client.
# @param user_current_name [String] The current name of the user.
# @param user_new_name [String] The new name for the user.
# @return [Boolean] true if the name of the user was changed;
#   otherwise, false.
# @example
#   exit 1 unless user_name_changed?(
#     Aws::IAM::Client.new,
#     'my-user',
#     'my-changed-user'
#   )
def user_name_changed?(iam_client, user_current_name, user_new_name)
  iam_client.update_user(
    user_name: user_current_name,
    new_user_name: user_new_name
  )
  return true
rescue StandardError => e
  puts "Error updating user name: #{e.message}"
  return false
end
```

```
# Deletes a user in AWS Identity and Access Management (IAM).
#
# Prerequisites:
# - The user in IAM.
#
# @param iam_client [Aws::IAM::Client] An initialized IAM client.
# @param user_name [String] The name of the user.
# @return [Boolean] true if the user was deleted; otherwise, false.
# @example
#   exit 1 unless user_deleted?(Aws::IAM::Client.new, 'my-user')
def user_deleted?(iam_client, user_name)
  iam_client.delete_user(user_name: user_name)
  return true
rescue StandardError => e
  puts "Error deleting user: #{e.message}"
  return false
end

# Full example call:
def run_me
  user_name = 'my-user'
  user_changed_name = 'my-changed-user'
  delete_user = true
  iam_client = Aws::IAM::Client.new

  puts "Initial user names are:\n\n"
  list_user_names(iam_client)

  puts "\nAttempting to create user '#{user_name}'..."

  if user_created?(iam_client, user_name)
    puts 'User created.'
  else
    puts 'Could not create user. Stopping program.'
    exit 1
  end

  puts "User names now are:\n\n"
  list_user_names(iam_client)

  puts "\nAttempting to change the name of the user '#{user_name}' " \
    "to '#{user_changed_name}'..."

  if user_name_changed?(iam_client, user_name, user_changed_name)
```

```
puts 'User name changed.'
puts "User names now are:\n\n"
list_user_names(iam_client)

if delete_user
  # Delete user with changed name.
  puts "\nAttempting to delete user '#{user_changed_name}'..."

  if user_deleted?(iam_client, user_changed_name)
    puts 'User deleted.'
  else
    puts 'Could not delete user. You must delete the user yourself.'
  end

  puts "User names now are:\n\n"
  list_user_names(iam_client)
end
else
  puts 'Could not change user name.'
  puts "User names now are:\n\n"
  list_user_names(iam_client)

  if delete_user
    # Delete user with initial name.
    puts "\nAttempting to delete user '#{user_name}'..."

    if user_deleted?(iam_client, user_name)
      puts 'User deleted.'
    else
      puts 'Could not delete user. You must delete the user yourself.'
    end

    puts "User names now are:\n\n"
    list_user_names(iam_client)
  end
end
end

run_me if $PROGRAM_NAME == __FILE__
```

使用 IAM 政策

IAM 政策是指定一或多個許可的文件。如需 IAM 政策的詳細資訊，請參閱 [IAM 政策概觀](#)。

在此範例中，您可以使用適用 AWS SDK for Ruby 搭配 IAM：

1. 使用 [AWS::IAM::用戶端 #create_policy](#) 建立政策。
2. 使用 [AWS::IAM::用戶端 #get_policy](#) 取得有關政策的資訊。
3. 使用 [AWS::IAM::客戶端 #attach_role_policy](#) 將政策附加到角色。
4. 使用 [AWS::IAM::用戶端 #list_attached_role_policies](#) 列出附加至該角色的政策。
5. 使用 [AWS::IAM::用戶端 #detach_role_policy](#) 將政策與角色分離。

先決條件

在執行範例程式碼之前，您必須先安裝並設定 AWS SDK for Ruby，如下所述：

- [安裝 AWS SDK for Ruby](#)
- [為紅寶石配置 AWS SDK](#)

您還需要創建腳本中指定的角色 (my-role)。您可以在 IAM 主控台中執行此操作。

範例

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX - License - Identifier: Apache - 2.0

# The following code example shows how to:
# 1. Create a policy in AWS Identity and Access Management (IAM).
# 2. Attach the policy to a role.
# 3. List the policies that are attached to the role.
# 4. Detach the policy from the role.

require 'aws-sdk-iam'

# Creates a policy in AWS Identity and Access Management (IAM).
#
# @param iam_client [Aws::IAM::Client] An initialized IAM client.
# @param policy_name [String] A name for the policy.
# @param policy_document [Hash] The policy definition.
# @return [String] The new policy's Amazon Resource Name (ARN);
# otherwise, the string 'Error'.
# @example
# puts create_policy(
#   Aws::IAM::Client.new,
```

```

#   'my-policy',
#   {
#     'Version': '2012-10-17',
#     'Statement': [
#       {
#         'Effect': 'Allow',
#         'Action': 's3:ListAllMyBuckets',
#         'Resource': 'arn:aws:s3:::*'
#       }
#     ]
#   }
# )
def create_policy(iam_client, policy_name, policy_document)
  response = iam_client.create_policy(
    policy_name: policy_name,
    policy_document: policy_document.to_json
  )
  return response.policy.arn
rescue StandardError => e
  puts "Error creating policy: #{e.message}"
  return 'Error'
end

# Attaches a policy to a role in AWS Identity and Access Management (IAM).
#
# Prerequisites:
# - An existing role.
#
# @param iam_client [Aws::IAM::Client] An initialized IAM client.
# @param role_name [String] The name of the role to attach the policy to.
# @param policy_arn [String] The policy's Amazon Resource Name (ARN).
# @return [Boolean] True if the policy was attached to the role;
#   otherwise, false.
# @example
#   exit 1 unless policy_attached_to_role?(
#     Aws::IAM::Client.new,
#     'my-role',
#     'arn:aws:iam::111111111111:policy/my-policy'
#   )
def policy_attached_to_role?(iam_client, role_name, policy_arn)
  iam_client.attach_role_policy(role_name: role_name, policy_arn: policy_arn)
  return true
rescue StandardError => e
  puts "Error attaching policy to role: #{e.message}"

```

```

    return false
  end

  # Displays a list of policy Amazon Resource Names (ARNs) that are attached to a
  # role in AWS Identity and Access Management (IAM).
  #
  # Prerequisites:
  # - An existing role.
  #
  # @param iam_client [Aws::IAM::Client] An initialized IAM client.
  # @param role_name [String] The name of the role.
  # @example
  #   list_policy_arns_attached_to_role(Aws::IAM::Client.new, 'my-role')
  def list_policy_arns_attached_to_role(iam_client, role_name)
    response = iam_client.list_attached_role_policies(role_name: role_name)
    if response.key?('attached_policies') && response.attached_policies.count.positive?
      response.attached_policies.each do |attached_policy|
        puts "  #{attached_policy.policy_arn}"
      end
    else
      puts 'No policies attached to role.'
    end
  end
rescue StandardError => e
  puts "Error checking for policies attached to role: #{e.message}"
end

# Detaches a policy from a role in AWS Identity and Access Management (IAM).
#
# Prerequisites:
# - An existing role with an attached policy.
#
# @param iam_client [Aws::IAM::Client] An initialized IAM client.
# @param role_name [String] The name of the role to detach the policy from.
# @param policy_arn [String] The policy's Amazon Resource Name (ARN).
# @return [Boolean] True if the policy was detached from the role;
#   otherwise, false.
# @example
#   exit 1 unless policy_detached_from_role?(
#     Aws::IAM::Client.new,
#     'my-role',
#     'arn:aws:iam::111111111111:policy/my-policy'
#   )
  def policy_detached_from_role?(iam_client, role_name, policy_arn)
    iam_client.detach_role_policy(role_name: role_name, policy_arn: policy_arn)
  end
end

```

```
    return true
  rescue StandardError => e
    puts "Error detaching policy from role: #{e.message}"
    return false
  end

# Full example call:
def run_me
  role_name = 'my-role'
  policy_name = 'my-policy'

  # Allows the caller to get a list of all buckets in
  # Amazon Simple Storage Service (Amazon S3) that are owned by the caller.
  policy_document = {
    'Version': '2012-10-17',
    'Statement': [
      {
        'Effect': 'Allow',
        'Action': 's3:ListAllMyBuckets',
        'Resource': 'arn:aws:s3:::*'
      }
    ]
  }

  detach_policy_from_role = true
  iam_client = Aws::IAM::Client.new

  puts "Attempting to create policy '#{policy_name}'..."
  policy_arn = create_policy(iam_client, policy_name, policy_document)

  if policy_arn == 'Error'
    puts 'Could not create policy. Stopping program.'
    exit 1
  else
    puts 'Policy created.'
  end

  puts "Attempting to attach policy '#{policy_name}' " \
    "to role '#{role_name}'..."

  if policy_attached_to_role?(iam_client, role_name, policy_arn)
    puts 'Policy attached.'
  else
    puts 'Could not attach policy to role.'
```

```
detach_policy_from_role = false
end

puts "Policy ARNs attached to role '#{role_name}':"
list_policy_arns_attached_to_role(iam_client, role_name)

if detach_policy_from_role
  puts "Attempting to detach policy '#{policy_name}' " \
    "from role '#{role_name}'..."

  if policy_detached_from_role?(iam_client, role_name, policy_arn)
    puts 'Policy detached.'
  else
    puts 'Could not detach policy from role. You must detach it yourself.'
  end
end

end
end

run_me if $PROGRAM_NAME == __FILE__
```

管理 IAM 存取金鑰

使用者需要自己的存取金鑰，才能AWS從 AWS SDK for Ruby 進程式設計呼叫。為了滿足這個需求，您可以為 IAM 使用者建立、修改、查看或輪換存取金鑰 (存取金鑰 ID 和私密存取金鑰)。在預設情況下，您建立存取金鑰時，其狀態為「作用中」。這表示使用者可以使用存取金鑰進行 API 呼叫。如需存取金鑰的詳細資訊，請參閱[管理 IAM 使用者的存取金鑰](#)。

在此範例中，您可以使用適用 AWS SDK for Ruby 搭配 IAM：

1. 使用 [AWS:: AWS IAM:: 客戶端 #list_access_keys](#) 列出 IAM 使用者存取金鑰。
2. 使用 [AWS:: IAM:: 用戶端 #create_access_key](#) 建立存取金鑰。
3. 使用 [Aws:: IAM:: 客戶端 #get_access_key_last_used](#) 確定上次使用存取金鑰的時間。
4. 使用 [AWS:: IAM:: 用戶端 #update_access_key](#) 停用存取金鑰。
5. 使用 [AWS:: IAM:: 客戶端 #delete_access_key](#) 刪除存取金鑰。

先決條件

在執行範例程式碼之前，您必須先安裝並設定 AWS SDK for Ruby，如下所述：

- [安裝 AWS SDK for Ruby](#)

- [為紅寶石配置 AWS SDK](#)

您還需要創建腳本中指定的用戶 (my-user)。您可以在 IAM 主控台中建立新的 IAM 使用者，或以程式設計方式建立新的 IAM 使用者，如[新增新的 IAM 使用者](#)所示。

範例

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX - License - Identifier: Apache - 2.0

# This code example demonstrates how to:
# 1. List access keys for a user in AWS Identity and Access Management (IAM).
# 2. Create an access key for a user.
# 3. Determine when a user's access keys were last used.
# 4. Deactivate an access key for a user.
# 5. Delete an access key for a user.

require 'aws-sdk-iam'

# Lists information about access keys for a user in
# AWS Identity and Access Management (IAM).
#
# Prerequisites:
# - The user in IAM.
#
# @param iam [Aws::IAM::Client] An initialized IAM client.
# @param user_name [String] The name of the user.
# @example
# puts list_access_keys(Aws::IAM::Client.new, 'my-user')
def list_access_keys(iam, user_name)
  response = iam.list_access_keys(user_name: user_name)

  if response.access_key_metadata.count.positive?
    puts 'Access key IDs:'
    response.access_key_metadata.each do |key_metadata|
      puts "  #{key_metadata.access_key_id}"
    end
  else
    puts "No access keys found for user '#{user_name}'."
  end
end

rescue Aws::IAM::Errors::NoSuchEntity
  puts "Error listing access keys: cannot find user '#{user_name}'."
end

exit 1
```

```

rescue StandardError => e
  puts "Error listing access keys: #{e.message}"
end

# Creates an access key for a user in AWS Identity and Access Management (IAM).
#
# Prerequisites:
# - The user in IAM.
#
# @param iam [Aws::IAM::Client] An initialized IAM client.
# @param user_name [String] The name of the user.
# @return [Aws::IAM::Types::AccessKey] Information about the new access key;
# otherwise, the string 'Error'.
# @example
# puts create_access_key(Aws::IAM::Client.new, 'my-user')
def create_access_key(iam, user_name)
  response = iam.create_access_key(user_name: user_name)
  access_key = response.access_key
  puts 'Access key created:'
  puts " Access key ID: #{access_key.access_key_id}"
  puts " Secret access key: #{access_key.secret_access_key}"
  puts 'Keep a record of this information in a secure location. ' \
    'This will be the only time you will be able to view the ' \
    'secret access key.'
  return access_key
rescue Aws::IAM::Errors::LimitExceeded
  puts 'Error creating access key: limit exceeded. Cannot create any more. ' \
    'To create more, delete an existing access key, and then try again.'
  return 'Error'
rescue StandardError => e
  puts "Error creating access key: #{e.message}"
  return 'Error'
end

# Lists information about when access keys for a user in
# AWS Identity and Access Management (IAM) were last used.
#
# Prerequisites:
# - The user in IAM.
#
# @param iam [Aws::IAM::Client] An initialized IAM client.
# @param user_name [String] The name of the user.
# @example
# puts access_keys_last_used(Aws::IAM::Client.new, 'my-user')

```

```

def access_keys_last_used(iam, user_name)
  response = iam.list_access_keys(user_name: user_name)

  response.access_key_metadata.each do |key_metadata|
    last_used = iam.get_access_key_last_used(access_key_id: key_metadata.access_key_id)
    if last_used.access_key_last_used.last_used_date.nil?
      puts "  Key '#{key_metadata.access_key_id}' not used or date undetermined."
    else
      puts "  Key '#{key_metadata.access_key_id}' last used on " \
        "#{last_used.access_key_last_used.last_used_date}"
    end
  end
end

rescue StandardError => e
  puts "Error determining when access keys were last used: #{e.message}"
end

# Deactivates an access key in AWS Identity and Access Management (IAM).
#
# Prerequisites:
# - A user in IAM.
# - An access key for that user.
#
# @param iam [Aws::IAM::Client] An initialized IAM client.
# @param user_name [String] The name of the user.
# @param access_key_id [String] The ID of the access key.
# @return [Boolean] true if the access key was deactivated;
#   otherwise, false.
# @example
#   exit 1 unless access_key_deactivated?(
#     Aws::IAM::Client.new,
#     'my-user',
#     'AKIAIOSFODNN7EXAMPLE'
#   )
def access_key_deactivated?(iam, user_name, access_key_id)
  iam.update_access_key(
    user_name: user_name,
    access_key_id: access_key_id,
    status: 'Inactive'
  )
  return true
rescue StandardError => e
  puts "Error deactivating access key: #{e.message}"
  return false
end

```



```
# Deletes an access key in AWS Identity and Access Management (IAM).
#
# Prerequisites:
# - A user in IAM.
# - An access key for that user.
#
# @param iam [Aws::IAM::Client] An initialized IAM client.
# @param user_name [String] The name of the user.
# @param access_key_id [String] The ID of the access key.
# @return [Boolean] true if the access key was deleted;
#   otherwise, false.
# @example
#   exit 1 unless access_key_deleted?(
#     Aws::IAM::Client.new,
#     'my-user',
#     'AKIAIOSFODNN7EXAMPLE'
#   )
def access_key_deleted?(iam, user_name, access_key_id)
  iam.delete_access_key(
    user_name: user_name,
    access_key_id: access_key_id
  )
  return true
rescue StandardError => e
  puts "Error deleting access key: #{e.message}"
  return false
end

# Full example call:
def run_me
  iam = Aws::IAM::Client.new
  user_name = 'my-user'
  create_key = true # Set to false to not create a new access key.
  delete_key = true # Set to false to not delete any generated access key.

  puts "Access keys for user '#{user_name}' before attempting to create an " \
    'additional access key for the user:'
  list_access_keys(iam, user_name)

  access_key = ''

  if create_key
    puts 'Attempting to create an additional access key...'
```

```
access_key = create_access_key(iam, user_name)

if access_key == 'Error'
  puts 'Additional access key not created. Stopping program.'
  exit 1
end

puts 'Additional access key created. Access keys for user now are:'
list_access_keys(iam, user_name)
end

puts 'Determining when current access keys were last used...'
access_keys_last_used(iam, user_name)

if create_key && delete_key
  puts 'Attempting to deactivate additional access key...'

  if access_key_deactivated?(iam, user_name, access_key.access_key_id)
    puts 'Access key deactivated. Access keys for user now are:'
    list_access_keys(iam, user_name)
  else
    puts 'Access key not deactivated. Stopping program.'
    puts 'You will need to delete the access key yourself.'
  end

  puts 'Attempting to delete additional access key...'

  if access_key_deleted?(iam, user_name, access_key.access_key_id)
    puts 'Access key deleted. Access keys for user now are:'
    list_access_keys(iam, user_name)
  else
    puts 'Access key not deleted. You will need to delete the ' \
        'access key yourself.'
  end
end
end

run_me if $PROGRAM_NAME == __FILE__
```

使用 IAM 伺服器憑證

若要在上啟用 HTTPS 連線到您的網站或應用程式AWS，您需要 SSL/TLS 伺服器憑證。若要將您從外部供應商取得的憑證與網站或應用程式搭配使用AWS，您必須將憑證上傳至 IAM，或將其匯入 Cer AWS tificate Manager。如需有關伺服器憑證的詳細資訊，請參閱[使用伺服器憑證](#)。

在此範例中，您可以使用適用 AWS SDK for Ruby 搭配 IAM：

1. 使用 `Aws::IAM::客戶端 #update_server_certificate` 更新伺服器憑證。
2. 使用 `AWS::IAM::客戶端 #delete_server_certificate` 刪除伺服器憑證。
3. 使用 `AWS::IAM::用戶端 #list_server_certificates` 列出有關任何剩餘伺服器憑證的資訊。

先決條件

在執行範例程式碼之前，您必須先安裝並設定 AWS SDK for Ruby，如下所述：

- [安裝 AWS SDK for Ruby](#)
- [為紅寶石配置 AWS SDK](#)

Note

伺服器證書必須已經存在，否則腳本將拋出一個 `Aws::IAM::錯誤::NoSuchEntity` 錯誤。

範例

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX - License - Identifier: Apache - 2.0

# The following code example shows how to:
# 1. Update a server certificate in AWS Identity and Access Management (IAM).
# 2. List the names of available server certificates.
# 3. Delete a server certificate.

require 'aws-sdk-iam'

# Gets a list of available server certificate names in
# AWS Identity and Access Management (IAM).
#
# @param iam_client [Aws::IAM::Client] An initialized IAM client.
```

```
# @example
# list_server_certificate_names(Aws::IAM::Client.new)
def list_server_certificate_names(iam_client)
  response = iam_client.list_server_certificates

  if response.key?('server_certificate_metadata_list') &&
    response.server_certificate_metadata_list.count.positive?

    response.server_certificate_metadata_list.each do |certificate_metadata|
      puts certificate_metadata.server_certificate_name
    end
  else
    puts 'No server certificates found. Stopping program.'
    exit 1
  end
rescue StandardError => e
  puts "Error getting server certificate names: #{e.message}"
end

# Changes the name of a server certificate in
# AWS Identity and Access Management (IAM).
#
# Prerequisites:
#
# - The server certificate in IAM.
#
# @param iam_client [Aws::IAM::Client] An initialized IAM client.
# @param server_certificate_current_name [String] The current name of
# the server certificate.
# @param server_certificate_new_name [String] The new name for the
# the server certificate.
# @return [Boolean] true if the name of the server certificate
# was changed; otherwise, false.
# @example
# exit 1 unless server_certificate_name_changed?(
#   Aws::IAM::Client.new,
#   'my-server-certificate',
#   'my-changed-server-certificate'
# )
def server_certificate_name_changed?(
  iam_client,
  server_certificate_current_name,
  server_certificate_new_name
)
```

```
iam_client.update_server_certificate(  
  server_certificate_name: server_certificate_current_name,  
  new_server_certificate_name: server_certificate_new_name  
)  
return true  
rescue StandardError => e  
  puts "Error updating server certificate name: #{e.message}"  
  return false  
end  
  
# Deletes a server certificate in  
# AWS Identity and Access Management (IAM).  
#  
# Prerequisites:  
#  
# - The server certificate in IAM.  
#  
# @param iam_client [Aws::IAM::Client] An initialized IAM client.  
# @param server_certificate_name [String] The name of the server certificate.  
# @return [Boolean] true if the server certificate was deleted;  
#   otherwise, false.  
# @example  
#   exit 1 unless server_certificate_deleted?(  
#     Aws::IAM::Client.new,  
#     'my-server-certificate'  
#   )  
def server_certificate_deleted?(iam_client, server_certificate_name)  
  iam_client.delete_server_certificate(  
    server_certificate_name: server_certificate_name  
  )  
  return true  
rescue StandardError => e  
  puts "Error deleting server certificate: #{e.message}"  
  return false  
end  
  
# Full example call:  
def run_me  
  server_certificate_name = 'my-server-certificate'  
  server_certificate_changed_name = 'my-changed-server-certificate'  
  delete_server_certificate = true  
  iam_client = Aws::IAM::Client.new  
  
  puts "Initial server certificate names are:\n\n"
```

```
list_server_certificate_names(iam_client)

puts "\nAttempting to change name of server certificate " \
     " '#{server_certificate_name}' " \
     "to '#{server_certificate_changed_name}'..."

if server_certificate_name_changed?(
  iam_client,
  server_certificate_name,
  server_certificate_changed_name
)
  puts 'Server certificate name changed.'
  puts "Server certificate names now are:\n\n"
  list_server_certificate_names(iam_client)

  if delete_server_certificate
    # Delete server certificate with changed name.
    puts "\nAttempting to delete server certificate " \
         "'#{server_certificate_changed_name}'..."

    if server_certificate_deleted?(iam_client, server_certificate_changed_name)
      puts 'Server certificate deleted.'
    else
      puts 'Could not delete server certificate. You must delete it yourself.'
    end

    puts "Server certificate names now are:\n\n"
    list_server_certificate_names(iam_client)
  end
else
  puts 'Could not change server certificate name.'
  puts "Server certificate names now are:\n\n"
  list_server_certificate_names(iam_client)

  if delete_server_certificate
    # Delete server certificate with initial name.
    puts "\nAttempting to delete server certificate '#{server_certificate_name}'..."

    if server_certificate_deleted?(iam_client, server_certificate_name)
      puts 'Server certificate deleted.'
    else
      puts 'Could not delete server certificate. You must delete it yourself.'
    end
  end
end
```

```
        puts "Server certificate names now are:\n\n"  
        list_server_certificate_names(iam_client)  
    end  
end  
end  
  
run_me if $PROGRAM_NAME == __FILE__
```

管理 IAM 帳戶別名

如果您希望登入頁面的 URL 包含您的公司名稱或其他易記識別碼，而不是您的AWS帳戶 ID，您可以為您的帳戶 ID 建立 IAM AWS 帳戶別名。如果您建立 IAM 帳戶別名，您的登入頁面 URL 會變更為合併別名。如需 IAM 帳戶別名的詳細資訊，請參閱[您的AWS帳戶 ID 及其別名](#)。

在此範例中，您可以使用適用 AWS SDK for Ruby 搭配 IAM：

1. 使用 [AWS::IAM::用AWS戶端 #list_account_aliases](#) 列出帳戶別名。
2. 使用 [AWS::IAM::用戶端 #create_account_alias](#) 建立帳戶別名。
3. 使用 [AWS::IAM::客戶端 #delete_account_alias](#) 刪除帳戶別名。

先決條件

在執行範例程式碼之前，您必須先安裝並設定 AWS SDK for Ruby，如下所述：

- [安裝 AWS SDK for Ruby](#)
- [為紅寶石配置 AWS SDK](#)

在範例程式碼中，將字串my-account-alias串變更為在所有 Amazon Web Services 產品中都是唯一的字串。

範例

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.  
# SPDX - License - Identifier: Apache - 2.0  
  
# The following code example shows how to:  
# 1. List available AWS account aliases.  
# 2. Create an account alias.  
# 3. Delete an account alias.
```

```
require 'aws-sdk-iam'

# Lists available AWS account aliases.
#
# @param iam [Aws::IAM::Client] An initialized IAM client.
# @example
#   puts list_aliases(Aws::IAM::Client.new)
def list_aliases(iam)
  response = iam.list_account_aliases

  if response.account_aliases.count.positive?
    response.account_aliases.each do |account_alias|
      puts " #{account_alias}"
    end
  else
    puts 'No account aliases found.'
  end
end

rescue StandardError => e
  puts "Error listing account aliases: #{e.message}"
end

# Creates an AWS account alias.
#
# @param iam [Aws::IAM::Client] An initialized IAM client.
# @param account_alias [String] The name of the account alias to create.
# @return [Boolean] true if the account alias was created; otherwise, false.
# @example
#   exit 1 unless alias_created?(Aws::IAM::Client.new, 'my-account-alias')
def alias_created?(iam, account_alias)
  iam.create_account_alias(account_alias: account_alias)
  return true
end

rescue StandardError => e
  puts "Error creating account alias: #{e.message}"
  return false
end

# Deletes an AWS account alias.
#
# @param iam [Aws::IAM::Client] An initialized IAM client.
# @param account_alias [String] The name of the account alias to delete.
# @return [Boolean] true if the account alias was deleted; otherwise, false.
# @example
#   exit 1 unless alias_deleted?(Aws::IAM::Client.new, 'my-account-alias')
def alias_deleted?(iam, account_alias)
```



```
iam.delete_account_alias(account_alias: account_alias)
return true
rescue StandardError => e
  puts "Error deleting account alias: #{e.message}"
  return false
end

# Full example call:
def run_me
  iam = Aws::IAM::Client.new
  account_alias = 'my-account-alias'
  create_alias = true # Change to false to not generate an account alias.
  delete_alias = true # Change to false to not delete any generated account alias.

  puts 'Account aliases are:'
  list_aliases(iam)

  if create_alias
    puts 'Attempting to create account alias...'
    if alias_created?(iam, account_alias)
      puts 'Account alias created. Account aliases now are:'
      list_aliases(iam)
    else
      puts 'Account alias not created. Stopping program.'
      exit 1
    end
  end

  if create_alias && delete_alias
    puts 'Attempting to delete account alias...'
    if alias_deleted?(iam, account_alias)
      puts 'Account alias deleted. Account aliases now are:'
      list_aliases(iam)
    else
      puts 'Account alias not deleted. You will need to delete ' \
        'the alias yourself.'
    end
  end
end

run_me if $PROGRAM_NAME == __FILE__
```

AWS Key Management Service使用 AWS SDK for Ruby 範例

AWS Key Management Service(AWS KMS) 是針對雲端調整的加密和金鑰管理服務。您可以使用下面的實例來訪問AWS KMS使用 AWS SDK for Ruby。如需 AWS KMS 的詳細資訊，請參閱 [AWS KMS 說明文件](#)。如需有關AWS KMS用戶端的參考資訊，請參閱 [AWS::KMS::用戶端](#)。

主題

- [建立 AWS KMS key](#)
- [加密中的資料 AWS KMS](#)
- [解密中的資料 Blob AWS KMS](#)
- [在中重新加密數據 Blob AWS KMS](#)

建立 AWS KMS key

下面的例子使用了 AWS SDK for Ruby [create_key](#) 方法，它實現了創建一個[CreateKey](#)操作。AWS KMS keys由於此範例只會加密少量資料，因此 KMS 金鑰適用於我們的目的。對於較大量的資料，請使用 KMS 金鑰加密資料加密金鑰 (DEK)。

```
require "aws-sdk-kms" # v2: require 'aws-sdk'

# Create a AWS KMS key.
# As long we are only encrypting small amounts of data (4 KiB or less) directly,
# a KMS key is fine for our purposes.
# For larger amounts of data,
# use the KMS key to encrypt a data encryption key (DEK).

client = Aws::KMS::Client.new

resp = client.create_key({
  tags: [
    {
      tag_key: "CreatedBy",
      tag_value: "ExampleUser"
    }
  ]
})

puts resp.key_metadata.key_id
```

請參閱 (詳見) 的[完整實例](#) GitHub。

加密中的資料 AWS KMS

下列範例會使用實作[加密作業的 Ruby 加密方法 AWS SDK 來加密](#)字串「1234567890」。此範例會顯示產生的加密 Blob 的可讀版本。

```
require "aws-sdk-kms" # v2: require 'aws-sdk'

# ARN of the AWS KMS key.
#
# Replace the fictitious key ARN with a valid key ID

keyId = "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"

text = "1234567890"

client = Aws::KMS::Client.new(region: "us-west-2")

resp = client.encrypt({
  key_id: keyId,
  plaintext: text,
})

# Display a readable version of the resulting encrypted blob.
puts "Blob:"
puts resp.ciphertext_blob.unpack("H*")
```

請參閱 (詳見) 的[完整實例](#) GitHub。

解密中的資料 Blob AWS KMS

下列範例會使用 AWS SDK for Ruby [解密方法](#) (實作「[解密](#)」作業) 來解密提供的字串並發出結果。

```
require "aws-sdk-kms" # v2: require 'aws-sdk'

# Decrypted blob

blob =
  "01020200785d68faeec386af1057904926253051eb2919d3c16078badf65b808b26dd057c101747cadf3593596e09"
blob_packed = [blob].pack("H*")
```

```
client = Aws::KMS::Client.new(region: "us-west-2")

resp = client.decrypt({
  ciphertext_blob: blob_packed
})

puts "Raw text: "
puts resp.plaintext
```

請參閱 (詳見) 的[完整實例](#) GitHub。

在中重新加密數據 Blob AWS KMS

下列範例會使用實[ReEncrypt](#)作作業的 Ruby [re_encrypt](#) 方法的 AWS SDK 解密加密資料，然後立即在新的資料下重新加密資料。AWS KMS key 此操作完全在伺服器端的 AWS KMS 中執行，所以絕不會在 AWS KMS 外部公開您的純文字。此範例會顯示產生之重新加密 Blob 的可讀版本。

```
require "aws-sdk-kms" # v2: require 'aws-sdk'

# Human-readable version of the ciphertext of the data to reencrypt.

blob =
  "01020200785d68faeec386af1057904926253051eb2919d3c16078badf65b808b26dd057c101747cadf3593596e09"
sourceCiphertextBlob = [blob].pack("H*")

# Replace the fictitious key ARN with a valid key ID

destinationKeyId = "arn:aws:kms:us-west-2:111122223333:key/0987dcba-09fe-87dc-65ba-ab0987654321"

client = Aws::KMS::Client.new(region: "us-west-2")

resp = client.re_encrypt({
  ciphertext_blob: sourceCiphertextBlob,
  destination_key_id: destinationKeyId
})

# Display a readable version of the resulting re-encrypted blob.
puts "Blob:"
puts resp.ciphertext_blob.unpack("H*")
```

請參閱 (詳見) 的[完整實例](#) GitHub。

AWS Lambda使用 AWS SDK for Ruby 範例

AWS Lambda(Lambda) 是適用於後端 Web 開發人員的零管理運算平台，可在AWS雲端中為您執行程式碼，並為您提供精細的定價結構。您可以使用下列範例，使用 AWS SDK for Ruby 件存取 Lambda。如需 Lambda 的詳細資訊，請參閱[AWS Lambda文件](#)。

主題

- [顯示所有 Lambda 函數相關資訊](#)
- [建立 Lambda 函數](#)
- [運行一個 Lambda 函數](#)
- [設定 Lambda 函數以接收通知](#)

顯示所有 Lambda 函數相關資訊

下列範例會顯示us-west-2區域中所有 Lambda 函數的名稱、ARN 和角色。

```
# Copyright 2010-2019 Amazon.com, Inc. or its affiliates. All Rights Reserved.
#
# This file is licensed under the Apache License, Version 2.0 (the "License").
# You may not use this file except in compliance with the License. A copy of the
# License is located at
#
# http://aws.amazon.com/apache2.0/
#
# This file is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS
# OF ANY KIND, either express or implied. See the License for the specific
# language governing permissions and limitations under the License.

require 'aws-sdk-lambda' # v2: require 'aws-sdk'

client = Aws::Lambda::Client.new(region: 'us-west-2')

client.list_functions.functions.each do |function|
  puts 'Name: ' + function.function_name
  puts 'ARN: ' + function.function_arn
  puts 'Role: ' + function.role
  puts
end
```

建立 Lambda 函數

下列範例會使用這些值建立us-west-2區域my-notification-function中命名的 Lambda 函數：

- 角色 ARN：my-resource-arn。在大多數情況下，您只需要將AWSLambdaExecute受管理的策略附加到此角色的策略。
- 功能入口點：my-package.my-class
- 執行時間：java8
- 壓縮檔案：my-zip-file.zip
- 鏟斗：my-notification-bucket
- 索引鍵：my-zip-file

```
# Copyright 2010-2019 Amazon.com, Inc. or its affiliates. All Rights Reserved.
#
# This file is licensed under the Apache License, Version 2.0 (the "License").
# You may not use this file except in compliance with the License. A copy of the
# License is located at
#
# http://aws.amazon.com/apache2.0/
#
# This file is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS
# OF ANY KIND, either express or implied. See the License for the specific
# language governing permissions and limitations under the License.

require 'aws-sdk-lambda' # v2: require 'aws-sdk'

client = Aws::Lambda::Client.new(region: 'us-west-2')

args = {}
args[:role] = 'my-resource-arn'
args[:function_name] = 'my-notification-function'
args[:handler] = 'my-package.my-class'

# Also accepts nodejs, nodejs4.3, and python2.7
args[:runtime] = 'java8'

code = {}
code[:zip_file] = 'my-zip-file.zip'
code[:s3_bucket] = 'my-notification-bucket'
code[:s3_key] = 'my-zip-file'
```

```
args[:code] = code

client.create_function(args)
```

運行一個 Lambda 函數

下列範例會執行us-west-2區域MyGetitemsFunction中名為的 Lambda 函數。該函數返回從數據庫中的項目列表。輸入的 JSON 如下所示。

```
{
  "SortBy": "name|time",
  "SortOrder": "ascending|descending",
  "Number": 50
}
```

其中：

- `SortBy`是排序結果的條件。我們的示例使用`time`，這意味著返回的項目按照它們添加到數據庫的順序進行排序。
- `SortOrder`是排序的順序。我們的示例使用`descending`，這意味著最近的項目是列表中的最後一個。
- `Number`是要擷取的項目數目上限 (預設值為 50)。我們的示例使用10，這意味著獲取 10 個最新的項目。

輸出 JSON 如下所示，其中：

- `STATUS-CODE`是 HTTP 狀態碼，200表示呼叫成功。
- `RESULT`是呼叫的結果，`success`或`failure`。
- `ERROR`如果是錯誤消息 `resultfailure`，否則為空字符串
- `DATA`是返回結果的數組，如果`result`是`success`，否則為零。

```
{
  "statusCode": "STATUS-CODE",
  "body": {
    "result": "RESULT",
    "error": "ERROR",
```

```
    "data": "DATA"
  }
}
```

第一步是加載我們使用的模塊：

- `aws-sdk`加載我們用來調用 Lambda 函數的 Ruby 模塊的 AWS SDK。
- `json`加載我們用於馬歇爾和解開請求和響應有效載荷的 JSON 模塊。
- `os`加載操作系統模塊，我們使用，以確保我們可以在 Microsoft Windows 上運行我們的 Ruby 應用程序。如果您使用的是不同的作業系統，則可以移除這些行。
- 然後我們建立用來叫用 Lambda 函數的 Lambda 用戶端。
- 接下來，我們為請求參數創建散列並調用 `MyGetItemsFunction`。
- 最後，我們解析響應，如果成功，我們打印出的項目。

```
# Copyright 2010-2019 Amazon.com, Inc. or its affiliates. All Rights Reserved.
#
# This file is licensed under the Apache License, Version 2.0 (the "License").
# You may not use this file except in compliance with the License. A copy of the
# License is located at
#
# http://aws.amazon.com/apache2.0/
#
# This file is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS
# OF ANY KIND, either express or implied. See the License for the specific
# language governing permissions and limitations under the License.

require 'aws-sdk-lambda' # v2: require 'aws-sdk'
require 'json'

# To run on Windows:
require 'os'
if OS.windows?
  Aws.use_bundled_cert!
end

client = Aws::Lambda::Client.new(region: 'us-west-2')

# Get the 10 most recent items
req_payload = {:SortBy => 'time', :SortOrder => 'descending', :NumberToGet => 10}
payload = JSON.generate(req_payload)
```



```
resp = client.invoke({
    function_name: 'MyGetItemsFunction',
    invocation_type: 'RequestResponse',
    log_type: 'None',
    payload: payload
})

resp_payload = JSON.parse(resp.payload.string) # , symbolize_names: true)

# If the status code is 200, the call succeeded
if resp_payload["statusCode"] == 200
  # If the result is success, we got our items
  if resp_payload["body"]["result"] == "success"
    # Print out items
    resp_payload["body"]["data"].each do |item|
      puts item
    end
  end
end
end
```

請參閱 (詳見) 的 [完整實例](#) GitHub。

設定 Lambda 函數以接收通知

下列範例會將us-west-2區域my-notification-function中名為的 Lambda 函數設定為接受來自 ARN 之資源的通知。my-resource-arn

```
# Copyright 2010-2019 Amazon.com, Inc. or its affiliates. All Rights Reserved.
#
# This file is licensed under the Apache License, Version 2.0 (the "License").
# You may not use this file except in compliance with the License. A copy of the
# License is located at
#
# http://aws.amazon.com/apache2.0/
#
# This file is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS
# OF ANY KIND, either express or implied. See the License for the specific
# language governing permissions and limitations under the License.

require 'aws-sdk-lambda' # v2: require 'aws-sdk'

client = Aws::Lambda::Client.new(region: 'us-west-2')
```

```
args = {}
args[:function_name] = 'my-notification-function'
args[:statement_id] = 'lambda_s3_notification'
args[:action] = 'lambda:InvokeFunction'
args[:principal] = 's3.amazonaws.com'
args[:source_arn] = 'my-resource-arn'

client.add_permission(args)
```

Amazon Polly 示例使用 AWS SDK for Ruby

Amazon Polly 是一種雲端服務，可將文字轉換為逼真的語音。Ruby 的 AWS 開發套件範例可以將 Amazon Polly 整合到您的應用程式中。[在 Amazon Polly 文檔中了解有關 Amazon Polly 的更多信息](#)。這些範例假設您已經設定並設定 SDK (也就是說，您已匯入所有必要的套件並設定認證和區域)。如需詳細資訊，請參閱[為 Ruby 安裝 AWS SDK](#) 和 [設定 AWS SDK for Ruby](#)。

主題

- [取得聲音清單](#)
- [獲取詞典列表](#)
- [合成語音](#)

取得聲音清單

此範例使用 [describe_voices](#) 方法來取得該地區中的美國英文語音清單。us-west-2

選擇 Copy 在本機儲存程式碼。

創建文件描述語音 .rb。

添加所需的寶石。

Note

Ruby 的 AWS SDK 版本 2 沒有特定於服務的寶石。

```
# Copyright 2010-2019 Amazon.com, Inc. or its affiliates. All Rights Reserved.
#
```

```
# This file is licensed under the Apache License, Version 2.0 (the "License").
# You may not use this file except in compliance with the License. A copy of the
# License is located at
#
# http://aws.amazon.com/apache2.0/
#
# This file is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS
# OF ANY KIND, either express or implied. See the License for the specific
# language governing permissions and limitations under the License.

require 'aws-sdk-polly' # In v2: require 'aws-sdk'

begin
  # Create an Amazon Polly client using
  # credentials from the shared credentials file ~/.aws/credentials
  # and the configuration (region) from the shared configuration file ~/.aws/config
  polly = Aws::Polly::Client.new

  # Get US English voices
  resp = polly.describe_voices(language_code: 'en-US')

  resp.voices.each do |v|
    puts v.name
    puts ' ' + v.gender
    puts
  end
rescue StandardError => ex
  puts 'Could not get voices'
  puts 'Error message:'
  puts ex.message
end
```

請參閱 (詳見) 的[完整實例](#) GitHub。

獲取詞典列表

這個範例使用 [list_lexicons](#) 方法來取得區域中的辭典清單。us-west-2

選擇Copy在本機儲存程式碼。

創建文件中的列表詞典 .rb。

添加所需的寶石。

Note

Ruby 的 AWS SDK 版本 2 沒有特定於服務的寶石。

```
# Copyright 2010-2019 Amazon.com, Inc. or its affiliates. All Rights Reserved.
#
# This file is licensed under the Apache License, Version 2.0 (the "License").
# You may not use this file except in compliance with the License. A copy of the
# License is located at
#
# http://aws.amazon.com/apache2.0/
#
# This file is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS
# OF ANY KIND, either express or implied. See the License for the specific
# language governing permissions and limitations under the License.

require 'aws-sdk-polly' # In v2: require 'aws-sdk'

begin
  # Create an Amazon Polly client using
  # credentials from the shared credentials file ~/.aws/credentials
  # and the configuration (region) from the shared configuration file ~/.aws/config
  polly = Aws::Polly::Client.new

  resp = polly.list_lexicons

  resp.lexicons.each do |l|
    puts l.name
    puts '  Alphabet:' + l.attributes.alphabet
    puts '  Language:' + l.attributes.language
    puts
  end
rescue StandardError => ex
  puts 'Could not get lexicons'
  puts 'Error message:'
  puts ex.message
end
```

請參閱 (詳見) 的 [完整實例](#) GitHub。

合成語音

此範例使用 `synthesize_speech` 方法從檔案取得文字，並產生包含合成語音的 MP3 檔案。

選擇Copy在本機儲存程式碼。

創建文件波麗_合成_語音.rb。

添加所需的寶石。

Note

Ruby 的 AWS SDK 版本 2 沒有特定於服務的寶石。

```
# Copyright 2010-2019 Amazon.com, Inc. or its affiliates. All Rights Reserved.
#
# This file is licensed under the Apache License, Version 2.0 (the "License").
# You may not use this file except in compliance with the License. A copy of the
# License is located at
#
# http://aws.amazon.com/apache2.0/
#
# This file is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS
# OF ANY KIND, either express or implied. See the License for the specific
# language governing permissions and limitations under the License.

require 'aws-sdk-polly' # In v2: require 'aws-sdk'

begin
  # Get the filename from the command line
  if ARGV.empty?()
    puts 'You must supply a filename'
    exit 1
  end

  filename = ARGV[0]

  # Open file and get the contents as a string
  if File.exist?(filename)
    contents = IO.read(filename)
  else
```

```
puts 'No such file: ' + filename
exit 1
end

# Create an Amazon Polly client using
# credentials from the shared credentials file ~/.aws/credentials
# and the configuration (region) from the shared configuration file ~/.aws/config
polly = Aws::Polly::Client.new

resp = polly.synthesize_speech({
  output_format: "mp3",
  text: contents,
  voice_id: "Joanna",
})

# Save output
# Get just the file name
# abc/xyz.txt -> xyz.txt
name = File.basename(filename)

# Split up name so we get just the xyz part
parts = name.split('.')
first_part = parts[0]
mp3_file = first_part + '.mp3'

IO.copy_stream(resp.audio_stream, mp3_file)

puts 'Wrote MP3 content to: ' + mp3_file
rescue StandardError => ex
  puts 'Got error:'
  puts 'Error message:'
  puts ex.message
end
```

Note

產生的 MP3 檔案採用 MPEG-2 格式。

請參閱 (詳見) 的 [完整實例](#) GitHub。

亞馬遜 RDS 示例使用AWS開發套件的紅寶石

Amazon Relational Database Service (Amazon RDS) 是一種 Web 服務，可讓您更輕鬆地在雲端中設定、操作和擴展關聯式資料庫。您可以使用下列範例，使用 AWS SDK for Ruby 存取 Amazon RDS。如需 Amazon RDS 的詳細資訊，請參閱 [Amazon Relational Database Service 文件](#)。

Note

下列部分範例會使用類別 2.2.18 版本中所引入的 `Aws::RDS::Resource` 方法。要運行這些示例，您必須使用該版本或更高版本的 `aws-sdk-gem`。

主題

- [取得所有 Amazon RDS 執行個體的相關資訊](#)
- [取得有關所有亞馬遜 RDS 快照的資訊](#)
- [取得所有 Amazon RDS 叢集及其快照的相關資訊](#)
- [取得所有亞馬遜 RDS 安全群組的相關資訊](#)
- [取得所有 Amazon RDS 子網路群組的相關資訊](#)
- [取得有關所有亞馬遜 RDS 參數群組的資訊](#)
- [建立亞馬遜 RDS 執行個體的快照](#)
- [建立亞馬遜 RDS 叢集的快照](#)

取得所有 Amazon RDS 執行個體的相關資訊

下列範例列出 us-west-2 區域中所有 Amazon RDS 執行個體的名稱 (ID) 和狀態。

```
# Copyright 2010-2019 Amazon.com, Inc. or its affiliates. All Rights Reserved.
#
# This file is licensed under the Apache License, Version 2.0 (the "License").
# You may not use this file except in compliance with the License. A copy of the
# License is located at
#
# http://aws.amazon.com/apache2.0/
#
# This file is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS
# OF ANY KIND, either express or implied. See the License for the specific
# language governing permissions and limitations under the License.
```

```
require 'aws-sdk-rds' # v2: require 'aws-sdk'

rds = Aws::RDS::Resource.new(region: 'us-west-2')

rds.db_instances.each do |i|
  puts "Name (ID): #{i.id}"
  puts "Status : #{i.db_instance_status}"
  puts
end
```

取得有關所有亞馬遜 RDS 快照的資訊

下列範例列出us-west-2區域中所有 Amazon RDS (執行個體) 快照的名稱 (ID) 和狀態。

```
# Copyright 2010-2019 Amazon.com, Inc. or its affiliates. All Rights Reserved.
#
# This file is licensed under the Apache License, Version 2.0 (the "License").
# You may not use this file except in compliance with the License. A copy of the
# License is located at
#
# http://aws.amazon.com/apache2.0/
#
# This file is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS
# OF ANY KIND, either express or implied. See the License for the specific
# language governing permissions and limitations under the License.

require 'aws-sdk-rds' # v2: require 'aws-sdk'

rds = Aws::RDS::Resource.new(region: 'us-west-2')

rds.db_snapshots.each do |s|
  puts "Name (ID): #{s.snapshot_id}"
  puts "Status:    #{s.status}"
end
```

取得所有 Amazon RDS 叢集及其快照的相關資訊

下列範例列出所有 Amazon RDS 叢集的名稱 (ID) 和狀態，以及其快照在us-west-2區域中的名稱 (ID) 和狀態。

```
# Copyright 2010-2019 Amazon.com, Inc. or its affiliates. All Rights Reserved.
```



```
#
# This file is licensed under the Apache License, Version 2.0 (the "License").
# You may not use this file except in compliance with the License. A copy of the
# License is located at
#
# http://aws.amazon.com/apache2.0/
#
# This file is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS
# OF ANY KIND, either express or implied. See the License for the specific
# language governing permissions and limitations under the License.

require 'aws-sdk-rds' # v2: require 'aws-sdk'

rds = Aws::RDS::Resource.new(region: 'us-west-2')

rds.db_clusters.each do |c|
  puts "Name (ID): #{c.id}"
  puts "Status:    #{c.status}"

  c.snapshots.each do |s|
    puts "  Snapshot: #{s.snapshot_id}"
    puts "  Status:    #{s.status}"
  end
end
end
```

取得所有亞馬遜 RDS 安全群組的相關資訊

下列範例列出us-west-2區域中所有 Amazon RDS 安全群組的名稱。

Note

亞馬遜 RDS 安全群組僅適用於您使用亞馬遜 EC2-典型平台時。如果您使用的是亞馬遜 EC2-VPC，請使用 VPC 安全群組。兩者都顯示在範例中。

Warning

我們將於 2022 年 8 月 15 日淘汰 EC2-Classic。建議您從 EC2-Classic 遷移至 VPC。如需詳細資訊，請參閱 [Amazon EC2 Linux 執行個體使用者指南](#) 中的從 EC2-Classic 遷移至 VPC 或 [Amazon EC2 Windows 執行個體使用者指南](#)。也請參閱部落格文章 [EC2-Classic 網路正在淘汰 - 本文介紹如何準備](#)。

```
# Copyright 2010-2019 Amazon.com, Inc. or its affiliates. All Rights Reserved.
#
# This file is licensed under the Apache License, Version 2.0 (the "License").
# You may not use this file except in compliance with the License. A copy of the
# License is located at
#
# http://aws.amazon.com/apache2.0/
#
# This file is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS
# OF ANY KIND, either express or implied. See the License for the specific
# language governing permissions and limitations under the License.

require 'aws-sdk-rds' # v2: require 'aws-sdk'

rds = Aws::RDS::Resource.new(region: 'us-west-2')

rds.db_instances.each do |i|
  # Show any security group IDs and descriptions
  puts 'Security Groups:'

  i.db_security_groups.each do |sg|
    puts sg.db_security_group_name
    puts '  ' + sg.db_security_group_description
    puts
  end

  # Show any VPC security group IDs and their status
  puts 'VPC Security Groups:'

  i.vpc_security_groups.each do |vsg|
    puts vsg.vpc_security_group_id
    puts '  ' + vsg.status
    puts
  end
end
end
```

取得所有 Amazon RDS 子網路群組的相關資訊

下列範例列出us-west-2區域中所有 Amazon RDS 子網路群組的名稱和狀態。

```
# Copyright 2010-2019 Amazon.com, Inc. or its affiliates. All Rights Reserved.
#
# This file is licensed under the Apache License, Version 2.0 (the "License").
```

```
# You may not use this file except in compliance with the License. A copy of the
# License is located at
#
# http://aws.amazon.com/apache2.0/
#
# This file is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS
# OF ANY KIND, either express or implied. See the License for the specific
# language governing permissions and limitations under the License.

require 'aws-sdk-rds' # v2: require 'aws-sdk'

rds = Aws::RDS::Resource.new(region: 'us-west-2')

rds.db_subnet_groups.each do |s|
  puts s.name
  puts ' ' + s.subnet_group_status
end
```

取得有關所有亞馬遜 RDS 參數群組的資訊

下列範例列出us-west-2區域中所有 Amazon RDS 參數群組的名稱和說明。

```
# Copyright 2010-2019 Amazon.com, Inc. or its affiliates. All Rights Reserved.
#
# This file is licensed under the Apache License, Version 2.0 (the "License").
# You may not use this file except in compliance with the License. A copy of the
# License is located at
#
# http://aws.amazon.com/apache2.0/
#
# This file is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS
# OF ANY KIND, either express or implied. See the License for the specific
# language governing permissions and limitations under the License.

require 'aws-sdk-rds' # v2: require 'aws-sdk'

rds = Aws::RDS::Resource.new(region: 'us-west-2')

rds.db_parameter_groups.each do |p|
  puts p.db_parameter_group_name
  puts ' ' + p.description
end
```

建立亞馬遜 RDS 執行個體的快照

下列範例會為區域中以執行個體名稱表示的 Amazon RDS 執行個體建立快照。us-west-2

Note

如果您的執行個體是叢集的成員，則無法建立執行個體的快照。您必須改為建立叢集的快照（請參閱[建立 Amazon RDS 叢集的快照](#)）。

```
# Copyright 2010-2019 Amazon.com, Inc. or its affiliates. All Rights Reserved.
#
# This file is licensed under the Apache License, Version 2.0 (the "License").
# You may not use this file except in compliance with the License. A copy of the
# License is located at
#
# http://aws.amazon.com/apache2.0/
#
# This file is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS
# OF ANY KIND, either express or implied. See the License for the specific
# language governing permissions and limitations under the License.

require 'aws-sdk-rds' # v2: require 'aws-sdk'

rds = Aws::RDS::Resource.new(region: 'us-west-2')

instance = rds.db_instance(instance_name)

date = Time.new
date_time = date.year.to_s + '-' + date.month.to_s + '-' + date.day.to_s + '-' +
  date.hour.to_s + '-' + date.min.to_s

id = instance_name + '-' + date_time

instance.create_snapshot({db_snapshot_identifier: id})

puts "Created snapshot #{id}"
```

建立亞馬遜 RDS 叢集的快照

下列範例會為區域中以叢集名稱表示的 Amazon RDS 叢集建立快照。us-west-2

```
# Copyright 2010-2019 Amazon.com, Inc. or its affiliates. All Rights Reserved.
#
# This file is licensed under the Apache License, Version 2.0 (the "License").
# You may not use this file except in compliance with the License. A copy of the
# License is located at
#
# http://aws.amazon.com/apache2.0/
#
# This file is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS
# OF ANY KIND, either express or implied. See the License for the specific
# language governing permissions and limitations under the License.

require 'aws-sdk-rds' # v2: require 'aws-sdk'

rds = Aws::RDS::Resource.new(region: 'us-west-2')

cluster = rds.db_cluster(cluster_name)

date = Time.new
date_time = date.year.to_s + '-' + date.month.to_s + '-' + date.day.to_s + '-' +
  date.hour.to_s + '-' + date.min.to_s

id = cluster_name + '-' + date_time

cluster.create_snapshot({db_cluster_snapshot_identifier: id})

puts "Created cluster snapshot #{id}"
```

Amazon SES 示例使用AWS開發套件的紅寶石

Amazon 簡易電子郵件服務 (Amazon SES) 是一種電子郵件平台，提供簡單、經濟實惠的方式，讓您使用自己的電子郵件地址和網域傳送和接收電子郵件。您可以使用下列範例，使用 AWS SDK for Ruby 存取 Amazon SES。如需有關 Amazon SES 的詳細資訊，請參閱 [Amazon SES 文件](#)。

主題

- [列出有效 Amazon SES 電子郵件](#)
- [驗證 Amazon SES 中的電子郵件地址](#)
- [將訊息傳送到 Amazon SES 中的電子郵件地址](#)
- [獲取 Amazon SES 統計](#)

列出有效 Amazon SES 電子郵件

下列範例示範如何使用適用於 Ruby 的 AWS 開發套件列出有效的 Amazon SES 電子郵件地址。

```
# Copyright 2010-2019 Amazon.com, Inc. or its affiliates. All Rights Reserved.
#
# This file is licensed under the Apache License, Version 2.0 (the "License").
# You may not use this file except in compliance with the License. A copy of the
# License is located at
#
# http://aws.amazon.com/apache2.0/
#
# This file is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS
# OF ANY KIND, either express or implied. See the License for the specific
# language governing permissions and limitations under the License.

require 'aws-sdk-ses' # v2: require 'aws-sdk'

# Create client in us-west-2 region
client = Aws::SES::Client.new(region: 'us-west-2')

# Get up to 1000 identities
ids = client.list_identities({
  identity_type: "EmailAddress"
})

ids.identities.each do |email|
  attrs = client.get_identity_verification_attributes({
    identities: [email]
  })

  status = attrs.verification_attributes[email].verification_status

  # Display email addresses that have been verified
  if status == "Success"
    puts email
  end
end
```

請參閱 (詳見) 的 [完整實例](#) GitHub。

驗證 Amazon SES 中的電子郵件地址

下列範例示範如何使用適用於 Ruby 的 AWS 開發套件來驗證 Amazon SES 電子郵件地址。

```
# Copyright 2010-2019 Amazon.com, Inc. or its affiliates. All Rights Reserved.
#
# This file is licensed under the Apache License, Version 2.0 (the "License").
# You may not use this file except in compliance with the License. A copy of the
# License is located at
#
# http://aws.amazon.com/apache2.0/
#
# This file is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS
# OF ANY KIND, either express or implied. See the License for the specific
# language governing permissions and limitations under the License.

require 'aws-sdk-ses' # v2: require 'aws-sdk'

# Replace recipient@example.com with a "To" address.
recipient = "recipient@example.com"

# Create a new SES resource in the us-west-2 region.
# Replace us-west-2 with the AWS Region you're using for Amazon SES.
ses = Aws::SES::Client.new(region: 'us-west-2')

# Try to verify email address.
begin
  ses.verify_email_identity({
    email_address: recipient
  })

  puts 'Email sent to ' + recipient

# If something goes wrong, display an error message.
rescue Aws::SES::Errors::ServiceError => error
  puts "Email not sent. Error message: #{error}"
end
```

請參閱 (詳見) 的 [完整實例](#) GitHub。

將訊息傳送到 Amazon SES 中的電子郵件地址

下列範例示範如何使用適用於 Ruby 的開發 AWS 套件，將訊息傳送至 Amazon SES 電子郵件地址。

```
# Copyright 2010-2019 Amazon.com, Inc. or its affiliates. All Rights Reserved.
#
# This file is licensed under the Apache License, Version 2.0 (the "License").
# You may not use this file except in compliance with the License. A copy of the
# License is located at
#
# http://aws.amazon.com/apache2.0/
#
# This file is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS
# OF ANY KIND, either express or implied. See the License for the specific
# language governing permissions and limitations under the License.

require 'aws-sdk-ses' # v2: require 'aws-sdk'

# Replace sender@example.com with your "From" address.
# This address must be verified with Amazon SES.
sender = 'sender@example.com'

# Replace recipient@example.com with a "To" address. If your account
# is still in the sandbox, this address must be verified.
recipient = 'recipient@example.com'

# Specify a configuration set. To use a configuration
# set, uncomment the next line and line 74.
# configsetname = "ConfigSet"

# The subject line for the email.
subject = 'Amazon SES test (AWS SDK for Ruby)'

# The HTML body of the email.
htmlbody =
  '<h1>Amazon SES test (AWS SDK for Ruby)</h1>\'
  '<p>This email was sent with <a href="https://aws.amazon.com/ses/">\'
  'Amazon SES</a> using the <a href="https://aws.amazon.com/sdk-for-ruby/">\'
  'AWS SDK for Ruby</a>.'
```



```
# Replace us-west-2 with the AWS Region you're using for Amazon SES.
ses = Aws::SES::Client.new(region: 'us-west-2')

# Try to send the email.
begin
  # Provide the contents of the email.
  ses.send_email(
    destination: {
      to_addresses: [
        recipient
      ]
    },
    message: {
      body: {
        html: {
          charset: encoding,
          data: htmlbody
        },
        text: {
          charset: encoding,
          data: textbody
        }
      },
      subject: {
        charset: encoding,
        data: subject
      }
    },
    source: sender,
    # Uncomment the following line to use a configuration set.
    # configuration_set_name: configsetname,
  )

  puts 'Email sent to ' + recipient

# If something goes wrong, display an error message.
rescue Aws::SES::Errors::ServiceError => error
  puts "Email not sent. Error message: #{error}"
end
```

請參閱 (詳見) 的 [完整實例](#) GitHub。

獲取 Amazon SES 統計

下列範例示範如何使用適用於 Ruby 的 AWS 開發套件來取得有關 Amazon SES 的統計資料。使用這些信息來避免在電子郵件被退回或拒絕時損壞您的聲譽。

```
# Copyright 2010-2019 Amazon.com, Inc. or its affiliates. All Rights Reserved.
#
# This file is licensed under the Apache License, Version 2.0 (the "License").
# You may not use this file except in compliance with the License. A copy of the
# License is located at
#
# http://aws.amazon.com/apache2.0/
#
# This file is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS
# OF ANY KIND, either express or implied. See the License for the specific
# language governing permissions and limitations under the License.

require 'aws-sdk-ses' # v2: require 'aws-sdk'

# Create a new SES resource in the us-west-2 region.
# Replace us-west-2 with the AWS Region you're using for Amazon SES.
ses = Aws::SES::Client.new(region: 'us-west-2')

begin
  # Get send statistics so we don't ruin our reputation
  resp = ses.get_send_statistics({})

  dps = resp.send_data_points

  puts "Got #{dps.count} data point(s):"
  puts

  dps.each do |dp|
    puts "Timestamp:  #{dp.timestamp}" #=> Time
    puts "Attempts:   #{dp.delivery_attempts}" #=> Integer
    puts "Bounces:     #{dp.bounces}" #=> Integer
    puts "Complaints:  #{dp.complaints}" #=> Integer
    puts "Rejects:     #{dp.rejects}" #-> Integer
    puts
  end

  # If something goes wrong, display an error message.
rescue Aws::SES::Errors::ServiceError => error
```

```
puts "Error: #{error}"
end
```

請參閱 (詳見) 的 [完整實例](#) GitHub。

使用 AWS SDK for Ruby 件的 Amazon SNS 示例

Amazon Simple Notification Service (Amazon SNS) 是一種 Web 服務，可讓應用程式、最終使用者和裝置立即從雲端傳送和接收通知。您可以使用下列範例，使用 AWS SDK for Ruby 件存取 Amazon SNS。如需有關 Amazon SNS 的詳細資訊，請參閱 [Amazon SNS 文件](#)。

主題

- [取得所有 Amazon SNS 主題的相關資訊](#)
- [創建一個 Amazon SNS 主題](#)
- [在 Amazon SNS 主題中獲取有關所有訂閱的信息](#)
- [在 Amazon SNS 主題中創建訂閱](#)
- [傳送訊息給所有 Amazon SNS 主題訂閱者](#)
- [啟用資源以發佈到 Amazon SNS 主題](#)

取得所有 Amazon SNS 主題的相關資訊

下列範例會列出 us-west-2 區域中 Amazon SNS 主題的 ARN。

```
# Copyright 2010-2019 Amazon.com, Inc. or its affiliates. All Rights Reserved.
#
# This file is licensed under the Apache License, Version 2.0 (the "License").
# You may not use this file except in compliance with the License. A copy of the
# License is located at
#
# http://aws.amazon.com/apache2.0/
#
# This file is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS
# OF ANY KIND, either express or implied. See the License for the specific
# language governing permissions and limitations under the License.

require 'aws-sdk-sns' # v2: require 'aws-sdk'

sns = Aws::SNS::Resource.new(region: 'us-west-2')
```

```
sns.topics.each do |topic|
  puts topic.arn
end
```

創建一個 Amazon SNS 主題

下列範例會在us-west-2區域MyGroovyTopic中建立主題，並顯示產生的主題 ARN。

```
# Copyright 2010-2019 Amazon.com, Inc. or its affiliates. All Rights Reserved.
#
# This file is licensed under the Apache License, Version 2.0 (the "License").
# You may not use this file except in compliance with the License. A copy of the
# License is located at
#
# http://aws.amazon.com/apache2.0/
#
# This file is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS
# OF ANY KIND, either express or implied. See the License for the specific
# language governing permissions and limitations under the License.

require 'aws-sdk-sns' # v2: require 'aws-sdk'

sns = Aws::SNS::Resource.new(region: 'us-west-2')

topic = sns.create_topic(name: 'MyGroovyTopic')
puts topic.arn
```

在 Amazon SNS 主題中獲取有關所有訂閱的信息

下列範例列出us-west-2區域arn:aws:sns:us-west-2:123456789:MyGroovyTopic中具有 ARN 之主題之 Amazon SNS 訂閱的電子郵件地址。

```
# Copyright 2010-2019 Amazon.com, Inc. or its affiliates. All Rights Reserved.
#
# This file is licensed under the Apache License, Version 2.0 (the "License").
# You may not use this file except in compliance with the License. A copy of the
# License is located at
#
# http://aws.amazon.com/apache2.0/
#
# This file is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS
# OF ANY KIND, either express or implied. See the License for the specific
# language governing permissions and limitations under the License.
```

```
require 'aws-sdk-sns' # v2: require 'aws-sdk'

sns = Aws::SNS::Resource.new(region: 'us-west-2')

topic = sns.topic('arn:aws:sns:us-west-2:123456789:MyGroovyTopic')

topic.subscriptions.each do |s|
  puts s.attributes['Endpoint']
end
```

在 Amazon SNS 主題中創建訂閱

下列範例會針對MyGroovyUser@MyGroovy.com在us-west-2區域中擁有電子郵件地址arn:aws:sns:us-west-2:123456789:MyGroovyTopic的使用者建立含 ARN 的主題訂閱，並顯示產生的 ARN。最初 ARN 值正在等待確認。當使用者確認其電子郵件地址時，此值會變成真正的 ARN。

```
# Copyright 2010-2019 Amazon.com, Inc. or its affiliates. All Rights Reserved.
#
# This file is licensed under the Apache License, Version 2.0 (the "License").
# You may not use this file except in compliance with the License. A copy of the
# License is located at
#
# http://aws.amazon.com/apache2.0/
#
# This file is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS
# OF ANY KIND, either express or implied. See the License for the specific
# language governing permissions and limitations under the License.

require 'aws-sdk-sns' # v2: require 'aws-sdk'

sns = Aws::SNS::Resource.new(region: 'us-west-2')

topic = sns.topic('arn:aws:sns:us-west-2:123456789:MyGroovyTopic')

sub = topic.subscribe({
  protocol: 'email',
  endpoint: 'MyGroovyUser@MyGroovy.com'
})

puts sub.arn
```

傳送訊息給所有 Amazon SNS 主題訂閱者

下列範例會傳送訊息「您好！」給所有使用 ARN `arn:aws:sns:us-west-2:123456789:MyGroovyTopic` 的 Amazon SNS 主題的訂閱者。

```
# Copyright 2010-2019 Amazon.com, Inc. or its affiliates. All Rights Reserved.
#
# This file is licensed under the Apache License, Version 2.0 (the "License").
# You may not use this file except in compliance with the License. A copy of the
# License is located at
#
# http://aws.amazon.com/apache2.0/
#
# This file is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS
# OF ANY KIND, either express or implied. See the License for the specific
# language governing permissions and limitations under the License.

require 'aws-sdk-sns' # v2: require 'aws-sdk'

sns = Aws::SNS::Resource.new(region: 'us-west-2')

topic = sns.topic('arn:aws:sns:us-west-2:123456789:MyGroovyTopic')

topic.publish({
  message: 'Hello!'
})
```

啟用資源以發佈到 Amazon SNS 主題

下列範例可讓具有 ARN 的資源發佈 `my-resource-arn` 至區域中含 ARN `my-topic-arn` 的 `us-west-2` 主題。

```
# Copyright 2010-2019 Amazon.com, Inc. or its affiliates. All Rights Reserved.
#
# This file is licensed under the Apache License, Version 2.0 (the "License").
# You may not use this file except in compliance with the License. A copy of the
# License is located at
#
# http://aws.amazon.com/apache2.0/
#
# This file is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS
# OF ANY KIND, either express or implied. See the License for the specific
# language governing permissions and limitations under the License.
```

```
require 'aws-sdk-sns' # v2: require 'aws-sdk'

policy = '{
  "Version":"2008-10-17",
  "Id":"__default_policy_ID",
  "Statement":[{
    "Sid":"__default_statement_ID",
    "Effect":"Allow",
    "Principal":{"
      "AWS":"*"
    },
    "Action":["SNS:Publish"],
    "Resource":"' + my-topic-arn + '",
    "Condition":{"
      "ArnEquals":{"
        "AWS:SourceArn":"' + my-resource-arn + '"}
      }
    }
  ]
}'

sns = Aws::SNS::Resource.new(region: 'us-west-2')

# Get topic by ARN
topic = sns.topic(my-topic-arn)

# Add policy to topic
topic.set_attributes({
  attribute_name: "Policy",
  attribute_value: policy
})
```

使用 AWS SDK for Ruby 件的 Amazon SQS 示例

Amazon Simple Queue Service (Amazon SQS) 是全受管訊息佇列服務，可輕鬆分離和擴展微型服務、分散式系統和無伺服器應用程式。您可以使用下列範例，使用 AWS SDK for Ruby 存取 Amazon SQS。如需有關 Amazon SQS 的詳細資訊，請參閱 [Amazon SQS](#) 文件。

主題

- [取得 Amazon SQS 中所有佇列的相關資訊](#)
- [在 Amazon SQS 中創建隊列](#)
- [使用 Amazon SQS 中的隊列](#)

- [在 Amazon SQS 中發送消息](#)
- [在 Amazon SQS 中傳送和接收訊息](#)
- [在 Amazon SQS 中接收訊息](#)
- [在 Amazon SQS 中使用長輪詢接收訊息](#)
- [在 Amazon SQS 中啟用長輪詢](#)
- [使用 Amazon SQS 中的 QueuePoller 類別接收訊息](#)
- [在 Amazon SQS 中重定向死信](#)
- [刪除 Amazon SQS 中的隊列](#)
- [啟用資源以發佈到 Amazon SQS 中的佇列](#)
- [在 Amazon SQS 中使用無效字母隊列](#)
- [在 Amazon SQS 中指定訊息可見性逾時](#)

取得 Amazon SQS 中所有佇列的相關資訊

下列範例列出區域中 Amazon SQS 佇列中傳送中的 URL、ARN、可用訊息和傳送中的 us-west-2 訊息。

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX-License-Identifier: Apache-2.0

require 'aws-sdk-sqs'
require 'aws-sdk-sts'

# Lists the URLs of available queues in Amazon Simple Queue Service (Amazon SQS).
#
# @param sqs_client [Aws::SQS::Client] An initialized Amazon SQS client.
# @example
#   list_queue_urls(Aws::SQS::Client.new(region: 'us-east-1'))
def list_queue_urls(sqs_client)
  queues = sqs_client.list_queues

  queues.queue_urls.each do |url|
    puts url
  end
rescue StandardError => e
  puts "Error listing queue URLs: #{e.message}"
end
```



```
# Lists the attributes of a queue in Amazon Simple Queue Service (Amazon SQS).
#
# @param sqs_client [Aws::SQS::Client] An initialized Amazon SQS client.
# @param queue_url [String] The URL of the queue.
# @example
#   list_queue_attributes(
#     Aws::SQS::Client.new(region: 'us-east-1'),
#     'https://sqs.us-east-1.amazonaws.com/111111111111/my-queue'
#   )
def list_queue_attributes(sqs_client, queue_url)
  attributes = sqs_client.get_queue_attributes(
    queue_url: queue_url,
    attribute_names: [ "All" ]
  )

  attributes.attributes.each do |key, value|
    puts "#{key}: #{value}"
  end

rescue StandardError => e
  puts "Error getting queue attributes: #{e.message}"
end

# Full example call:
def run_me
  region = 'us-east-1'
  queue_name = 'my-queue'

  sqs_client = Aws::SQS::Client.new(region: region)

  puts 'Listing available queue URLs...'
  list_queue_urls(sqs_client)

  sts_client = Aws::STS::Client.new(region: region)

  # For example:
  # 'https://sqs.us-east-1.amazonaws.com/111111111111/my-queue'
  queue_url = 'https://sqs.' + region + '.amazonaws.com/' +
    sts_client.get_caller_identity.account + '/' + queue_name

  puts "\nGetting information about queue '#{queue_name}'..."
  list_queue_attributes(sqs_client, queue_url)
end
```

```
run_me if $PROGRAM_NAME == __FILE__
```

在 Amazon SQS 中創建隊列

下列範例會建立us-west-2區域MyGroovyQueue中命名的 Amazon SQS 佇列，並顯示其 URL。

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX-License-Identifier: Apache-2.0

require 'aws-sdk-sqs'

# Creates a queue in Amazon Simple Queue Service (Amazon SQS).
#
# @param sqs_client [Aws::SQS::Client] An initialized Amazon SQS client.
# @param queue_name [String] The name of the queue.
# @return [Boolean] true if the queue was created; otherwise, false.
# @example
#   exit 1 unless queue_created?(
#     Aws::SQS::Client.new(region: 'us-east-1'),
#     'my-queue'
#   )
def queue_created?(sqs_client, queue_name)
  sqs_client.create_queue(queue_name: queue_name)
  true
rescue StandardError => e
  puts "Error creating queue: #{e.message}"
  false
end

# Full example call:
def run_me
  region = 'us-east-1'
  queue_name = 'my-queue'
  sqs_client = Aws::SQS::Client.new(region: region)

  puts "Creating the queue named '#{queue_name}'..."

  if queue_created?(sqs_client, queue_name)
    puts 'Queue created.'
  else
    puts 'Queue not created.'
  end
end
```

```
run_me if $PROGRAM_NAME == __FILE__
```

使用 Amazon SQS 中的隊列

Amazon SQS 提供可高度擴展的託管佇列，可在訊息在應用程式或微型服務之間傳送時存放訊息。若要進一步了解佇列，請參閱 [Amazon SQS 佇列如何運作](#)。

在此範例中，您可以使用適用於 Ruby 的 AWS 開發套件搭配 Amazon SQS 來執行下列作業：

1. 使用取得佇列清單 [Aws::SQS::Client#list_queues](#)。
2. 使用建立佇列 [Aws::SQS::Client#create_queue](#)。
3. 使用取得佇列的 URL [Aws::SQS::Client#get_queue_url](#)。
4. 使用刪除佇列 [Aws::SQS::Client#delete_queue](#)。

先決條件

在執行範例程式碼之前，您必須先安裝並設定 AWS SDK for Ruby，如下所述：

- [安裝 AWS SDK for Ruby](#)
- [為紅寶石配置 AWS SDK](#)

範例

```
# Copyright 2010-2019 Amazon.com, Inc. or its affiliates. All Rights Reserved.
#
# This file is licensed under the Apache License, Version 2.0 (the "License").
# You may not use this file except in compliance with the License. A copy of the
# License is located at
#
# http://aws.amazon.com/apache2.0/
#
# This file is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS
# OF ANY KIND, either express or implied. See the License for the specific
# language governing permissions and limitations under the License.

# Demonstrates how to:
# 1. Get a list of your queues.
# 2. Create a queue.
# 3. Get the queue's URL.
```

```
# 4. Delete the queue.

require 'aws-sdk-sqs' # v2: require 'aws-sdk'

sqs = Aws::SQS::Client.new(region: 'us-east-1')

# Get a list of your queues.
sqs.list_queues.queue_urls.each do |queue_url|
  puts queue_url
end

# Create a queue.
queue_name = "my-queue"

begin
  sqs.create_queue({
    queue_name: queue_name,
    attributes: {
      "DelaySeconds" => "60", # Delay message delivery for 1 minute (60 seconds).
      "MessageRetentionPeriod" => "86400" # Delete message after 1 day (24 hours * 60
minutes * 60 seconds).
    }
  })
rescue Aws::SQS::Errors::QueueDeletedRecently
  puts "A queue with the name '#{queue_name}' was recently deleted. Wait at least 60
seconds and try again."
  exit(false)
end

# Get the queue's URL.
queue_url = sqs.get_queue_url(queue_name: queue_name).queue_url
puts queue_url

# Delete the queue.
sqs.delete_queue(queue_url: queue_url)
```

在 Amazon SQS 中發送消息

下列範例會透過 Amazon SQS 佇列傳送訊息「您好世界」，其中包含us-west-2區域URL中的URL。

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX-License-Identifier: Apache-2.0
```

```
require 'aws-sdk-sqs'
require 'aws-sdk-sts'

# Sends a message to a queue in Amazon Simple Queue Service (Amazon SQS).
#
# @param sqs_client [Aws::SQS::Client] An initialized Amazon SQS client.
# @param queue_url [String] The URL of the queue.
# @param message_body [String] The contents of the message to be sent.
# @return [Boolean] true if the message was sent; otherwise, false.
# @example
#   exit 1 unless message_sent?(
#     Aws::SQS::Client.new(region: 'us-east-1'),
#     'https://sqs.us-east-1.amazonaws.com/111111111111/my-queue',
#     'This is my message.'
#   )
def message_sent?(sqs_client, queue_url, message_body)
  sqs_client.send_message(
    queue_url: queue_url,
    message_body: message_body
  )
  true
rescue StandardError => e
  puts "Error sending message: #{e.message}"
  false
end

# Full example call:
def run_me
  region = 'us-east-1'
  queue_name = 'my-queue'
  message_body = 'This is my message.'

  sts_client = Aws::STS::Client.new(region: region)

  # For example:
  # 'https://sqs.us-east-1.amazonaws.com/111111111111/my-queue'
  queue_url = 'https://sqs.' + region + '.amazonaws.com/' +
    sts_client.get_caller_identity.account + '/' + queue_name

  sqs_client = Aws::SQS::Client.new(region: region)

  puts "Sending a message to the queue named '#{queue_name}'..."
end
```

```
if message_sent?(sqs_client, queue_url, message_body)
  puts 'Message sent.'
else
  puts 'Message not sent.'
end
end

run_me if $PROGRAM_NAME == __FILE__
```

下面的例子發送消息「你好世界」和「天氣如何？」透過 Amazon SQS 佇列，其中包含該us-west-2區域URL中的網址。

Note

如果佇列是 FIFO 佇列，除了id和message_group_id參數之外，您還必須包含message_body參數。

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX-License-Identifier: Apache-2.0

require 'aws-sdk-sqs'
require 'aws-sdk-sts'

# Sends multiple messages as a batch to a queue in
# Amazon Simple Queue Service (Amazon SQS).
#
# @param sqs_client [Aws::SQS::Client] An initialized Amazon SQS client.
# @param queue_url [String] The URL of the queue.
# @param entries [Hash] The contents of the messages to be sent,
#   in the correct format.
# @return [Boolean] true if the messages were sent; otherwise, false.
# @example
#   exit 1 unless messages_sent?(
#     Aws::SQS::Client.new(region: 'us-east-1'),
#     'https://sqs.us-east-1.amazonaws.com/111111111111/my-queue',
#     [
#       {
#         id: 'Message1',
#         message_body: 'This is the first message.'
#       },
#     ]
#   )
```

```
#         id: 'Message2',
#         message_body: 'This is the second message.'
#     }
# ]
# )
def messages_sent?(sqs_client, queue_url, entries)
  sqs_client.send_message_batch(
    queue_url: queue_url,
    entries: entries
  )
  true
rescue StandardError => e
  puts "Error sending messages: #{e.message}"
  false
end

# Full example call:
def run_me
  region = 'us-east-1'
  queue_name = 'my-queue'
  entries = [
    {
      id: 'Message1',
      message_body: 'This is the first message.'
    },
    {
      id: 'Message2',
      message_body: 'This is the second message.'
    }
  ]

  sts_client = Aws::STS::Client.new(region: region)

  # For example:
  # 'https://sqs.us-east-1.amazonaws.com/111111111111/my-queue'
  queue_url = 'https://sqs.' + region + '.amazonaws.com/' +
    sts_client.get_caller_identity.account + '/' + queue_name

  sqs_client = Aws::SQS::Client.new(region: region)

  puts "Sending messages to the queue named '#{queue_name}'..."

  if messages_sent?(sqs_client, queue_url, entries)
    puts 'Messages sent.'
```

```
else
  puts 'Messages not sent.'
end
end

run_me if $PROGRAM_NAME == __FILE__
```

在 Amazon SQS 中傳送和接收訊息

在 Amazon SQS 中建立佇列之後，您可以傳送訊息給該佇列，然後使用它。若要進一步了解，請參閱 [教學：將訊息傳送至 Amazon SQS 佇列](#) 和 [教學：從 Amazon SQS 佇列接收和刪除訊息](#)。

在此範例中，您可以使用適用於 Ruby 的 AWS 開發套件搭配 Amazon SQS 來執行下列作業：

1. 使用 [AWS::SQS::](#) 客戶端 #send_message 將消息發送到隊列。

Note

如果佇列是 FIFO 佇列，除了 id 和 message_group_id 參數之外，您還必須包含 message_body 參數。

1. 使用 [AWS::SQS::](#) 客戶端 #receive_message 接收佇列中的訊息。
2. 顯示有關訊息的資訊。
3. 使用 [AWS::SQS::](#) 客戶端 #delete_message 刪除佇列中的訊息。

先決條件

在執行範例程式碼之前，您必須先安裝並設定 AWS SDK for Ruby，如下所述：

- [安裝 AWS SDK for Ruby](#)
- [為紅寶石配置 AWS SDK](#)

您還需要創建隊列我的隊列，您可以在 Amazon SQS 控制台中執行此操作。

範例

```
# Copyright 2010-2019 Amazon.com, Inc. or its affiliates. All Rights Reserved.
#
```



```
# This file is licensed under the Apache License, Version 2.0 (the "License").
# You may not use this file except in compliance with the License. A copy of the
# License is located at
#
# http://aws.amazon.com/apache2.0/
#
# This file is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS
# OF ANY KIND, either express or implied. See the License for the specific
# language governing permissions and limitations under the License.

# Demonstrates how to:
# 1. Send a message to a queue.
# 2. Receive the message in the queue.
# 3. Display information about the message.
# 4. Delete the message from the queue.

require 'aws-sdk-sqs' # v2: require 'aws-sdk'

sqs = Aws::SQS::Client.new(region: 'us-east-1')

# Send a message to a queue.
queue_name = "my-queue"

begin
  queue_url = sqs.get_queue_url(queue_name: queue_name).queue_url

  # Create a message with three custom attributes: Title, Author, and WeeksOn.
  send_message_result = sqs.send_message({
    queue_url: queue_url,
    message_body: "Information about current NY Times fiction bestseller for week of
2016-12-11.",
    message_attributes: {
      "Title" => {
        string_value: "The Whistler",
        data_type: "String"
      },
      "Author" => {
        string_value: "John Grisham",
        data_type: "String"
      },
      "WeeksOn" => {
        string_value: "6",
        data_type: "Number"
      }
    }
  })
end
```

```
    }
  })
rescue Aws::SQS::Errors::NonExistentQueue
  puts "A queue named '#{queue_name}' does not exist."
  exit(false)
end

puts send_message_result.message_id

# Receive the message in the queue.
receive_message_result = sqs.receive_message({
  queue_url: queue_url,
  message_attribute_names: ["All"], # Receive all custom attributes.
  max_number_of_messages: 1, # Receive at most one message.
  wait_time_seconds: 0 # Do not wait to check for the message.
})

# Display information about the message.
# Display the message's body and each custom attribute value.
receive_message_result.messages.each do |message|
  puts message.body
  puts "Title: #{message.message_attributes["Title"]["string_value"]}"
  puts "Author: #{message.message_attributes["Author"]["string_value"]}"
  puts "WeeksOn: #{message.message_attributes["WeeksOn"]["string_value"]}"

  # Delete the message from the queue.
  sqs.delete_message({
    queue_url: queue_url,
    receipt_handle: message.receipt_handle
  })
end
```

在 Amazon SQS 中接收訊息

下列範例顯示 Amazon SQS 佇列中最多 10 則訊息的內文，其中包含該us-west-2區域URL中的URL。

Note

`receive_message` 不保證會取得所有郵件 (請參閱[分散式佇列的內容](#))，且預設不會刪除郵件。

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX-License-Identifier: Apache-2.0

require 'aws-sdk-sqs'
require 'aws-sdk-sts'

# Receives messages in a queue in Amazon Simple Queue Service (Amazon SQS).
#
# @param sqs_client [Aws::SQS::Client] An initialized Amazon SQS client.
# @param queue_url [String] The URL of the queue.
# @param max_number_of_messages [Integer] The maximum number of messages
#   to receive. This number must be 10 or less. The default is 10.
# @example
#   receive_messages(
#     Aws::SQS::Client.new(region: 'us-east-1'),
#     'https://sqs.us-east-1.amazonaws.com/111111111111/my-queue',
#     10
#   )
def receive_messages(sqs_client, queue_url, max_number_of_messages = 10)

  if max_number_of_messages > 10
    puts 'Maximum number of messages to receive must be 10 or less. ' \
      'Stopping program.'
    return
  end

  response = sqs_client.receive_message(
    queue_url: queue_url,
    max_number_of_messages: max_number_of_messages
  )

  if response.messages.count.zero?
    puts 'No messages to receive, or all messages have already ' \
      'been previously received.'
    return
  end

  response.messages.each do |message|
    puts '-' * 20
    puts "Message body: #{message.body}"
    puts "Message ID:  #{message.message_id}"
  end
end
```

```
rescue StandardError => e
  puts "Error receiving messages: #{e.message}"
end

# Full example call:
def run_me
  region = 'us-east-1'
  queue_name = 'my-queue'
  max_number_of_messages = 10

  sts_client = Aws::STS::Client.new(region: region)

  # For example:
  # 'https://sqs.us-east-1.amazonaws.com/111111111111/my-queue'
  queue_url = 'https://sqs.' + region + '.amazonaws.com/' +
    sts_client.get_caller_identity.account + '/' + queue_name

  sqs_client = Aws::SQS::Client.new(region: region)

  puts "Receiving messages from queue '#{queue_name}'..."

  receive_messages(sqs_client, queue_url, max_number_of_messages)
end

run_me if $PROGRAM_NAME == __FILE__
```

在 Amazon SQS 中使用長輪詢接收訊息

下列範例會等待最多 10 秒鐘，在 Amazon SQS 佇列中顯示最多 10 則訊息的內文，其中包含區域 URL 中的 us-west-2 URL。

如果您未指定等待時間，則預設值為 0 (Amazon SQS 不會等待)。

```
# Copyright 2010-2019 Amazon.com, Inc. or its affiliates. All Rights Reserved.
#
# This file is licensed under the Apache License, Version 2.0 (the "License").
# You may not use this file except in compliance with the License. A copy of the
# License is located at
#
# http://aws.amazon.com/apache2.0/
#
# This file is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS
# OF ANY KIND, either express or implied. See the License for the specific
```

```
# language governing permissions and limitations under the License.

require 'aws-sdk-sqs' # v2: require 'aws-sdk'

sqs = Aws::SQS::Client.new(region: 'us-west-2')

resp = sqs.receive_message(queue_url: URL, max_number_of_messages: 10,
  wait_time_seconds: 10)

resp.messages.each do |m|
  puts m.body
end
```

在 Amazon SQS 中啟用長輪詢

長輪詢有助於降低使用 Amazon SQS 的成本，方法是減少空白回應的數量並消除錯誤的空白回應。如需長輪詢的詳細資訊，請參閱 [Amazon SQS 長輪詢](#)。

在此範例中，您可以使用適用於 Ruby 的 AWS 開發套件搭配 Amazon SQS 來執行下列作業：

1. 使用 `Aws::SQS::客戶端 #create_queue` 創建一個隊列並將其設置為長輪詢。
2. 使用 `Aws::SQS::用戶端 #set_queue_attributes`，為現有佇列設定長輪詢。
3. 使用 `AWS::SQS::用戶端 #receive_message`，在接收佇列訊息時設定長輪詢。

先決條件

在執行範例程式碼之前，您必須先安裝並設定 AWS SDK for Ruby，如下所述：

- [安裝 AWS SDK for Ruby](#)
- [為紅寶石配置 AWS SDK](#)

您還需要建立現有佇列和接收佇列的佇列，您可以在 Amazon SQS 主控台中執行這項操作。

範例

```
# Copyright 2010-2019 Amazon.com, Inc. or its affiliates. All Rights Reserved.
#
# This file is licensed under the Apache License, Version 2.0 (the "License").
# You may not use this file except in compliance with the License. A copy of the
# License is located at
#
```

```
# http://aws.amazon.com/apache2.0/
#
# This file is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS
# OF ANY KIND, either express or implied. See the License for the specific
# language governing permissions and limitations under the License.

# Demonstrates how to:
# 1. Create a queue and set it for long polling.
# 2. Set long polling for an existing queue.
# 3. Set long polling when receiving messages for a queue.

require 'aws-sdk-sqs' # v2: require 'aws-sdk'

sqs = Aws::SQS::Client.new(region: 'us-east-1')

# Create a queue and set it for long polling.
new_queue_name = "new-queue"

create_queue_result = sqs.create_queue({
  queue_name: new_queue_name,
  attributes: {
    "ReceiveMessageWaitTimeSeconds" => "20" # Wait 20 seconds to receive messages.
  },
})

puts create_queue_result.queue_url

# Set long polling for an existing queue.
begin
  existing_queue_name = "existing-queue"
  existing_queue_url = sqs.get_queue_url(queue_name: existing_queue_name).queue_url

  sqs.set_queue_attributes({
    queue_url: existing_queue_url,
    attributes: {
      "ReceiveMessageWaitTimeSeconds" => "20" # Wait 20 seconds to receive messages.
    },
  })
rescue Aws::SQS::Errors::NonExistentQueue
  puts "Cannot set long polling for a queue named '#{existing_queue_name}', as it does
  not exist."
end

# Set long polling when receiving messages for a queue.
```

```
# 1. Using receive_message.
begin
  receive_queue_name = "receive-queue"
  receive_queue_url = sqs.get_queue_url(queue_name: receive_queue_name).queue_url

  puts "Begin receipt of any messages using receive_message..."
  receive_message_result = sqs.receive_message({
    queue_url: receive_queue_url,
    attribute_names: ["All"], # Receive all available built-in message attributes.
    message_attribute_names: ["All"], # Receive any custom message attributes.
    max_number_of_messages: 10 # Receive up to 10 messages, if there are that many.
  })

  puts "Received #{receive_message_result.messages.count} message(s)."
rescue Aws::SQS::Errors::NonExistentQueue
  puts "Cannot receive messages using receive_message for a queue named
'#{receive_queue_name}', as it does not exist."
end

# 2. Using Aws::SQS::QueuePoller.
begin
  puts "Begin receipt of any messages using Aws::SQS::QueuePoller..."
  puts "(Will keep polling until no more messages available for at least 60 seconds.)"
  poller = Aws::SQS::QueuePoller.new(receive_queue_url)

  poller_stats = poller.poll({
    max_number_of_messages: 10,
    idle_timeout: 60 # Stop polling after 60 seconds of no more messages available
  }) do |messages|
    messages.each do |message|
      puts "Message body: #{message.body}"
    end
  end
end

# Note: If poller.poll is successful, all received messages are automatically deleted
from the queue.

puts "Poller stats:"
puts "  Polling started at: #{poller_stats.polling_started_at}"
puts "  Polling stopped at: #{poller_stats.polling_stopped_at}"
puts "  Last message received at: #{poller_stats.last_message_received_at}"
puts "  Number of polling requests: #{poller_stats.request_count}"
puts "  Number of received messages: #{poller_stats.received_message_count}"
```

```
rescue Aws::SQS::Errors::NonExistentQueue
  puts "Cannot receive messages using Aws::SQS::QueuePoller for a queue named
    '#{receive_queue_name}', as it does not exist."
end
```

使用 Amazon SQS 中的 QueuePoller 類別接收訊息

下列範例使用公用QueuePoller程式類別來顯示 Amazon SQS 佇列中所有訊息的內文，其中包含該us-west-2區域URL中的 URL，並刪除訊息。閒置約 15 秒後，指令碼逾時。

```
# Copyright 2010-2019 Amazon.com, Inc. or its affiliates. All Rights Reserved.
#
# This file is licensed under the Apache License, Version 2.0 (the "License").
# You may not use this file except in compliance with the License. A copy of the
# License is located at
#
# http://aws.amazon.com/apache2.0/
#
# This file is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS
# OF ANY KIND, either express or implied. See the License for the specific
# language governing permissions and limitations under the License.

require 'aws-sdk-sqs' # v2: require 'aws-sdk'

Aws.config.update({region: 'us-west-2'})

poller = Aws::SQS::QueuePoller.new(URL)

poller.poll(idle_timeout: 15) do |msg|
  puts msg.body
end
```

下列範例會使用 URL 在 Amazon SQS 佇列中迴圈URL，並等待最多持續時間秒數。

您可以透過在 Amazon SQS 中取得[所有佇列的相關資訊中執行 Amazon SQS 範例來取得正確的 URL](#)。

```
# Copyright 2010-2019 Amazon.com, Inc. or its affiliates. All Rights Reserved.
#
# This file is licensed under the Apache License, Version 2.0 (the "License").
# You may not use this file except in compliance with the License. A copy of the
# License is located at
```



```
#
# http://aws.amazon.com/apache2.0/
#
# This file is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS
# OF ANY KIND, either express or implied. See the License for the specific
# language governing permissions and limitations under the License.

require 'aws-sdk-sqs' # v2: require 'aws-sdk'

Aws.config.update({region: 'us-west-2'})

poller = Aws::SQS::QueuePoller.new(URL)

poller.poll(wait_time_seconds: duration, idle_timeout: duration + 1) do |msg|
  puts msg.body
end
```

下列範例會使用 URL 在 Amazon SQS 佇列中迴圈 URL，並提供處理訊息的可見性逾時秒數 (以方法 `do_something` 表示)。

```
# Copyright 2010-2019 Amazon.com, Inc. or its affiliates. All Rights Reserved.
#
# This file is licensed under the Apache License, Version 2.0 (the "License").
# You may not use this file except in compliance with the License. A copy of the
# License is located at
#
# http://aws.amazon.com/apache2.0/
#
# This file is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS
# OF ANY KIND, either express or implied. See the License for the specific
# language governing permissions and limitations under the License.

require 'aws-sdk-sqs' # v2: require 'aws-sdk'

# Process the message
def do_something(msg)
  puts msg.body
end

Aws.config.update({region: 'us-west-2'})

poller = Aws::SQS::QueuePoller.new(URL)
```

```
poller.poll(visibility_timeout: timeout, idle_timeout: timeout + 1) do |msg|
  do_something(msg)
end
```

下列範例會使用 URL 循環瀏覽 Amazon SQS 佇列URL，並針對需要該方法do_something2進行其他處理的任何訊息變更可見性逾時秒數。

```
# Copyright 2010-2019 Amazon.com, Inc. or its affiliates. All Rights Reserved.
#
# This file is licensed under the Apache License, Version 2.0 (the "License").
# You may not use this file except in compliance with the License. A copy of the
# License is located at
#
# http://aws.amazon.com/apache2.0/
#
# This file is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS
# OF ANY KIND, either express or implied. See the License for the specific
# language governing permissions and limitations under the License.

require 'aws-sdk-sqs' # v2: require 'aws-sdk'

# Process the message
def do_something(_)
  true
end

# Do additional processing
def do_something2(msg)
  puts msg.body
end

Aws.config.update({region: 'us-west-2'})

poller = Aws::SQS::QueuePoller.new(URL)

poller.poll(idle_timeout: timeout + 1) do |msg|
  if do_something(msg)
    # need more time for processing
    poller.change_message_visibility_timeout(msg, timeout)

    do_something2(msg)
  end
end
```

在 Amazon SQS 中重定向死信

下列範例會將含有 URL 之佇列中的任何無效字母重新導向URL至具有 ARN 的佇列。

```
# Copyright 2010-2019 Amazon.com, Inc. or its affiliates. All Rights Reserved.
#
# This file is licensed under the Apache License, Version 2.0 (the "License").
# You may not use this file except in compliance with the License. A copy of the
# License is located at
#
# http://aws.amazon.com/apache2.0/
#
# This file is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS
# OF ANY KIND, either express or implied. See the License for the specific
# language governing permissions and limitations under the License.

require 'aws-sdk-sqs' # v2: require 'aws-sdk'

sqs = Aws::SQS::Client.new(region: 'us-west-2')

sqs.set_queue_attributes({
  queue_url: URL,
  attributes:
    {
      'RedrivePolicy' => "{\"maxReceiveCount\": \"5\", \"deadLetterTargetArn\":
\"#{ARN}\"}"
    }
})
```

刪除 Amazon SQS 中的隊列

下列範例會刪除us-west-2區域URL中包含 URL 的 Amazon SQS 佇列。

```
# Copyright 2010-2019 Amazon.com, Inc. or its affiliates. All Rights Reserved.
#
# This file is licensed under the Apache License, Version 2.0 (the "License").
# You may not use this file except in compliance with the License. A copy of the
# License is located at
#
# http://aws.amazon.com/apache2.0/
#
# This file is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS
# OF ANY KIND, either express or implied. See the License for the specific
```

```
# language governing permissions and limitations under the License.

require 'aws-sdk-sqs' # v2: require 'aws-sdk'

sqs = Aws::SQS::Client.new(region: 'us-west-2')

sqs.delete_queue(queue_url: URL)
```

啟用資源以發佈到 Amazon SQS 中的佇列

下列範例可讓具有 ARN 的資源發佈 my-resource-arn 至區域中具有 ARN my-queue-arn 和 URL my-queue-url 的佇列。us-west-2

```
# Copyright 2010-2019 Amazon.com, Inc. or its affiliates. All Rights Reserved.
#
# This file is licensed under the Apache License, Version 2.0 (the "License").
# You may not use this file except in compliance with the License. A copy of the
# License is located at
#
# http://aws.amazon.com/apache2.0/
#
# This file is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS
# OF ANY KIND, either express or implied. See the License for the specific
# language governing permissions and limitations under the License.

require 'aws-sdk-sqs' # v2: require 'aws-sdk'

sqs = Aws::SQS::Client.new(region: 'us-west-2')

policy = '{
  "Version":"2008-10-17",
  "Id":"' + my-queue-arn + '/SQSDefaultPolicy",
  "Statement":[{
    "Sid":"__default_statement_ID",
    "Effect":"Allow",
    "Principal":{"
      "AWS": "*"
    },
    "Action":["SQS:SendMessage"],
    "Resource":"' + my-queue-arn + '",
    "Condition":{"
      "ArnEquals":{"
        "AWS:SourceArn":"' + my-resource-arn + '"}
```

```
    }
  }]
}'

sqs.set_queue_attributes({
  queue_url: my-queue-url,
  attributes: {
    Policy: policy
  }
})
```

在 Amazon SQS 中使用無效字母佇列

Amazon SQS 提供無效字母佇列的支援。無效字母佇列是其他 (來源) 佇列可以針對無法成功處理之郵件的目標佇列。您可以在無效字母佇列中保留並隔離這些郵件，以判斷其處理不成功的原因。如需無效字母佇列的詳細資訊，請參閱[使用 Amazon SQS 無效字母佇列](#)。

在此範例中，您可以使用適用於 Ruby 的 AWS 開發套件搭配 Amazon SQS 來執行下列作業：

1. 使用 `Aws::SQS::客戶端 #create_queue` 創建一個代表無效字母佇列的佇列。
2. 使用 `Aws::SQS::用戶端 #set_queue_attributes`，將無效字母佇列與現有佇列產生關聯。
3. 使用 `AWS::SQS::客戶端 #send_message` 將消息發送到現有佇列。
4. 使用 `Aws::SQS::輪詢佇列。QueuePoller`
5. 使用 `Aws::SQS::客戶端 #receive_message` 接收無效字母佇列中的消息。

先決條件

在執行範例程式碼之前，您必須先安裝並設定 AWS SDK for Ruby，如下所述：

- [安裝 AWS SDK for Ruby](#)
- [為紅寶石配置 AWS SDK](#)

您還需要使用 AWS Management Console 來創建現有的佇列，我的佇列。

Note

為了簡單起見，此示例代碼未演示 `Aws::SQS::客戶端 #add_permission`。在真實世界案例中，您應始終限制對諸如 `SendMessage`、`ReceiveMessage` `DeleteMessage`、和之類的動作的存取 `DeleteQueue`。否則可能會造成資訊洩漏、拒絕服務或將訊息插入佇列。

範例

```
# Copyright 2010-2019 Amazon.com, Inc. or its affiliates. All Rights Reserved.
#
# This file is licensed under the Apache License, Version 2.0 (the "License").
# You may not use this file except in compliance with the License. A copy of the
# License is located at
#
# http://aws.amazon.com/apache2.0/
#
# This file is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS
# OF ANY KIND, either express or implied. See the License for the specific
# language governing permissions and limitations under the License.

# Demonstrates how to:
# 1. Create a queue representing a dead letter queue.
# 2. Associate the dead letter queue with an existing queue.

require 'aws-sdk-sqs' # v2: require 'aws-sdk'

# Uncomment for Windows.
# Aws.use_bundled_cert!

sqs = Aws::SQS::Client.new(region: 'us-east-1')

# Create a queue representing a dead letter queue.
dead_letter_queue_name = "dead-letter-queue"

sqs.create_queue({
  queue_name: dead_letter_queue_name
})

# Get the dead letter queue's URL and ARN, so that you can associate it with an
# existing queue.
dead_letter_queue_url = sqs.get_queue_url(queue_name: dead_letter_queue_name).queue_url
```

```
dead_letter_queue_arn = sqs.get_queue_attributes({
  queue_url: dead_letter_queue_url,
  attribute_names: ["QueueArn"]
}).attributes["QueueArn"]

# Associate the dead letter queue with an existing queue.
begin
  queue_name = "my-queue"
  queue_url = sqs.get_queue_url(queue_name: queue_name).queue_url

  # Use a redrive policy to specify the dead letter queue and its behavior.
  redrive_policy = {
    "maxReceiveCount" => "5", # After the queue receives the same message 5 times, send
    that message to the dead letter queue.
    "deadLetterTargetArn" => dead_letter_queue_arn
  }.to_json

  sqs.set_queue_attributes({
    queue_url: queue_url,
    attributes: {
      "RedrivePolicy" => redrive_policy
    }
  })
end

rescue Aws::SQS::Errors::NonExistentQueue
  puts "A queue named '#{queue_name}' does not exist."
  exit(false)
end

# Send a message to the queue.
puts "Sending a message..."

sqs.send_message({
  queue_url: queue_url,
  message_body: "I hope I get moved to the dead letter queue."
})

30.downto(0) do |i|
  print "\rWaiting #{i} second(s) for sent message to be receivable..."
  sleep(1)
end

puts "\n"
```

```
poller = Aws::SQS::QueuePoller.new(queue_url)
# Receive 5 messages max and stop polling after 20 seconds of no received messages.
poller.poll(max_number_of_messages:5, idle_timeout: 20) do |messages|
  messages.each do |msg|
    puts "Received message ID: #{msg.message_id}"
  end
end

# Check to see if Amazon SQS moved the message to the dead letter queue.
receive_message_result = sqs.receive_message({
  queue_url: dead_letter_queue_url,
  max_number_of_messages: 1
})

if receive_message_result.messages.count > 0
  puts "\n#{receive_message_result.messages[0].body}"
else
  puts "\nNo messages received."
end
```

在 Amazon SQS 中指定訊息可見性逾時

在 Amazon SQS 中，在收到訊息後立即保留在佇列中。為了防止其他消費者再次處理訊息，Amazon SQS 會設定可見性逾時。這是 Amazon SQS 阻止其他消費元件接收和處理訊息的期間。如需進一步了解，請參閱[可見性逾時](#)。

在此範例中，您可以使用適用於 Ruby 的 AWS 開發套件搭配 Amazon SQS 來執行下列作業：

1. 使用 [AWS::SQS::](#) 客戶端 #get_queue_url 獲取現有佇列的網址。
2. 使用 [AWS::SQS::](#) 用戶端 #receive_message 接收最多 10 則訊息。
3. 使用 [AWS::SQS::](#) Client #change_message_visibility，指定收到訊息後不可見的時間間隔。

先決條件

在執行範例程式碼之前，您必須先安裝並設定 AWS SDK for Ruby，如下所述：

- [安裝 AWS SDK for Ruby](#)
- [為紅寶石配置 AWS SDK](#)

您還需要創建隊列我的隊列，您可以在 Amazon SQS 控制台中執行此操作。

範例

```
# Copyright 2010-2019 Amazon.com, Inc. or its affiliates. All Rights Reserved.
#
# This file is licensed under the Apache License, Version 2.0 (the "License").
# You may not use this file except in compliance with the License. A copy of the
# License is located at
#
# http://aws.amazon.com/apache2.0/
#
# This file is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS
# OF ANY KIND, either express or implied. See the License for the specific
# language governing permissions and limitations under the License.

# Demonstrates how to specify the time interval during which messages to a queue are
# not visible after being received.

require 'aws-sdk-sqs' # v2: require 'aws-sdk'

sqs = Aws::SQS::Client.new(region: 'us-east-1')

begin
  queue_name = "my-queue"
  queue_url = sqs.get_queue_url(queue_name: queue_name).queue_url

  receive_message_result_before = sqs.receive_message({
    queue_url: queue_url,
    max_number_of_messages: 10 # Receive up to 10 messages, if there are that many.
  })

  puts "Before attempting to change message visibility timeout: received
#{receive_message_result_before.messages.count} message(s)."

  receive_message_result_before.messages.each do |message|
    sqs.change_message_visibility({
      queue_url: queue_url,
      receipt_handle: message.receipt_handle,
      visibility_timeout: 30 # This message will not be visible for 30 seconds after
first receipt.
    })
  end
end
```

```
# Try to retrieve the original messages after setting their visibility timeout.
receive_message_result_after = sqs.receive_message({
  queue_url: queue_url,
  max_number_of_messages: 10
})

puts "\nAfter attempting to change message visibility timeout: received
#{receive_message_result_after.messages.count} message(s)."

rescue Aws::SQS::Errors::NonExistentQueue
  puts "Cannot receive messages for a queue named '#{receive_queue_name}', as it does
not exist."
end
```

亞馬遜 WorkDocs 示例

您可以使用下面的例子來訪問亞馬遜 WorkDocs (亞馬遜 WorkDocs) 使用 AWS SDK for Ruby。有關亞馬遜的更多信息 WorkDocs , 請參閱[亞馬遜 WorkDocs 文檔](#)。

您需要您的組織 ID 才能使用這些範例。使用下列步驟從主AWS控制台取得組織 ID :

- 選取 AWS Directory Service
- 選取 Directories

組織 ID 是Directory ID對應於您的亞馬遜 WorkDocs 網站。

範例

主題

- [列出使用者](#)
- [列出使用者文件](#)

列出使用者

下列範例會列出組織中所有使用者的名稱、電子郵件地址和根資料夾。選Copy擇在本機儲存程式碼 , 或在本主題結尾查看完整範例的連結。

1. 需要 Ruby 模塊的 AWS SDK 並創建一個亞馬遜 WorkDocs 客戶端。

2. 使 `describe_users` 用您的組織 ID 撥打電話，並按升序獲取所有用戶名。

1. 顯示有關使用者的資訊。

```
# Copyright 2010-2019 Amazon.com, Inc. or its affiliates. All Rights Reserved.
#
# This file is licensed under the Apache License, Version 2.0 (the "License").
# You may not use this file except in compliance with the License. A copy of the
# License is located at
#
# http://aws.amazon.com/apache2.0/
#
# This file is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS
# OF ANY KIND, either express or implied. See the License for the specific
# language governing permissions and limitations under the License.

require 'aws-sdk-workdocs' # v2: require 'aws-sdk'

client = Aws::WorkDocs::Client.new(region: 'us-west-2')

# Set to the OrganizationId of your WorkDocs site
orgId = 'd-123456789c'

resp = client.describe_users({
  organization_id: orgId,
  include: "ALL", # accepts ALL, ACTIVE_PENDING
  order: "ASCENDING", # accepts ASCENDING, DESCENDING
  sort: "USER_NAME", # accepts USER_NAME, FULL_NAME, STORAGE_LIMIT, USER_STATUS,
  STORAGE_USED
})

resp.users.each do |user|
  puts "First name:  #{user.given_name}"
  puts "Last name:   #{user.surname}"
  puts "Email:        #{user.email_address}"
  puts "Root folder:  #{user.root_folder_id}"
  puts
end
```

請參閱 (詳見) 的 [完整實例](#) GitHub。

列出使用者文件

下列範例會列出使用者的文件。選Copy擇在本機儲存程式碼，或在本主題結尾查看完整範例的連結。

1. 需要紅寶石模塊的 AWS SDK。
2. 創建一個幫助程序方法來獲取用戶的根文件夾。
3. 創建一個亞馬遜 WorkDocs 客戶端。
4. 取得該使用者的根資料夾。
5. 呼叫describe_folder_contents以按升序獲取文件夾的內容。
6. 顯示用戶根文件夾中每個文檔的名稱，大小（以字節為單位），上次修改日期，文檔 ID 和版本 ID。

```
# Copyright 2010-2019 Amazon.com, Inc. or its affiliates. All Rights Reserved.
#
# This file is licensed under the Apache License, Version 2.0 (the "License").
# You may not use this file except in compliance with the License. A copy of the
# License is located at
#
# http://aws.amazon.com/apache2.0/
#
# This file is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS
# OF ANY KIND, either express or implied. See the License for the specific
# language governing permissions and limitations under the License.

require 'aws-sdk-workdocs' # v2: require 'aws-sdk'

def get_user_folder(client, orgId, user_email)
  root_folder = ''

  resp = client.describe_users({
    organization_id: orgId,
  })

  # resp.users should have only one entry
  resp.users.each do |user|
    if user.email_address == user_email
      root_folder = user.root_folder_id
    end
  end
end
```

```
    return root_folder
  end

  client = Aws::WorkDocs::Client.new(region: 'us-west-2')

  # Set to the email address of a user
  user_email = 'someone@somewhere'

  # Set to the OrganizationId of your WorkDocs site.
  orgId = 'd-123456789c'

  user_folder = get_user_folder(client, orgId, user_email)

  if user_folder == ''
    puts 'Could not get root folder for user with email address ' + user_email
    exit(1)
  end

  resp = client.describe_folder_contents({
    folder_id: user_folder, # required
    sort: "NAME", # accepts DATE, NAME
    order: "ASCENDING", # accepts ASCENDING, DESCENDING
  })

  resp.documents.each do |doc|
    md = doc.latest_version_metadata

    puts "Name:           #{md.name}"
    puts "Size (bytes):    #{md.size}"
    puts "Last modified:    #{doc.modified_timestamp}"
    puts "Doc ID:           #{doc.id}"
    puts "Version ID:      #{md.id}"
    puts
  end
end
```

請參閱 (詳見) 的[完整實例](#) GitHub。

適用於 Ruby 程式碼範例的 SDK

本主題中的程式碼範例會示範如何使用 AWS SDK for Ruby 與 AWS。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境和跨服務範例中查看內容中的動作。

案例是向您展示如何呼叫相同服務中的多個函數來完成特定任務的程式碼範例。

Cross-service examples (跨服務範例) 是跨多個 AWS 服務執行的應用程式範例。

範例

- [使用 SDK 進行紅寶石的動作和案例](#)
- [使用 SDK for Ruby 跨服務範例](#)

使用 SDK 進行紅寶石的動作和案例

下列程式碼範例會示範如何使用 AWS SDK for Ruby 與來執行動作及實作常見案例 AWS 服務。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境和跨服務範例中查看內容中的動作。

案例是向您展示如何呼叫相同服務中的多個函數來完成特定任務的程式碼範例。

服務

- [CloudTrail 使用 SDK for Ruby 示例](#)
- [CloudWatch 使用 SDK for Ruby 示例](#)
- [使用 SDK for Ruby DynamoDB 範例](#)
- [Amazon EC2 示例使用 SDK for Ruby](#)
- [使用 SDK for Ruby Elastic Beanstalk 示例](#)
- [EventBridge 使用 SDK for Ruby 示例](#)
- [AWS Glue 使用 SDK for Ruby 示例](#)
- [使用 SDK for Ruby IAM 範例](#)
- [Kinesis 示例使用 SDK for Ruby](#)
- [AWS KMS 使用 SDK for Ruby 示例](#)

- [使用 SDK for Ruby Lambda](#)
- [Amazon Polly 示例使用 SDK for Ruby](#)
- [Amazon RDS 示例使用 SDK for Ruby](#)
- [Amazon S3 示例使用 SDK for Ruby](#)
- [Amazon SES 示例使用 SDK for Ruby](#)
- [Amazon SES API v2 示例使用 SDK for Ruby](#)
- [使用 SDK for Ruby Amazon SNS 範例](#)
- [Amazon SQS 示例使用 SDK for Ruby](#)
- [AWS STS 使用 SDK for Ruby 示例](#)
- [Amazon WorkDocs 示例使用 SDK for Ruby](#)

CloudTrail 使用 SDK for Ruby 示例

下列程式碼範例說明如何使用 AWS SDK for Ruby 與來執行動作及實作常見案例 CloudTrail。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境和跨服務範例中查看內容中的動作。

案例是向您展示如何呼叫相同服務中的多個函數來完成特定任務的程式碼範例。

每個範例都包含一個連結 GitHub，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- [動作](#)

動作

建立軌跡

下列程式碼範例顯示如何建立 AWS CloudTrail 追蹤。

適用於 Ruby 的開發套件

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫](#)。

```
require "aws-sdk-cloudtrail" # v2: require 'aws-sdk'
require "aws-sdk-s3"
require "aws-sdk-sts"

def create_trail_example(s3_client, sts_client, cloudtrail_client, trail_name,
  bucket_name)

  resp = sts_client.get_caller_identity({})
  account_id = resp.account

  # Attach policy to an Amazon Simple Storage Service (S3) bucket.
  s3_client.create_bucket(bucket: bucket_name)
  begin
    policy = {
      "Version" => "2012-10-17",
      "Statement" => [
        {
          "Sid" => "AWSCloudTrailAclCheck20150319",
          "Effect" => "Allow",
          "Principal" => {
            "Service" => "cloudtrail.amazonaws.com"
          },
          "Action" => "s3:GetBucketAcl",
          "Resource" => "arn:aws:s3:::#{bucket_name}"
        },
        {
          "Sid" => "AWSCloudTrailWrite20150319",
          "Effect" => "Allow",
          "Principal" => {
            "Service" => "cloudtrail.amazonaws.com"
          },
          "Action" => "s3:PutObject",
          "Resource" => "arn:aws:s3:::#{bucket_name}/AWSLogs/#{account_id}/*",
          "Condition" => {
            "StringEquals" => {
              "s3:x-amz-acl" => "bucket-owner-full-control"
            }
          }
        }
      ]
    }.to_json

    s3_client.put_bucket_policy(
```



```

    bucket: bucket_name,
    policy: policy
  )
  puts "Successfully added policy to bucket #{bucket_name}"
end

begin
  cloudtrail_client.create_trail({
                                name: trail_name, # required
                                s3_bucket_name: bucket_name # required
  })

  puts "Successfully created trail: #{trail_name}."
rescue StandardError => e
  puts "Got error trying to create trail #{trail_name}:\n #{e}"
  puts e
  exit 1
end

```

- 如需 API 詳細資訊，請參閱 AWS SDK for Ruby API 參考 [CreateTrail](#) 中的。

刪除軌跡

下列程式碼範例顯示如何刪除 AWS CloudTrail 追蹤。

適用於 Ruby 的開發套件

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在 [AWS 設定和執行程式碼範例儲存庫](#)。

```

client.delete_trail({
                  name: trail_name # required
                })
puts "Successfully deleted trail: " + trail_name
rescue StandardError => err
puts "Got error trying to delete trail: " + trail_name + ":"
puts err
exit 1
end

```

- 如需 API 詳細資訊，請參閱 AWS SDK for Ruby API 參考[DeleteTrail](#)中的。

列出路徑事件

下列程式碼範例顯示如何列出 AWS CloudTrail 追蹤事件。

適用於 Ruby 的開發套件

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫](#)。

```
require "aws-sdk-cloudtrail" # v2: require 'aws-sdk'

# @param [Object] client
def lookup_events_example(client)
  resp = client.lookup_events
  puts "Found #{resp.events.count} events:"
  resp.events.each do |e|
    puts "Event name:   #{e.event_name}"
    puts "Event ID:     #{e.event_id}"
    puts "Event time:    #{e.event_time}"
    puts "Resources:"

    e.resources.each do |r|
      puts "  Name:       #{r.resource_name}"
      puts "  Type:       #{r.resource_type}"
      puts ""
    end
  end
end
```

- 如需 API 詳細資訊，請參閱 AWS SDK for Ruby API 參考[LookupEvents](#)中的。

列出路徑

下列程式碼範例顯示如何列出 AWS CloudTrail 追蹤。

適用於 Ruby 的開發套件

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫](#)。

```
require "aws-sdk-cloudtrail" # v2: require 'aws-sdk'

def describe_trails_example(client)
  resp = client.describe_trails({})
  puts "Found #{resp.trail_list.count} trail(s)."

  resp.trail_list.each do |trail|
    puts "Name:          " + trail.name
    puts "S3 bucket name: " + trail.s3_bucket_name
    puts
  end
end
```

- 如需 API 詳細資訊，請參閱 AWS SDK for Ruby API 參考[ListTrails](#)中的。

CloudWatch 使用 SDK for Ruby 示例

下列程式碼範例說明如何使用 AWS SDK for Ruby 與來執行動作及實作常見案例 CloudWatch。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境和跨服務範例中查看內容中的動作。

案例是向您展示如何呼叫相同服務中的多個函數來完成特定任務的程式碼範例。

每個範例都包含一個連結 GitHub，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- [動作](#)

動作

建立指標警示

下列程式碼範例顯示如何建立或更新 Amazon CloudWatch 警示，並將其與指定的量度、量度數學運算式、異常偵測模型或指標洞見查詢建立關聯。

適用於 Ruby 的開發套件

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫](#)。

```
# Creates or updates an alarm in Amazon CloudWatch.
#
# @param cloudwatch_client [Aws::CloudWatch::Client]
#   An initialized CloudWatch client.
# @param alarm_name [String] The name of the alarm.
# @param alarm_description [String] A description about the alarm.
# @param metric_name [String] The name of the metric associated with the alarm.
# @param alarm_actions [Array] A list of Strings representing the
#   Amazon Resource Names (ARNs) to execute when the alarm transitions to the
#   ALARM state.
# @param namespace [String] The namespace for the metric to alarm on.
# @param statistic [String] The statistic for the metric.
# @param dimensions [Array] A list of dimensions for the metric, specified as
#   Aws::CloudWatch::Types::Dimension.
# @param period [Integer] The number of seconds before re-evaluating the metric.
# @param unit [String] The unit of measure for the statistic.
# @param evaluation_periods [Integer] The number of periods over which data is
#   compared to the specified threshold.
# @param threshold [Float] The value against which the specified statistic is
#   compared.
# @param comparison_operator [String] The arithmetic operation to use when
#   comparing the specified statistic and threshold.
# @return [Boolean] true if the alarm was created or updated; otherwise, false.
# @example
#   exit 1 unless alarm_created_or_updated?(
#     Aws::CloudWatch::Client.new(region: 'us-east-1'),
#     'ObjectsInBucket',
#     'Objects exist in this bucket for more than 1 day.',
#     'NumberOfObjects',
```

```
# ['arn:aws:sns:us-east-1:111111111111:Default_CloudWatch_Alarms_Topic'],
# 'AWS/S3',
# 'Average',
# [
#   {
#     name: 'BucketName',
#     value: 'doc-example-bucket'
#   },
#   {
#     name: 'StorageType',
#     value: 'AllStorageTypes'
#   }
# ],
# 86_400,
# 'Count',
# 1,
# 1,
# 'GreaterThanThreshold'
# )
def alarm_created_or_updated?(
  cloudwatch_client,
  alarm_name,
  alarm_description,
  metric_name,
  alarm_actions,
  namespace,
  statistic,
  dimensions,
  period,
  unit,
  evaluation_periods,
  threshold,
  comparison_operator
)
  cloudwatch_client.put_metric_alarm(
    alarm_name: alarm_name,
    alarm_description: alarm_description,
    metric_name: metric_name,
    alarm_actions: alarm_actions,
    namespace: namespace,
    statistic: statistic,
    dimensions: dimensions,
    period: period,
    unit: unit,
```

```
    evaluation_periods: evaluation_periods,
    threshold: threshold,
    comparison_operator: comparison_operator
  )
  return true
rescue StandardError => e
  puts "Error creating alarm: #{e.message}"
  return false
end
```

- 如需 API 詳細資訊，請參閱 AWS SDK for Ruby API 參考[PutMetricAlarm](#)中的。

描述警示

下列程式碼範例顯示如何描述 Amazon CloudWatch 警示。

適用於 Ruby 的開發套件

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫](#)。

```
require "aws-sdk-cloudwatch"

# Lists the names of available Amazon CloudWatch alarms.
#
# @param cloudwatch_client [Aws::CloudWatch::Client]
#   An initialized CloudWatch client.
# @example
#   list_alarms(Aws::CloudWatch::Client.new(region: 'us-east-1'))
def list_alarms(cloudwatch_client)
  response = cloudwatch_client.describe_alarms
  if response.metric_alarms.count.positive?
    response.metric_alarms.each do |alarm|
      puts alarm.alarm_name
    end
  else
    puts "No alarms found."
  end
end
rescue StandardError => e
```

```
puts "Error getting information about alarms: #{e.message}"
end
```

- 如需 API 詳細資訊，請參閱 AWS SDK for Ruby API 參考 [DescribeAlarms](#) 中的。

描述指標的警示

下列程式碼範例顯示如何描述指標的 Amazon CloudWatch 警示。

適用於 Ruby 的開發套件

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在 [AWS 設定和執行程式碼範例儲存庫](#)。

```
#
# @param cloudwatch_client [Aws::CloudWatch::Client]
#   An initialized CloudWatch client.
# @example
#   describe_metric_alarms(Aws::CloudWatch::Client.new(region: 'us-east-1'))
def describe_metric_alarms(cloudwatch_client)
  response = cloudwatch_client.describe_alarms

  if response.metric_alarms.count.positive?
    response.metric_alarms.each do |alarm|
      puts "-" * 16
      puts "Name:           " + alarm.alarm_name
      puts "State value:      " + alarm.state_value
      puts "State reason:     " + alarm.state_reason
      puts "Metric:           " + alarm.metric_name
      puts "Namespace:        " + alarm.namespace
      puts "Statistic:         " + alarm.statistic
      puts "Period:           " + alarm.period.to_s
      puts "Unit:              " + alarm.unit.to_s
      puts "Eval. periods:    " + alarm.evaluation_periods.to_s
      puts "Threshold:         " + alarm.threshold.to_s
      puts "Comp. operator:   " + alarm.comparison_operator

      if alarm.key?(:ok_actions) && alarm.ok_actions.count.positive?
```

```
    puts "OK actions:"
    alarm.ok_actions.each do |a|
      puts "  " + a
    end
  end

  if alarm.key?(:alarm_actions) && alarm.alarm_actions.count.positive?
    puts "Alarm actions:"
    alarm.alarm_actions.each do |a|
      puts "  " + a
    end
  end

  if alarm.key?(:insufficient_data_actions) &&
    alarm.insufficient_data_actions.count.positive?
    puts "Insufficient data actions:"
    alarm.insufficient_data_actions.each do |a|
      puts "  " + a
    end
  end

  puts "Dimensions:"
  if alarm.key?(:dimensions) && alarm.dimensions.count.positive?
    alarm.dimensions.each do |d|
      puts "  Name: " + d.name + ", Value: " + d.value
    end
  else
    puts "  None for this alarm."
  end
end
else
  puts "No alarms found."
end
rescue StandardError => e
  puts "Error getting information about alarms: #{e.message}"
end

# Example usage:
def run_me
  region = ""

  # Print usage information and then stop.
  if ARGV[0] == "--help" || ARGV[0] == "-h"
    puts "Usage:  ruby cw-ruby-example-show-alarms.rb REGION"
```



```

    puts "Example: ruby cw-ruby-example-show-alarms.rb us-east-1"
    exit 1
# If no values are specified at the command prompt, use these default values.
elsif ARGV.count.zero?
    region = "us-east-1"
# Otherwise, use the values as specified at the command prompt.
else
    region = ARGV[0]
end

cloudwatch_client = Aws::CloudWatch::Client.new(region: region)
puts "Available alarms:"
describe_metric_alarms(cloudwatch_client)
end

run_me if $PROGRAM_NAME == __FILE__

```

- 如需 API 詳細資訊，請參閱 AWS SDK for Ruby API 參考 [DescribeAlarmsForMetric](#) 中的。

停用警示動作

下列程式碼範例顯示如何停用 Amazon CloudWatch 警示動作。

適用於 Ruby 的開發套件

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在 [AWS 設定和執行程式碼範例儲存庫](#)。

```

# Disables an alarm in Amazon CloudWatch.
#
# Prerequisites.
#
# - The alarm to disable.
#
# @param cloudwatch_client [Aws::CloudWatch::Client]
#   An initialized CloudWatch client.
# @param alarm_name [String] The name of the alarm to disable.
# @return [Boolean] true if the alarm was disabled; otherwise, false.

```

```
# @example
#   exit 1 unless alarm_actions_disabled?(
#     Aws::CloudWatch::Client.new(region: 'us-east-1'),
#     'ObjectsInBucket'
#   )
def alarm_actions_disabled?(cloudwatch_client, alarm_name)
  cloudwatch_client.disable_alarm_actions(alarm_names: [alarm_name])
  return true
rescue StandardError => e
  puts "Error disabling alarm actions: #{e.message}"
  return false
end

# Example usage:
def run_me
  alarm_name = "ObjectsInBucket"
  alarm_description = "Objects exist in this bucket for more than 1 day."
  metric_name = "NumberOfObjects"
  # Notify this Amazon Simple Notification Service (Amazon SNS) topic when
  # the alarm transitions to the ALARM state.
  alarm_actions = ["arn:aws:sns:us-
east-1:111111111111:Default_CloudWatch_Alarms_Topic"]
  namespace = "AWS/S3"
  statistic = "Average"
  dimensions = [
    {
      name: "BucketName",
      value: "doc-example-bucket"
    },
    {
      name: "StorageType",
      value: "AllStorageTypes"
    }
  ]
  period = 86_400 # Daily (24 hours * 60 minutes * 60 seconds = 86400 seconds).
  unit = "Count"
  evaluation_periods = 1 # More than one day.
  threshold = 1 # One object.
  comparison_operator = "GreaterThanThreshold" # More than one object.
  # Replace us-west-2 with the AWS Region you're using for Amazon CloudWatch.
  region = "us-east-1"

  cloudwatch_client = Aws::CloudWatch::Client.new(region: region)
```

```
if alarm_created_or_updated?(
  cloudwatch_client,
  alarm_name,
  alarm_description,
  metric_name,
  alarm_actions,
  namespace,
  statistic,
  dimensions,
  period,
  unit,
  evaluation_periods,
  threshold,
  comparison_operator
)
  puts "Alarm '#{alarm_name}' created or updated."
else
  puts "Could not create or update alarm '#{alarm_name}'."
end

if alarm_actions_disabled?(cloudwatch_client, alarm_name)
  puts "Alarm '#{alarm_name}' disabled."
else
  puts "Could not disable alarm '#{alarm_name}'."
end
end

run_me if $PROGRAM_NAME == __FILE__
```

- 如需 API 詳細資訊，請參閱 AWS SDK for Ruby API 參考[DisableAlarmActions](#)中的。

列出指標

下列程式碼範例顯示如何列出 Amazon CloudWatch 指標的中繼資料。若要取得量度的資料，請使用 `GetMetricData` 或 `GetMetricStatistics` 動作。

適用於 Ruby 的開發套件

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫](#)。

```
# Lists available metrics for a metric namespace in Amazon CloudWatch.
#
# @param cloudwatch_client [Aws::CloudWatch::Client]
#   An initialized CloudWatch client.
# @param metric_namespace [String] The namespace of the metric.
# @example
#   list_metrics_for_namespace(
#     Aws::CloudWatch::Client.new(region: 'us-east-1'),
#     'SITE/TRAFFIC'
#   )
def list_metrics_for_namespace(cloudwatch_client, metric_namespace)
  response = cloudwatch_client.list_metrics(namespace: metric_namespace)

  if response.metrics.count.positive?
    response.metrics.each do |metric|
      puts "  Metric name: #{metric.metric_name}"
      if metric.dimensions.count.positive?
        puts "    Dimensions:"
        metric.dimensions.each do |dimension|
          puts "      Name: #{dimension.name}, Value: #{dimension.value}"
        end
      else
        puts "No dimensions found."
      end
    end
  else
    puts "No metrics found for namespace '#{metric_namespace}'. " \
      "Note that it could take up to 15 minutes for recently-added metrics " \
      "to become available."
  end
end

# Example usage:
def run_me
  metric_namespace = "SITE/TRAFFIC"
  # Replace us-west-2 with the AWS Region you're using for Amazon CloudWatch.
  region = "us-east-1"

  cloudwatch_client = Aws::CloudWatch::Client.new(region: region)

  # Add three datapoints.
  puts "Continuing..." unless datapoint_added_to_metric?(
    cloudwatch_client,
```

```
metric_namespace,  
"UniqueVisitors",  
"SiteName",  
"example.com",  
5_885.0,  
"Count"  
)  
  
puts "Continuing..." unless datapoint_added_to_metric?(  
  cloudwatch_client,  
  metric_namespace,  
  "UniqueVisits",  
  "SiteName",  
  "example.com",  
  8_628.0,  
  "Count"  
)  
  
puts "Continuing..." unless datapoint_added_to_metric?(  
  cloudwatch_client,  
  metric_namespace,  
  "PageViews",  
  "PageURL",  
  "example.html",  
  18_057.0,  
  "Count"  
)  
  
puts "Metrics for namespace '#{metric_namespace}':"  
list_metrics_for_namespace(cloudwatch_client, metric_namespace)  
end  
  
run_me if $PROGRAM_NAME == __FILE__
```

- 如需 API 詳細資訊，請參閱 AWS SDK for Ruby API 參考 [ListMetrics](#) 中的。

將資料放入指標

下列程式碼範例顯示如何將指標資料點發佈到 Amazon CloudWatch。

適用於 Ruby 的開發套件

 Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫](#)。

```
require "aws-sdk-cloudwatch"

# Adds a datapoint to a metric in Amazon CloudWatch.
#
# @param cloudwatch_client [Aws::CloudWatch::Client]
#   An initialized CloudWatch client.
# @param metric_namespace [String] The namespace of the metric to add the
#   datapoint to.
# @param metric_name [String] The name of the metric to add the datapoint to.
# @param dimension_name [String] The name of the dimension to add the
#   datapoint to.
# @param dimension_value [String] The value of the dimension to add the
#   datapoint to.
# @param metric_value [Float] The value of the datapoint.
# @param metric_unit [String] The unit of measurement for the datapoint.
# @return [Boolean]
# @example
#   exit 1 unless datapoint_added_to_metric?(
#     Aws::CloudWatch::Client.new(region: 'us-east-1'),
#     'SITE/TRAFFIC',
#     'UniqueVisitors',
#     'SiteName',
#     'example.com',
#     5_885.0,
#     'Count'
#   )
def datapoint_added_to_metric?(
  cloudwatch_client,
  metric_namespace,
  metric_name,
  dimension_name,
  dimension_value,
  metric_value,
  metric_unit
)
  cloudwatch_client.put_metric_data(
```

```
namespace: metric_namespace,
metric_data: [
  {
    metric_name: metric_name,
    dimensions: [
      {
        name: dimension_name,
        value: dimension_value
      }
    ],
    value: metric_value,
    unit: metric_unit
  }
]
)
puts "Added data about '#{metric_name}' to namespace " \
    "'#{metric_namespace}'."
return true
rescue StandardError => e
  puts "Error adding data about '#{metric_name}' to namespace " \
    "'#{metric_namespace}': #{e.message}"
  return false
end
```

- 如需 API 詳細資訊，請參閱 AWS SDK for Ruby API 參考[PutMetricData](#)中的。

使用 SDK for Ruby DynamoDB 範例

下列程式碼範例說明如何透過 AWS SDK for Ruby 與 DynamoDB 搭配使用來執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境和跨服務範例中查看內容中的動作。

案例是向您展示如何呼叫相同服務中的多個函數來完成特定任務的程式碼範例。

每個範例都包含一個連結 GitHub，您可以在其中找到如何在內容中設定和執程式碼的指示。

主題

- [動作](#)

- [案例](#)

動作

建立資料表

下列程式碼範例示範如何建立 DynamoDB 資料表。

適用於 Ruby 的開發套件

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫](#)。

```
# Encapsulates an Amazon DynamoDB table of movie data.
class Scaffold
  attr_reader :dynamo_resource
  attr_reader :table_name
  attr_reader :table

  def initialize(table_name)
    client = Aws::DynamoDB::Client.new(region: "us-east-1")
    @dynamo_resource = Aws::DynamoDB::Resource.new(client: client)
    @table_name = table_name
    @table = nil
    @logger = Logger.new($stdout)
    @logger.level = Logger::DEBUG
  end

  # Creates an Amazon DynamoDB table that can be used to store movie data.
  # The table uses the release year of the movie as the partition key and the
  # title as the sort key.
  #
  # @param table_name [String] The name of the table to create.
  # @return [Aws::DynamoDB::Table] The newly created table.
  def create_table(table_name)
    @table = @dynamo_resource.create_table(
      table_name: table_name,
      key_schema: [
        {attribute_name: "year", key_type: "HASH"}, # Partition key
        {attribute_name: "title", key_type: "RANGE"} # Sort key
      ]
    )
  end
end
```



```

    ],
    attribute_definitions: [
      {attribute_name: "year", attribute_type: "N"},
      {attribute_name: "title", attribute_type: "S"}
    ],
    provisioned_throughput: {read_capacity_units: 10, write_capacity_units: 10})
@dynamo_resource.client.wait_until(:table_exists, table_name: table_name)
@table
rescue Aws::DynamoDB::Errors::ServiceError => e
  @logger.error("Failed create table #{table_name}:\n#{e.code}: #{e.message}")
  raise
end

```

- 如需 API 詳細資訊，請參閱 AWS SDK for Ruby API 參考[CreateTable](#)中的。

刪除資料表

下列程式碼範例示範如何刪除 DynamoDB 資料表。

適用於 Ruby 的開發套件

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫](#)。

```

# Encapsulates an Amazon DynamoDB table of movie data.
class Scaffold
  attr_reader :dynamo_resource
  attr_reader :table_name
  attr_reader :table

  def initialize(table_name)
    client = Aws::DynamoDB::Client.new(region: "us-east-1")
    @dynamo_resource = Aws::DynamoDB::Resource.new(client: client)
    @table_name = table_name
    @table = nil
    @logger = Logger.new($stdout)
    @logger.level = Logger::DEBUG
  end
end

```

```
# Deletes the table.
def delete_table
  @table.delete
  @table = nil
rescue Aws::DynamoDB::Errors::ServiceError => e
  puts("Couldn't delete table. Here's why:")
  puts("\t#{e.code}: #{e.message}")
  raise
end
```

- 如需 API 詳細資訊，請參閱 AWS SDK for Ruby API 參考[DeleteTable](#)中的。

從資料表刪除項目

下列程式碼範例示範如何從 DynamoDB 資料表中刪除項目。

適用於 Ruby 的開發套件

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫](#)。

```
class DynamoDBBasics
  attr_reader :dynamo_resource
  attr_reader :table

  def initialize(table_name)
    client = Aws::DynamoDB::Client.new(region: "us-east-1")
    @dynamo_resource = Aws::DynamoDB::Resource.new(client: client)
    @table = @dynamo_resource.table(table_name)
  end

  # Deletes a movie from the table.
  #
  # @param title [String] The title of the movie to delete.
  # @param year [Integer] The release year of the movie to delete.
  def delete_item(title, year)
    @table.delete_item(key: {"year" => year, "title" => title})
  rescue Aws::DynamoDB::Errors::ServiceError => e
    puts("Couldn't delete movie #{title}. Here's why:")
  end
end
```

```
puts("\t#{e.code}: #{e.message}")
raise
end
```

- 如需 API 詳細資訊，請參閱 AWS SDK for Ruby API 參考[DeleteItem](#)中的。

從資料表取得項目

下列程式碼範例示範如何從 DynamoDB 資料表取得項目。

適用於 Ruby 的開發套件

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫](#)。

```
class DynamoDBBasics
  attr_reader :dynamo_resource
  attr_reader :table

  def initialize(table_name)
    client = Aws::DynamoDB::Client.new(region: "us-east-1")
    @dynamo_resource = Aws::DynamoDB::Resource.new(client: client)
    @table = @dynamo_resource.table(table_name)
  end

  # Gets movie data from the table for a specific movie.
  #
  # @param title [String] The title of the movie.
  # @param year [Integer] The release year of the movie.
  # @return [Hash] The data about the requested movie.
  def get_item(title, year)
    @table.get_item(key: {"year" => year, "title" => title})
  rescue Aws::DynamoDB::Errors::ServiceError => e
    puts("Couldn't get movie #{title} (#{year}) from table #{@table.name}:\n")
    puts("\t#{e.code}: #{e.message}")
    raise
  end
end
```

- 如需 API 詳細資訊，請參閱 AWS SDK for Ruby API 參考 [GetItem](#) 中的。

取得資料表的相關資訊

下列程式碼範例示範如何取得 DynamoDB 資料表的相關資訊。

適用於 Ruby 的開發套件

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在 [AWS 設定和執行程式碼範例儲存庫](#)。

```
# Encapsulates an Amazon DynamoDB table of movie data.
class Scaffold
  attr_reader :dynamo_resource
  attr_reader :table_name
  attr_reader :table

  def initialize(table_name)
    client = Aws::DynamoDB::Client.new(region: "us-east-1")
    @dynamo_resource = Aws::DynamoDB::Resource.new(client: client)
    @table_name = table_name
    @table = nil
    @logger = Logger.new($stdout)
    @logger.level = Logger::DEBUG
  end

  # Determines whether a table exists. As a side effect, stores the table in
  # a member variable.
  #
  # @param table_name [String] The name of the table to check.
  # @return [Boolean] True when the table exists; otherwise, False.
  def exists?(table_name)
    @dynamo_resource.client.describe_table(table_name: table_name)
    @logger.debug("Table #{table_name} exists")
  rescue Aws::DynamoDB::Errors::ResourceNotFoundException
    @logger.debug("Table #{table_name} doesn't exist")
    false
  rescue Aws::DynamoDB::Errors::ServiceError => e
    puts("Couldn't check for existence of #{table_name}:\n")
    puts("\t#{e.code}: #{e.message}")
  end
end
```

```
    raise
  end
```

- 如需 API 詳細資訊，請參閱 AWS SDK for Ruby API 參考 [DescribeTable](#) 中的。

列出資料表

下列程式碼範例示範如何列出 DynamoDB 資料表。

適用於 Ruby 的開發套件

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在 [AWS 設定和執行程式碼範例儲存庫](#)。

判斷資料表是否存在。

```
# Encapsulates an Amazon DynamoDB table of movie data.
class Scaffold
  attr_reader :dynamo_resource
  attr_reader :table_name
  attr_reader :table

  def initialize(table_name)
    client = Aws::DynamoDB::Client.new(region: "us-east-1")
    @dynamo_resource = Aws::DynamoDB::Resource.new(client: client)
    @table_name = table_name
    @table = nil
    @logger = Logger.new($stdout)
    @logger.level = Logger::DEBUG
  end

  # Determines whether a table exists. As a side effect, stores the table in
  # a member variable.
  #
  # @param table_name [String] The name of the table to check.
  # @return [Boolean] True when the table exists; otherwise, False.
  def exists?(table_name)
    @dynamo_resource.client.describe_table(table_name: table_name)
    @logger.debug("Table #{table_name} exists")
  end
end
```

```

rescue Aws::DynamoDB::Errors::ResourceNotFoundException
  @logger.debug("Table #{table_name} doesn't exist")
  false
rescue Aws::DynamoDB::Errors::ServiceError => e
  puts("Couldn't check for existence of #{table_name}:\n")
  puts("\t#{e.code}: #{e.message}")
  raise
end

```

- 如需 API 詳細資訊，請參閱 AWS SDK for Ruby API 參考 [ListTables](#) 中的。

將項目放入資料表

下列程式碼範例示範如何將項目放入 DynamoDB 資料表。

適用於 Ruby 的開發套件

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在 [AWS 設定和執行程式碼範例儲存庫](#)。

```

class DynamoDBBasics
  attr_reader :dynamo_resource
  attr_reader :table

  def initialize(table_name)
    client = Aws::DynamoDB::Client.new(region: "us-east-1")
    @dynamo_resource = Aws::DynamoDB::Resource.new(client: client)
    @table = @dynamo_resource.table(table_name)
  end

  # Adds a movie to the table.
  #
  # @param movie [Hash] The title, year, plot, and rating of the movie.
  def add_item(movie)
    @table.put_item(
      item: {
        "year" => movie[:year],
        "title" => movie[:title],
        "info" => {"plot" => movie[:plot], "rating" => movie[:rating]})
  end
end

```

```
rescue Aws::DynamoDB::Errors::ServiceError => e
  puts("Couldn't add movie #{title} to table #{@table.name}. Here's why:")
  puts("\t#{e.code}: #{e.message}")
  raise
end
```

- 如需 API 詳細資訊，請參閱 AWS SDK for Ruby API 參考 [PutItem](#) 中的。

查詢資料表

下列程式碼範例示範如何查詢 DynamoDB 資料表。

適用於 Ruby 的開發套件

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在 [AWS 設定和執行程式碼範例儲存庫](#)。

```
class DynamoDBBasics
  attr_reader :dynamo_resource
  attr_reader :table

  def initialize(table_name)
    client = Aws::DynamoDB::Client.new(region: "us-east-1")
    @dynamo_resource = Aws::DynamoDB::Resource.new(client: client)
    @table = @dynamo_resource.table(table_name)
  end

  # Queries for movies that were released in the specified year.
  #
  # @param year [Integer] The year to query.
  # @return [Array] The list of movies that were released in the specified year.
  def query_items(year)
    response = @table.query(
      key_condition_expression: "#yr = :year",
      expression_attribute_names: {"#yr" => "year"},
      expression_attribute_values: {":year" => year})
  rescue Aws::DynamoDB::Errors::ServiceError => e
    puts("Couldn't query for movies released in #{year}. Here's why:")
    puts("\t#{e.code}: #{e.message}")
  end
end
```

```

    raise
  else
    response.items
  end

```

- 如需 API 的詳細資訊，請參閱《AWS SDK for Ruby API 參考》中的 [Query](#)。

執行 PartiQL 陳述式

下列程式碼範例示範如何在 DynamoDB 資料表上執行 PartiQL 陳述式。

適用於 Ruby 的開發套件

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在 [AWS 設定和執行程式碼範例儲存庫](#)。

使用 PartiQL 選取單一項目。

```

class DynamoDBPartiQLSingle

  attr_reader :dynamo_resource
  attr_reader :table

  def initialize(table_name)
    client = Aws::DynamoDB::Client.new(region: "us-east-1")
    @dynamodb = Aws::DynamoDB::Resource.new(client: client)
    @table = @dynamodb.table(table_name)
  end

  # Gets a single record from a table using PartiQL.
  # Note: To perform more fine-grained selects,
  # use the Client.query instance method instead.
  #
  # @param title [String] The title of the movie to search.
  # @return [Aws::DynamoDB::Types::ExecuteStatementOutput]
  def select_item_by_title(title)
    request = {
      statement: "SELECT * FROM \"#{@table.name}\" WHERE title=?",
      parameters: [title]
    }
  end
end

```



```

    }
    @dynamodb.client.execute_statement(request)
  end

```

使用 PartiQL 更新單一項目。

```

class DynamoDBPartiQLSingle

  attr_reader :dynamo_resource
  attr_reader :table

  def initialize(table_name)
    client = Aws::DynamoDB::Client.new(region: "us-east-1")
    @dynamodb = Aws::DynamoDB::Resource.new(client: client)
    @table = @dynamodb.table(table_name)
  end

  # Updates a single record from a table using PartiQL.
  #
  # @param title [String] The title of the movie to update.
  # @param year [Integer] The year the movie was released.
  # @param rating [Float] The new rating to assign the title.
  # @return [Aws::DynamoDB::Types::ExecuteStatementOutput]
  def update_rating_by_title(title, year, rating)
    request = {
      statement: "UPDATE \"#{@table.name}\" SET info.rating=? WHERE title=? and
year=?",
      parameters: [{ "N": rating }, title, year]
    }
    @dynamodb.client.execute_statement(request)
  end
end

```

使用 PartiQL 新增單一項目。

```

class DynamoDBPartiQLSingle

  attr_reader :dynamo_resource
  attr_reader :table

  def initialize(table_name)
    client = Aws::DynamoDB::Client.new(region: "us-east-1")

```

```

    @dynamodb = Aws::DynamoDB::Resource.new(client: client)
    @table = @dynamodb.table(table_name)
  end

  # Adds a single record to a table using PartiQL.
  #
  # @param title [String] The title of the movie to update.
  # @param year [Integer] The year the movie was released.
  # @param plot [String] The plot of the movie.
  # @param rating [Float] The new rating to assign the title.
  # @return [Aws::DynamoDB::Types::ExecuteStatementOutput]
  def insert_item(title, year, plot, rating)
    request = {
      statement: "INSERT INTO \"#{@table.name}\" VALUE {'title': ?, 'year': ?,
'info': ?}",
      parameters: [title, year, {'plot': plot, 'rating': rating}]
    }
    @dynamodb.client.execute_statement(request)
  end

```

使用 PartiQL 刪除單一項目。

```

class DynamoDBPartiQLSingle

  attr_reader :dynamo_resource
  attr_reader :table

  def initialize(table_name)
    client = Aws::DynamoDB::Client.new(region: "us-east-1")
    @dynamodb = Aws::DynamoDB::Resource.new(client: client)
    @table = @dynamodb.table(table_name)
  end

  # Deletes a single record from a table using PartiQL.
  #
  # @param title [String] The title of the movie to update.
  # @param year [Integer] The year the movie was released.
  # @return [Aws::DynamoDB::Types::ExecuteStatementOutput]
  def delete_item_by_title(title, year)
    request = {
      statement: "DELETE FROM \"#{@table.name}\" WHERE title=? and year=?",
      parameters: [title, year]
    }

```

```

    }
    @dynamodb.client.execute_statement(request)
  end

```

- 如需 API 詳細資訊，請參閱 AWS SDK for Ruby API 參考 [ExecuteStatement](#) 中的。

執行多批 PartiQL 陳述式

下列程式碼範例示範如何在 DynamoDB 資料表上執行批次的 PartiQL 陳述式。

適用於 Ruby 的開發套件

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在 [AWS 設定和執行程式碼範例儲存庫](#)。

使用 PartiQL 讀取一批項目。

```

class DynamoDBPartiQLBatch

  attr_reader :dynamo_resource
  attr_reader :table

  def initialize(table_name)
    client = Aws::DynamoDB::Client.new(region: "us-east-1")
    @dynamodb = Aws::DynamoDB::Resource.new(client: client)
    @table = @dynamodb.table(table_name)
  end

  # Selects a batch of items from a table using PartiQL
  #
  # @param batch_titles [Array] Collection of movie titles
  # @return [Aws::DynamoDB::Types::BatchExecuteStatementOutput]
  def batch_execute_select(batch_titles)
    request_items = batch_titles.map do |title, year|
      {
        statement: "SELECT * FROM \"#{@table.name}\" WHERE title=? and year=?",
        parameters: [title, year]
      }
    end
  end
end

```

```
@dynamodb.client.batch_execute_statement({statements: request_items})
end
```

使用 PartiQL 刪除一批項目。

```
class DynamoDBPartiQLBatch

  attr_reader :dynamo_resource
  attr_reader :table

  def initialize(table_name)
    client = Aws::DynamoDB::Client.new(region: "us-east-1")
    @dynamodb = Aws::DynamoDB::Resource.new(client: client)
    @table = @dynamodb.table(table_name)
  end

  # Deletes a batch of items from a table using PartiQL
  #
  # @param batch_titles [Array] Collection of movie titles
  # @return [Aws::DynamoDB::Types::BatchExecuteStatementOutput]
  def batch_execute_write(batch_titles)
    request_items = batch_titles.map do |title, year|
      {
        statement: "DELETE FROM \"#{@table.name}\" WHERE title=? and year=?",
        parameters: [title, year]
      }
    end
    @dynamodb.client.batch_execute_statement({statements: request_items})
  end
end
```

- 如需 API 詳細資訊，請參閱 AWS SDK for Ruby API 參考[BatchExecuteStatement](#)中的。

掃描資料表

下列程式碼範例示範如何掃描 DynamoDB 資料表。

適用於 Ruby 的開發套件

 Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫](#)。

```
class DynamoDBBasics
  attr_reader :dynamo_resource
  attr_reader :table

  def initialize(table_name)
    client = Aws::DynamoDB::Client.new(region: "us-east-1")
    @dynamo_resource = Aws::DynamoDB::Resource.new(client: client)
    @table = @dynamo_resource.table(table_name)
  end

  # Scans for movies that were released in a range of years.
  # Uses a projection expression to return a subset of data for each movie.
  #
  # @param year_range [Hash] The range of years to retrieve.
  # @return [Array] The list of movies released in the specified years.
  def scan_items(year_range)
    movies = []
    scan_hash = {
      filter_expression: "#yr between :start_yr and :end_yr",
      projection_expression: "#yr, title, info.rating",
      expression_attribute_names: {"#yr" => "year"},
      expression_attribute_values: {
        ":start_yr" => year_range[:start], ":end_yr" => year_range[:end]}
    }
    done = false
    start_key = nil
    until done
      scan_hash[:exclusive_start_key] = start_key unless start_key.nil?
      response = @table.scan(scan_hash)
      movies.concat(response.items) unless response.items.empty?
      start_key = response.last_evaluated_key
      done = start_key.nil?
    end
  end

  rescue Aws::DynamoDB::Errors::ServiceError => e
    puts("Couldn't scan for movies. Here's why:")
    puts("\t#{e.code}: #{e.message}")
  end
end
```

```
    raise
  else
    movies
  end
```

- 如需 API 的詳細資訊，請參閱 [《AWS SDK for Ruby API 參考》](#) 中的 Scan。

更新資料表中的項目

下列程式碼範例示範如何更新 DynamoDB 資料表中的項目。

適用於 Ruby 的開發套件

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在 [AWS 設定和執行程式碼範例儲存庫](#)。

```
class DynamoDBBasics
  attr_reader :dynamo_resource
  attr_reader :table

  def initialize(table_name)
    client = Aws::DynamoDB::Client.new(region: "us-east-1")
    @dynamo_resource = Aws::DynamoDB::Resource.new(client: client)
    @table = @dynamo_resource.table(table_name)
  end

  # Updates rating and plot data for a movie in the table.
  #
  # @param movie [Hash] The title, year, plot, rating of the movie.
  def update_item(movie)

    response = @table.update_item(
      key: {"year" => movie[:year], "title" => movie[:title]},
      update_expression: "set info.rating=:r",
      expression_attribute_values: { ":r" => movie[:rating] },
      return_values: "UPDATED_NEW")
  rescue Aws::DynamoDB::Errors::ServiceError => e
    puts("Couldn't update movie #{movie[:title]} (#{movie[:year]}) in table
    #{@table.name}\n")
  end
end
```

```
puts("\t#{e.code}: #{e.message}")
raise
else
  response.attributes
end
```

- 如需 API 詳細資訊，請參閱 AWS SDK for Ruby API 參考 [UpdateItem](#) 中的。

批次寫入項目

下列程式碼範例示範如何分批寫入 DynamoDB 項目。

適用於 Ruby 的開發套件

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在 [AWS 設定和執行程式碼範例儲存庫](#)。

```
class DynamoDBBasics
  attr_reader :dynamo_resource
  attr_reader :table

  def initialize(table_name)
    client = Aws::DynamoDB::Client.new(region: "us-east-1")
    @dynamo_resource = Aws::DynamoDB::Resource.new(client: client)
    @table = @dynamo_resource.table(table_name)
  end

  # Fills an Amazon DynamoDB table with the specified data. Items are sent in
  # batches of 25 until all items are written.
  #
  # @param movies [Enumerable] The data to put in the table. Each item must contain
  # at least
  #           the keys required by the schema that was specified
  # when the
  #           table was created.
  def write_batch(movies)
    index = 0
    slice_size = 25
    while index < movies.length
```

```
movie_items = []
movies[index, slice_size].each do |movie|
  movie_items.append({put_request: { item: movie }})
end
@dynamo_resource.client.batch_write_item({request_items: { @table.name =>
movie_items }})
  index += slice_size
end
rescue Aws::DynamoDB::Errors::ServiceError => e
  puts(
    "Couldn't load data into table #{@table.name}. Here's why:")
  puts("\t#{e.code}: #{e.message}")
  raise
end
```

- 如需 API 詳細資訊，請參閱 AWS SDK for Ruby API 參考[BatchWriteItem](#)中的。

案例

開始使用資料表、項目和查詢

以下程式碼範例顯示做法：

- 建立可存放電影資料的資料表。
- 放入、取得和更新資料表中的單個電影。
- 將影片資料從範例 JSON 檔案寫入資料表。
- 查詢特定年份發表的電影。
- 掃描某個年份範圍內發表的電影。
- 從資料表刪除電影，然後刪除資料表。

適用於 Ruby 的開發套件

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫](#)。

建立封裝 DynamoDB 資料表的類別。


```

# Creates an Amazon DynamoDB table that can be used to store movie data.
# The table uses the release year of the movie as the partition key and the
# title as the sort key.
#
# @param table_name [String] The name of the table to create.
# @return [Aws::DynamoDB::Table] The newly created table.
def create_table(table_name)
  @table = @dynamo_resource.create_table(
    table_name: table_name,
    key_schema: [
      {attribute_name: "year", key_type: "HASH"}, # Partition key
      {attribute_name: "title", key_type: "RANGE"} # Sort key
    ],
    attribute_definitions: [
      {attribute_name: "year", attribute_type: "N"},
      {attribute_name: "title", attribute_type: "S"}
    ],
    provisioned_throughput: {read_capacity_units: 10, write_capacity_units: 10})
  @dynamo_resource.client.wait_until(:table_exists, table_name: table_name)
  @table
rescue Aws::DynamoDB::Errors::ServiceError => e
  @logger.error("Failed create table #{table_name}:\n#{e.code}: #{e.message}")
  raise
end

```

建立 Helper 函數以下載並擷取範例 JSON 檔案。

```

# Gets sample movie data, either from a local file or by first downloading it from
# the Amazon DynamoDB Developer Guide.
#
# @param movie_file_name [String] The local file name where the movie data is
# stored in JSON format.
# @return [Hash] The movie data as a Hash.
def fetch_movie_data(movie_file_name)
  if !File.file?(movie_file_name)
    @logger.debug("Downloading #{movie_file_name}...")
    movie_content = URI.open(
      "https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/samples/
moviedata.zip"
    )
    movie_json = ""
    Zip::File.open_buffer(movie_content) do |zip|

```

```

        zip.each do |entry|
          movie_json = entry.get_input_stream.read
        end
      end
    else
      movie_json = File.read(movie_file_name)
    end
    movie_data = JSON.parse(movie_json)
    # The sample file lists over 4000 movies. This returns only the first 250.
    movie_data.slice(0, 250)
  rescue StandardError => e
    puts("Failure downloading movie data:\n#{e}")
    raise
  end
end

```

執行互動式案例以建立資料表並對其執行動作。

```

table_name = "doc-example-table-movies-#{rand(10**4)}"
scaffold = Scaffold.new(table_name)
dynamodb_wrapper = DynamoDBBasics.new(table_name)

new_step(1, "Create a new DynamoDB table if none already exists.")
unless scaffold.exists?(table_name)
  puts("\nNo such table: #{table_name}. Creating it...")
  scaffold.create_table(table_name)
  print "Done!\n".green
end

new_step(2, "Add a new record to the DynamoDB table.")
my_movie = {}
my_movie[:title] = CLI::UI::Prompt.ask("Enter the title of a movie to add to the
table. E.g. The Matrix")
my_movie[:year] = CLI::UI::Prompt.ask("What year was it released? E.g. 1989").to_i
my_movie[:rating] = CLI::UI::Prompt.ask("On a scale of 1 - 10, how do you rate it?
E.g. 7").to_i
my_movie[:plot] = CLI::UI::Prompt.ask("Enter a brief summary of the plot. E.g. A
man awakens to a new reality.")
dynamodb_wrapper.add_item(my_movie)
puts("\nNew record added:")
puts JSON.pretty_generate(my_movie).green
print "Done!\n".green

```

```
new_step(3, "Update a record in the DynamoDB table.")
my_movie[:rating] = CLI::UI::Prompt.ask("Let's update the movie you added with a
new rating, e.g. 3:").to_i
response = dynamodb_wrapper.update_item(my_movie)
puts("Updated '#{my_movie[:title]}' with new attributes:")
puts JSON.pretty_generate(response).green
print "Done!\n".green

new_step(4, "Get a record from the DynamoDB table.")
puts("Searching for #{my_movie[:title]} (#{my_movie[:year]})...")
response = dynamodb_wrapper.get_item(my_movie[:title], my_movie[:year])
puts JSON.pretty_generate(response).green
print "Done!\n".green

new_step(5, "Write a batch of items into the DynamoDB table.")
download_file = "moviedata.json"
puts("Downloading movie database to #{download_file}...")
movie_data = scaffold.fetch_movie_data(download_file)
puts("Writing movie data from #{download_file} into your table...")
scaffold.write_batch(movie_data)
puts("Records added: #{movie_data.length}.")
print "Done!\n".green

new_step(5, "Query for a batch of items by key.")
loop do
  release_year = CLI::UI::Prompt.ask("Enter a year between 1972 and 2018, e.g.
1999:").to_i
  results = dynamodb_wrapper.query_items(release_year)
  if results.any?
    puts("There were #{results.length} movies released in #{release_year}:")
    results.each do |movie|
      print "\t #{movie["title"]}".green
    end
    break
  else
    continue = CLI::UI::Prompt.ask("Found no movies released in #{release_year}!
Try another year? (y/n)")
    break if !continue.eql?("y")
  end
end
print "\nDone!\n".green

new_step(6, "Scan for a batch of items using a filter expression.")
years = {}
```

```

years[:start] = CLI::UI::Prompt.ask("Enter a starting year between 1972 and
2018:")
years[:end] = CLI::UI::Prompt.ask("Enter an ending year between 1972 and 2018:")
releases = dynamodb_wrapper.scan_items(years)
if !releases.empty?
  puts("Found #{releases.length} movies.")
  count = Question.ask(
    "How many do you want to see? ", method(:is_int), in_range(1,
releases.length))
  puts("Here are your #{count} movies:")
  releases.take(count).each do |release|
    puts("\t#{release["title"]}")
  end
else
  puts("I don't know about any movies released between #{years[:start]} "\
    "and #{years[:end]}".)
end
print "\nDone!\n".green

new_step(7, "Delete an item from the DynamoDB table.")
answer = CLI::UI::Prompt.ask("Do you want to remove '#{my_movie[:title]}'? (y/n)
")
if answer.eql?("y")
  dynamodb_wrapper.delete_item(my_movie[:title], my_movie[:year])
  puts("Removed '#{my_movie[:title]}' from the table.")
  print "\nDone!\n".green
end

new_step(8, "Delete the DynamoDB table.")
answer = CLI::UI::Prompt.ask("Delete the table? (y/n)")
if answer.eql?("y")
  scaffold.delete_table
  puts("Deleted #{table_name}.")
else
  puts("Don't forget to delete the table when you're done!")
end
print "\nThanks for watching!\n".green
rescue Aws::Errors::ServiceError
  puts("Something went wrong with the demo.")
rescue Errno::ENOENT
  true
end

```


- 如需 API 詳細資訊，請參閱《AWS SDK for Ruby API 參考》中的下列主題。
 - [BatchWriteItem](#)
 - [CreateTable](#)
 - [DeleteItem](#)
 - [DeleteTable](#)
 - [DescribeTable](#)
 - [GetItem](#)
 - [PutItem](#)
 - [查詢](#)
 - [掃描](#)
 - [UpdateItem](#)

使用多批 PartiQL 陳述式查詢資料表

以下程式碼範例顯示做法：

- 透過執行多個 SELECT 陳述式取得一批項目。
- 透過執行多個 INSERT 陳述式新增一批項目。
- 透過執行多個 UPDATE 陳述式更新一批項目。
- 透過執行多個 DELETE 陳述式刪除一批項目。

適用於 Ruby 的開發套件

 Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫](#)。

執行一個情境，該情境會建立資料表並執行批次 PartiQL 查詢。

```
table_name = "doc-example-table-movies-partiql-#{rand(10**4)}"
scaffold = Scaffold.new(table_name)
sdk = DynamoDBPartiQLBatch.new(table_name)

new_step(1, "Create a new DynamoDB table if none already exists.")
unless scaffold.exists?(table_name)
```

```
puts("\nNo such table: #{table_name}. Creating it...")
  scaffold.create_table(table_name)
  print "Done!\n".green
end

new_step(2, "Populate DynamoDB table with movie data.")
download_file = "moviedata.json"
puts("Downloading movie database to #{download_file}...")
movie_data = scaffold.fetch_movie_data(download_file)
puts("Writing movie data from #{download_file} into your table...")
scaffold.write_batch(movie_data)
puts("Records added: #{movie_data.length}.")
print "Done!\n".green

new_step(3, "Select a batch of items from the movies table.")
puts "Let's select some popular movies for side-by-side comparison."
response = sdk.batch_execute_select([["Mean Girls", 2004], ["Goodfellas", 1977],
["The Prancing of the Lambs", 2005]])
puts("Items selected: #{response['responses'].length}\n")
print "\nDone!\n".green

new_step(4, "Delete a batch of items from the movies table.")
sdk.batch_execute_write([["Mean Girls", 2004], ["Goodfellas", 1977], ["The
Prancing of the Lambs", 2005]])
print "\nDone!\n".green

new_step(5, "Delete the table.")
if scaffold.exists?(table_name)
  scaffold.delete_table
end
end
```

- 如需 API 詳細資訊，請參閱 AWS SDK for Ruby API 參考[BatchExecuteStatement](#)中的。

使用 PartiQL 查詢資料表

以下程式碼範例顯示做法：

- 透過執行 SELECT 陳述式取得項目。
- 透過執行 INSERT 陳述式新增項目。
- 透過執行 UPDATE 陳述式更新項目。

- 透過執行 DELETE 陳述式刪除項目。

適用於 Ruby 的開發套件

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在 [AWS 設定和執行程式碼範例儲存庫](#)。

執行一個情境，該情境建立資料表並執行 PartiQL 查詢。

```
table_name = "doc-example-table-movies-partiql-#{rand(10**8)}"
scaffold = Scaffold.new(table_name)
sdk = DynamoDBPartiQLSingle.new(table_name)

new_step(1, "Create a new DynamoDB table if none already exists.")
unless scaffold.exists?(table_name)
  puts("\nNo such table: #{table_name}. Creating it...")
  scaffold.create_table(table_name)
  print "Done!\n".green
end

new_step(2, "Populate DynamoDB table with movie data.")
download_file = "moviedata.json"
puts("Downloading movie database to #{download_file}...")
movie_data = scaffold.fetch_movie_data(download_file)
puts("Writing movie data from #{download_file} into your table...")
scaffold.write_batch(movie_data)
puts("Records added: #{movie_data.length}.")
print "Done!\n".green

new_step(3, "Select a single item from the movies table.")
response = sdk.select_item_by_title("Star Wars")
puts("Items selected for title 'Star Wars': #{response.items.length}\n")
print "#{response.items.first}".yellow
print "\n\nDone!\n".green

new_step(4, "Update a single item from the movies table.")
puts "Let's correct the rating on The Big Lebowski to 10.0."
sdk.update_rating_by_title("The Big Lebowski", 1998, 10.0)
print "\n\nDone!\n".green
```

```
new_step(5, "Delete a single item from the movies table.")
puts "Let's delete The Silence of the Lambs because it's just too scary."
sdk.delete_item_by_title("The Silence of the Lambs", 1991)
print "\nDone!\n".green

new_step(6, "Insert a new item into the movies table.")
puts "Let's create a less-scary movie called The Prancing of the Lambs."
sdk.insert_item("The Prancing of the Lambs", 2005, "A movie about happy
livestock.", 5.0)
print "\nDone!\n".green

new_step(7, "Delete the table.")
if scaffold.exists?(table_name)
  scaffold.delete_table
end
end
```

- 如需 API 詳細資訊，請參閱 AWS SDK for Ruby API 參考[ExecuteStatement](#)中的。

Amazon EC2 示例使用 SDK for Ruby

下列程式碼範例說明如何透過 AWS SDK for Ruby 搭配 Amazon EC2 使用來執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境和跨服務範例中查看內容中的動作。

案例是向您展示如何呼叫相同服務中的多個函數來完成特定任務的程式碼範例。

每個範例都包含一個連結 GitHub，您可以在其中找到如何在內容中設定和執程式碼的指示。

主題


- [動作](#)

動作

配置彈性 IP 地址

下列程式碼範例顯示如何為 Amazon EC2 配置彈性 IP 位址。

適用於 Ruby 的開發套件

 Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫](#)。


```
# Creates an Elastic IP address in Amazon Virtual Private Cloud (Amazon VPC).
#
# @param ec2_client [Aws::EC2::Client] An initialized EC2 client.
# @return [String] The allocation ID corresponding to the Elastic IP address.
# @example
#   puts allocate_elastic_ip_address(Aws::EC2::Client.new(region: 'us-west-2'))
def allocate_elastic_ip_address(ec2_client)
  response = ec2_client.allocate_address(domain: "vpc")
  return response.allocation_id
rescue StandardError => e
  puts "Error allocating Elastic IP address: #{e.message}"
  return "Error"
end
```

- 如需 API 詳細資訊，請參閱 AWS SDK for Ruby API 參考[AllocateAddress](#)中的。

將彈性 IP 地址與執行個體建立關聯

下列程式碼範例顯示如何將彈性 IP 地址與 Amazon EC2 執行個體建立關聯。

適用於 Ruby 的開發套件

 Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫](#)。

```
# Associates an Elastic IP address with an Amazon Elastic Compute Cloud
# (Amazon EC2) instance.
#
# Prerequisites:
#
```

```

# - The allocation ID corresponding to the Elastic IP address.
# - The Amazon EC2 instance.
#
# @param ec2_client [Aws::EC2::Client] An initialized EC2 client.
# @param allocation_id [String] The ID of the allocation corresponding to
#   the Elastic IP address.
# @param instance_id [String] The ID of the instance.
# @return [String] The association ID corresponding to the association of the
#   Elastic IP address to the instance.
# @example
#   puts allocate_elastic_ip_address(
#     Aws::EC2::Client.new(region: 'us-west-2'),
#     'eipalloc-04452e528a66279EX',
#     'i-033c48ef067af3dEX')
def associate_elastic_ip_address_with_instance(
  ec2_client,
  allocation_id,
  instance_id
)
  response = ec2_client.associate_address(
    allocation_id: allocation_id,
    instance_id: instance_id,
  )
  return response.association_id
rescue StandardError => e
  puts "Error associating Elastic IP address with instance: #{e.message}"
  return "Error"
end

```

- 如需 API 詳細資訊，請參閱 AWS SDK for Ruby API 參考 [AssociateAddress](#) 中的。

建立 Amazon Virtual Private Cloud (Amazon VPC)

下面的程式碼範例演示如何建立 Amazon Virtual Private Cloud (Amazon VPC)。

適用於 Ruby 的開發套件

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在 [AWS 設定和執行程式碼範例儲存庫](#)。

```
require "aws-sdk-ec2"

# Creates a virtual private cloud (VPC) in
# Amazon Virtual Private Cloud (Amazon VPC) and then tags
# the VPC.
#
# @param ec2_resource [Aws::EC2::Resource] An initialized
#   Amazon Elastic Compute Cloud (Amazon EC2) resource object.
# @param cidr_block [String] The IPv4 CIDR block for the subnet.
# @param tag_key [String] The key portion of the tag for the VPC.
# @param tag_value [String] The value portion of the tag for the VPC.
# @return [Boolean] true if the VPC was created and tagged;
#   otherwise, false.
# @example
#   exit 1 unless vpc_created_and_tagged?(
#     Aws::EC2::Resource.new(region: 'us-west-2'),
#     '10.0.0.0/24',
#     'my-key',
#     'my-value'
#   )
def vpc_created_and_tagged?(
  ec2_resource,
  cidr_block,
  tag_key,
  tag_value
)
  vpc = ec2_resource.create_vpc(cidr_block: cidr_block)

  # Create a public DNS by enabling DNS support and DNS hostnames.
  vpc.modify_attribute(enable_dns_support: { value: true })
  vpc.modify_attribute(enable_dns_hostnames: { value: true })

  vpc.create_tags(tags: [{ key: tag_key, value: tag_value }])

  puts "Created VPC with ID '#{vpc.id}' and tagged with key " \
    "'#{tag_key}' and value '#{tag_value}'."
  return true
rescue StandardError => e
  puts "#{e.message}"
  return false
end
```

```
# Example usage:
def run_me
  cidr_block = ""
  tag_key = ""
  tag_value = ""
  region = ""
  # Print usage information and then stop.
  if ARGV[0] == "--help" || ARGV[0] == "-h"
    puts "Usage:  ruby ec2-ruby-example-create-vpc.rb " \
         "CIDR_BLOCK TAG_KEY TAG_VALUE REGION"
    # Replace us-west-2 with the AWS Region you're using for Amazon EC2.
    puts "Example: ruby ec2-ruby-example-create-vpc.rb " \
         "10.0.0.0/24 my-key my-value us-west-2"
    exit 1
  # If no values are specified at the command prompt, use these default values.
  elsif ARGV.count.zero?
    cidr_block = "10.0.0.0/24"
    tag_key = "my-key"
    tag_value = "my-value"
    # Replace us-west-2 with the AWS Region you're using for Amazon EC2.
    region = "us-west-2"
  # Otherwise, use the values as specified at the command prompt.
  else
    cidr_block = ARGV[0]
    tag_key = ARGV[1]
    tag_value = ARGV[2]
    region = ARGV[3]
  end

  ec2_resource = Aws::EC2::Resource.new(region: region)

  if vpc_created_and_tagged?(
    ec2_resource,
    cidr_block,
    tag_key,
    tag_value
  )
    puts "VPC created and tagged."
  else
    puts "VPC not created or not tagged."
  end
end

run_me if $PROGRAM_NAME == __FILE__
```

- 如需 API 詳細資訊，請參閱 AWS SDK for Ruby API 參考 [CreateVpc](#) 中的。

建立路由表

下列程式碼範例顯示如何建立路由表並與 Amazon EC2 子網路產生關聯。

適用於 Ruby 的開發套件

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在 [AWS 設定和執行程式碼範例儲存庫](#)。

```
require "aws-sdk-ec2"

# Prerequisites:
#
# - A VPC in Amazon VPC.
# - A subnet in that VPC.
# - A gateway attached to that subnet.
#
# @param ec2_resource [Aws::EC2::Resource] An initialized
#   Amazon Elastic Compute Cloud (Amazon EC2) resource object.
# @param vpc_id [String] The ID of the VPC for the route table.
# @param subnet_id [String] The ID of the subnet for the route table.
# @param gateway_id [String] The ID of the gateway for the route.
# @param destination_cidr_block [String] The destination CIDR block
#   for the route.
# @param tag_key [String] The key portion of the tag for the route table.
# @param tag_value [String] The value portion of the tag for the route table.
# @return [Boolean] true if the route table was created and associated;
#   otherwise, false.
# @example
#   exit 1 unless route_table_created_and_associated?(
#     Aws::EC2::Resource.new(region: 'us-west-2'),
#     'vpc-0b6f769731EXAMPLE',
#     'subnet-03d9303b57EXAMPLE',
#     'igw-06ca90c011EXAMPLE',
#     '0.0.0.0/0',
#     'my-key',
```

```
#   'my-value'
#   )
def route_table_created_and_associated?(
  ec2_resource,
  vpc_id,
  subnet_id,
  gateway_id,
  destination_cidr_block,
  tag_key,
  tag_value
)
  route_table = ec2_resource.create_route_table(vpc_id: vpc_id)
  puts "Created route table with ID '#{route_table.id}'."
  route_table.create_tags(
    tags: [
      {
        key: tag_key,
        value: tag_value
      }
    ]
  )
  puts "Added tags to route table."
  route_table.create_route(
    destination_cidr_block: destination_cidr_block,
    gateway_id: gateway_id
  )
  puts "Created route with destination CIDR block " \
    "'#{destination_cidr_block}' and associated with gateway " \
    "with ID '#{gateway_id}'."
  route_table.associate_with_subnet(subnet_id: subnet_id)
  puts "Associated route table with subnet with ID '#{subnet_id}'."
  return true
rescue StandardError => e
  puts "Error creating or associating route table: #{e.message}"
  puts "If the route table was created but not associated, you should " \
    "clean up by deleting the route table."
  return false
end

# Example usage:
def run_me
  vpc_id = ""
  subnet_id = ""
  gateway_id = ""
```

```
destination_cidr_block = ""
tag_key = ""
tag_value = ""
region = ""
# Print usage information and then stop.
if ARGV[0] == "--help" || ARGV[0] == "-h"
  puts "Usage: ruby ec2-ruby-example-create-route-table.rb " \
    "VPC_ID SUBNET_ID GATEWAY_ID DESTINATION_CIDR_BLOCK " \
    "TAG_KEY TAG_VALUE REGION"
# Replace us-west-2 with the AWS Region you're using for Amazon EC2.
  puts "Example: ruby ec2-ruby-example-create-route-table.rb " \
    "vpc-0b6f769731EXAMPLE subnet-03d9303b57EXAMPLE igw-06ca90c011EXAMPLE " \
    "'0.0.0.0/0' my-key my-value us-west-2"
  exit 1
# If no values are specified at the command prompt, use these default values.
elsif ARGV.count.zero?
  vpc_id = "vpc-0b6f769731EXAMPLE"
  subnet_id = "subnet-03d9303b57EXAMPLE"
  gateway_id = "igw-06ca90c011EXAMPLE"
  destination_cidr_block = "0.0.0.0/0"
  tag_key = "my-key"
  tag_value = "my-value"
  # Replace us-west-2 with the AWS Region you're using for Amazon EC2.
  region = "us-west-2"
# Otherwise, use the values as specified at the command prompt.
else
  vpc_id = ARGV[0]
  subnet_id = ARGV[1]
  gateway_id = ARGV[2]
  destination_cidr_block = ARGV[3]
  tag_key = ARGV[4]
  tag_value = ARGV[5]
  region = ARGV[6]
end

ec2_resource = Aws::EC2::Resource.new(region: region)

if route_table_created_and_associated?(
  ec2_resource,
  vpc_id,
  subnet_id,
  gateway_id,
  destination_cidr_block,
  tag_key,
```

```
    tag_value
  )
  puts "Route table created and associated."
else
  puts "Route table not created or not associated."
end
end
end

run_me if $PROGRAM_NAME == __FILE__
```

- 如需 API 詳細資訊，請參閱 AWS SDK for Ruby API 參考[CreateRouteTable](#)中的。

建立安全群組

下列程式碼範例示範如何建立 Amazon EC2 安全群組。

適用於 Ruby 的開發套件

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫](#)。

```
# This code example does the following:
# 1. Creates an Amazon Elastic Compute Cloud (Amazon EC2) security group.
# 2. Adds inbound rules to the security group.
# 3. Displays information about available security groups.
# 4. Deletes the security group.

require "aws-sdk-ec2"

# Creates an Amazon Elastic Compute Cloud (Amazon EC2) security group.
#
# Prerequisites:
#
# - A VPC in Amazon Virtual Private Cloud (Amazon VPC).
#
# @param ec2_client [Aws::EC2::Client] An initialized
#   Amazon EC2 client.
```



```

# @param group_name [String] A name for the security group.
# @param description [String] A description for the security group.
# @param vpc_id [String] The ID of the VPC for the security group.
# @return [String] The ID of security group that was created.
# @example
#   puts create_security_group(
#     Aws::EC2::Client.new(region: 'us-west-2'),
#     'my-security-group',
#     'This is my security group.',
#     'vpc-6713dfEX'
#   )
def create_security_group(
  ec2_client,
  group_name,
  description,
  vpc_id
)
  security_group = ec2_client.create_security_group(
    group_name: group_name,
    description: description,
    vpc_id: vpc_id
  )
  puts "Created security group '#{group_name}' with ID " \
    "'#{security_group.group_id}' in VPC with ID '#{vpc_id}'."
  return security_group.group_id
rescue StandardError => e
  puts "Error creating security group: #{e.message}"
  return "Error"
end

# Adds an inbound rule to an Amazon Elastic Compute Cloud (Amazon EC2)
# security group.
#
# Prerequisites:
#
# - The security group.
#
# @param ec2_client [Aws::EC2::Client] An initialized Amazon EC2 client.
# @param security_group_id [String] The ID of the security group.
# @param ip_protocol [String] The network protocol for the inbound rule.
# @param from_port [String] The originating port for the inbound rule.
# @param to_port [String] The destination port for the inbound rule.
# @param cidr_ip_range [String] The CIDR IP range for the inbound rule.
# @return

```

```
# @example
#   exit 1 unless security_group_ingress_authorized?(
#     Aws::EC2::Client.new(region: 'us-west-2'),
#     'sg-030a858e078f1b9EX',
#     'tcp',
#     '80',
#     '80',
#     '0.0.0.0/0'
#   )
def security_group_ingress_authorized?(
  ec2_client,
  security_group_id,
  ip_protocol,
  from_port,
  to_port,
  cidr_ip_range
)
  ec2_client.authorize_security_group_ingress(
    group_id: security_group_id,
    ip_permissions: [
      {
        ip_protocol: ip_protocol,
        from_port: from_port,
        to_port: to_port,
        ip_ranges: [
          {
            cidr_ip: cidr_ip_range
          }
        ]
      }
    ]
  )
  puts "Added inbound rule to security group '#{security_group_id}' for protocol " \
    "'#{ip_protocol}' from port '#{from_port}' to port '#{to_port}' " \
    "with CIDR IP range '#{cidr_ip_range}'."
  return true
rescue StandardError => e
  puts "Error adding inbound rule to security group: #{e.message}"
  return false
end

# Displays information about a security group's IP permissions set in
# Amazon Elastic Compute Cloud (Amazon EC2).
#
```

```
# Prerequisites:
#
# - A security group with inbound rules, outbound rules, or both.
#
# @param p [Aws::EC2::Types::IpPermission] The IP permissions set.
# @example
#   ec2_client = Aws::EC2::Client.new(region: 'us-west-2')
#   response = ec2_client.describe_security_groups
#   unless sg.ip_permissions.empty?
#     describe_security_group_permissions(
#       response.security_groups[0].ip_permissions[0]
#     )
#   end
def describe_security_group_permissions(perm)
  print " Protocol: #{perm.ip_protocol == '-1' ? 'All' : perm.ip_protocol}"

  unless perm.from_port.nil?
    if perm.from_port == "-1" || perm.from_port == -1
      print ", From: All"
    else
      print ", From: #{perm.from_port}"
    end
  end

  unless perm.to_port.nil?
    if perm.to_port == "-1" || perm.to_port == -1
      print ", To: All"
    else
      print ", To: #{perm.to_port}"
    end
  end

  if perm.key?(:ipv_6_ranges) && perm.ipv_6_ranges.count.positive?
    print ", CIDR IPv6: #{perm.ipv_6_ranges[0].cidr_ipv_6}"
  end

  if perm.key?(:ip_ranges) && perm.ip_ranges.count.positive?
    print ", CIDR IPv4: #{perm.ip_ranges[0].cidr_ip}"
  end

  print "\n"
end

# Displays information about available security groups in
```

```
# Amazon Elastic Compute Cloud (Amazon EC2).
#
# @param ec2_client [Aws::EC2::Client] An initialized Amazon EC2 client.
# @example
#   describe_security_groups(Aws::EC2::Client.new(region: 'us-west-2'))
def describe_security_groups(ec2_client)
  response = ec2_client.describe_security_groups

  if response.security_groups.count.positive?
    response.security_groups.each do |sg|
      puts "-" * (sg.group_name.length + 13)
      puts "Name:      #{sg.group_name}"
      puts "Description: #{sg.description}"
      puts "Group ID:   #{sg.group_id}"
      puts "Owner ID:   #{sg.owner_id}"
      puts "VPC ID:    #{sg.vpc_id}"

      if sg.tags.count.positive?
        puts "Tags:"
        sg.tags.each do |tag|
          puts "  Key: #{tag.key}, Value: #{tag.value}"
        end
      end

      unless sg.ip_permissions.empty?
        puts "Inbound rules:" if sg.ip_permissions.count.positive?
        sg.ip_permissions.each do |p|
          describe_security_group_permissions(p)
        end
      end

      unless sg.ip_permissions_egress.empty?
        puts "Outbound rules:" if sg.ip_permissions_egress.count.positive?
        sg.ip_permissions_egress.each do |p|
          describe_security_group_permissions(p)
        end
      end
    end
  else
    puts "No security groups found."
  end
rescue StandardError => e
  puts "Error getting information about security groups: #{e.message}"
end
```

```

# Deletes an Amazon Elastic Compute Cloud (Amazon EC2)
# security group.
#
# Prerequisites:
#
# - The security group.
#
# @param ec2_client [Aws::EC2::Client] An initialized
#   Amazon EC2 client.
# @param security_group_id [String] The ID of the security group to delete.
# @return [Boolean] true if the security group was deleted; otherwise, false.
# @example
#   exit 1 unless security_group_deleted?(
#     Aws::EC2::Client.new(region: 'us-west-2'),
#     'sg-030a858e078f1b9EX'
#   )
def security_group_deleted?(ec2_client, security_group_id)
  ec2_client.delete_security_group(group_id: security_group_id)
  puts "Deleted security group '#{security_group_id}'."
  return true
rescue StandardError => e
  puts "Error deleting security group: #{e.message}"
  return false
end

# Example usage:
def run_me
  group_name = ""
  description = ""
  vpc_id = ""
  ip_protocol_http = ""
  from_port_http = ""
  to_port_http = ""
  cidr_ip_range_http = ""
  ip_protocol_ssh = ""
  from_port_ssh = ""
  to_port_ssh = ""
  cidr_ip_range_ssh = ""
  region = ""
  # Print usage information and then stop.
  if ARGV[0] == "--help" || ARGV[0] == "-h"
    puts "Usage:  ruby ec2-ruby-example-security-group.rb " \
      "GROUP_NAME DESCRIPTION VPC_ID IP_PROTOCOL_1 FROM_PORT_1 TO_PORT_1 " \

```

```

    "CIDR_IP_RANGE_1 IP_PROTOCOL_2 FROM_PORT_2 TO_PORT_2 " \
    "CIDR_IP_RANGE_2 REGION"
puts "Example: ruby ec2-ruby-example-security-group.rb " \
    "my-security-group 'This is my security group.' vpc-6713dfEX " \
    "tcp 80 80 '0.0.0.0/0' tcp 22 22 '0.0.0.0/0' us-west-2"
exit 1
# If no values are specified at the command prompt, use these default values.
elsif ARGV.count.zero?
  group_name = "my-security-group"
  description = "This is my security group."
  vpc_id = "vpc-6713dfEX"
  ip_protocol_http = "tcp"
  from_port_http = "80"
  to_port_http = "80"
  cidr_ip_range_http = "0.0.0.0/0"
  ip_protocol_ssh = "tcp"
  from_port_ssh = "22"
  to_port_ssh = "22"
  cidr_ip_range_ssh = "0.0.0.0/0"
  # Replace us-west-2 with the AWS Region you're using for Amazon EC2.
  region = "us-west-2"
# Otherwise, use the values as specified at the command prompt.
else
  group_name = ARGV[0]
  description = ARGV[1]
  vpc_id = ARGV[2]
  ip_protocol_http = ARGV[3]
  from_port_http = ARGV[4]
  to_port_http = ARGV[5]
  cidr_ip_range_http = ARGV[6]
  ip_protocol_ssh = ARGV[7]
  from_port_ssh = ARGV[8]
  to_port_ssh = ARGV[9]
  cidr_ip_range_ssh = ARGV[10]
  region = ARGV[11]
end

security_group_id = ""
security_group_exists = false
ec2_client = Aws::EC2::Client.new(region: region)

puts "Attempting to create security group..."
security_group_id = create_security_group(
  ec2_client,

```

```
    group_name,
    description,
    vpc_id
  )
  if security_group_id == "Error"
    puts "Could not create security group. Skipping this step."
  else
    security_group_exists = true
  end

  if security_group_exists
    puts "Attempting to add inbound rules to security group..."
    unless security_group_ingress_authorized?(
      ec2_client,
      security_group_id,
      ip_protocol_http,
      from_port_http,
      to_port_http,
      cidr_ip_range_http
    )
      puts "Could not add inbound HTTP rule to security group. " \
        "Skipping this step."
    end

    unless security_group_ingress_authorized?(
      ec2_client,
      security_group_id,
      ip_protocol_ssh,
      from_port_ssh,
      to_port_ssh,
      cidr_ip_range_ssh
    )
      puts "Could not add inbound SSH rule to security group. " \
        "Skipping this step."
    end
  end

  puts "\nInformation about available security groups:"
  describe_security_groups(ec2_client)

  if security_group_exists
    puts "\nAttempting to delete security group..."
    unless security_group_deleted?(ec2_client, security_group_id)
      puts "Could not delete security group. You must delete it yourself."
    end
  end
end
```

```

    end
  end
end

run_me if $PROGRAM_NAME == __FILE__

```

- 如需 API 詳細資訊，請參閱 AWS SDK for Ruby API 參考[CreateSecurityGroup](#)中的。

建立安全金鑰對

下列程式碼範例顯示如何為 Amazon EC2 建立安全 key pair。

適用於 Ruby 的開發套件

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫](#)。

```

# This code example does the following:
# 1. Creates a key pair in Amazon Elastic Compute Cloud (Amazon EC2).
# 2. Displays information about available key pairs.
# 3. Deletes the key pair.

require "aws-sdk-ec2"

# @param ec2_client [Aws::EC2::Client] An initialized EC2 client.
# @param key_pair_name [String] The name for the key pair and private
#   key file.
# @return [Boolean] true if the key pair and private key file were
#   created; otherwise, false.
# @example
#   exit 1 unless key_pair_created?(
#     Aws::EC2::Client.new(region: 'us-west-2'),
#     'my-key-pair'
#   )
def key_pair_created?(ec2_client, key_pair_name)
  key_pair = ec2_client.create_key_pair(key_name: key_pair_name)
  puts "Created key pair '#{key_pair.key_name}' with fingerprint " \
    "'#{key_pair.key_fingerprint}' and ID '#{key_pair.key_pair_id}'."
end

```



```

    filename = File.join(Dir.home, key_pair_name + ".pem")
    File.open(filename, "w") { |file| file.write(key_pair.key_material) }
    puts "Private key file saved locally as '#{filename}'."
    return true
rescue Aws::EC2::Errors::InvalidKeyPairDuplicate
    puts "Error creating key pair: a key pair named '#{key_pair_name}' " \
        "already exists."
    return false
rescue StandardError => e
    puts "Error creating key pair or saving private key file: #{e.message}"
    return false
end

# Displays information about available key pairs in
# Amazon Elastic Compute Cloud (Amazon EC2).
#
# @param ec2_client [Aws::EC2::Client] An initialized EC2 client.
# @example
#   describe_key_pairs(Aws::EC2::Client.new(region: 'us-west-2'))
def describe_key_pairs(ec2_client)
    result = ec2_client.describe_key_pairs
    if result.key_pairs.count.zero?
        puts "No key pairs found."
    else
        puts "Key pair names:"
        result.key_pairs.each do |key_pair|
            puts key_pair.key_name
        end
    end
end
rescue StandardError => e
    puts "Error getting information about key pairs: #{e.message}"
end

# Deletes a key pair in Amazon Elastic Compute Cloud (Amazon EC2).
#
# Prerequisites:
#
# - The key pair to delete.
#
# @param ec2_client [Aws::EC2::Client] An initialized EC2 client.
# @param key_pair_name [String] The name of the key pair to delete.
# @return [Boolean] true if the key pair was deleted; otherwise, false.
# @example
#   exit 1 unless key_pair_deleted?(

```

```
# Aws::EC2::Client.new(region: 'us-west-2'),
# 'my-key-pair'
# )
def key_pair_deleted?(ec2_client, key_pair_name)
  ec2_client.delete_key_pair(key_name: key_pair_name)
  return true
rescue StandardError => e
  puts "Error deleting key pair: #{e.message}"
  return false
end

# Example usage:
def run_me
  key_pair_name = ""
  region = ""
  # Print usage information and then stop.
  if ARGV[0] == "--help" || ARGV[0] == "-h"
    puts "Usage:  ruby ec2-ruby-example-key-pairs.rb KEY_PAIR_NAME REGION"
    puts "Example: ruby ec2-ruby-example-key-pairs.rb my-key-pair us-west-2"
    exit 1
  # If no values are specified at the command prompt, use these default values.
  # Replace us-west-2 with the AWS Region you're using for Amazon EC2.
  elsif ARGV.count.zero?
    key_pair_name = "my-key-pair"
    region = "us-west-2"
  # Otherwise, use the values as specified at the command prompt.
  else
    key_pair_name = ARGV[0]
    region = ARGV[1]
  end

  ec2_client = Aws::EC2::Client.new(region: region)

  puts "Displaying existing key pair names before creating this key pair..."
  describe_key_pairs(ec2_client)

  puts "-" * 10
  puts "Creating key pair..."
  unless key_pair_created?(ec2_client, key_pair_name)
    puts "Stopping program."
    exit 1
  end

  puts "-" * 10
```

```
puts "Displaying existing key pair names after creating this key pair..."
describe_key_pairs(ec2_client)

puts "-" * 10
puts "Deleting key pair..."
unless key_pair_deleted?(ec2_client, key_pair_name)
  puts "Stopping program. You must delete the key pair yourself."
  exit 1
end
puts "Key pair deleted."

puts "-" * 10
puts "Now that the key pair is deleted, " \
      "also deleting the related private key pair file..."
filename = File.join(Dir.home, key_pair_name + ".pem")
File.delete(filename)
if File.exist?(filename)
  puts "Could not delete file at '#{filename}'. You must delete it yourself."
else
  puts "File deleted."
end

puts "-" * 10
puts "Displaying existing key pair names after deleting this key pair..."
describe_key_pairs(ec2_client)
end

run_me if $PROGRAM_NAME == __FILE__
```

- 如需 API 詳細資訊，請參閱 AWS SDK for Ruby API 參考[CreateKeyPair](#)中的。

建立子網

以下程式碼範例顯示如何建立 Amazon EC2 子網路。

適用於 Ruby 的開發套件

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫](#)。

```
require "aws-sdk-ec2"

# Creates a subnet within a virtual private cloud (VPC) in
# Amazon Virtual Private Cloud (Amazon VPC) and then tags
# the subnet.
#
# Prerequisites:
#
# - A VPC in Amazon VPC.
#
# @param ec2_resource [Aws::EC2::Resource] An initialized
#   Amazon Elastic Compute Cloud (Amazon EC2) resource object.
# @param vpc_id [String] The ID of the VPC for the subnet.
# @param cidr_block [String] The IPv4 CIDR block for the subnet.
# @param availability_zone [String] The ID of the Availability Zone
#   for the subnet.
# @param tag_key [String] The key portion of the tag for the subnet.
# @param tag_vlue [String] The value portion of the tag for the subnet.
# @return [Boolean] true if the subnet was created and tagged;
#   otherwise, false.
# @example
#   exit 1 unless subnet_created_and_tagged?(
#     Aws::EC2::Resource.new(region: 'us-west-2'),
#     'vpc-6713dfEX',
#     '10.0.0.0/24',
#     'us-west-2a',
#     'my-key',
#     'my-value'
#   )
def subnet_created_and_tagged?(
  ec2_resource,
  vpc_id,
  cidr_block,
  availability_zone,
  tag_key,
  tag_value
)
  subnet = ec2_resource.create_subnet(
    vpc_id: vpc_id,
    cidr_block: cidr_block,
    availability_zone: availability_zone
  )
end
```

```
subnet.create_tags(
  tags: [
    {
      key: tag_key,
      value: tag_value
    }
  ]
)
puts "Subnet created with ID '#{subnet.id}' in VPC with ID '#{vpc_id}' " \
    "and CIDR block '#{cidr_block}' in availability zone " \
    "'#{availability_zone}' and tagged with key '#{tag_key}' and " \
    "value '#{tag_value}'."
return true
rescue StandardError => e
  puts "Error creating or tagging subnet: #{e.message}"
  return false
end

# Example usage:
def run_me
  vpc_id = ""
  cidr_block = ""
  availability_zone = ""
  tag_key = ""
  tag_value = ""
  region = ""
  # Print usage information and then stop.
  if ARGV[0] == "--help" || ARGV[0] == "-h"
    puts "Usage:  ruby ec2-ruby-example-create-subnet.rb " \
        "VPC_ID CIDR_BLOCK AVAILABILITY_ZONE TAG_KEY TAG_VALUE REGION"
    # Replace us-west-2 with the AWS Region you're using for Amazon EC2.
    puts "Example: ruby ec2-ruby-example-create-subnet.rb " \
        "vpc-6713dfEX 10.0.0.0/24 us-west-2a my-key my-value us-west-2"
    exit 1
  # If no values are specified at the command prompt, use these default values.
  elsif ARGV.count.zero?
    vpc_id = "vpc-6713dfEX"
    cidr_block = "10.0.0.0/24"
    availability_zone = "us-west-2a"
    tag_key = "my-key"
    tag_value = "my-value"
    # Replace us-west-2 with the AWS Region you're using for Amazon EC2.
    region = "us-west-2"
  # Otherwise, use the values as specified at the command prompt.
```

```
else
  vpc_id = ARGV[0]
  cidr_block = ARGV[1]
  availability_zone = ARGV[2]
  tag_key = ARGV[3]
  tag_value = ARGV[4]
  region = ARGV[5]
end

ec2_resource = Aws::EC2::Resource.new(region: region)

if subnet_created_and_tagged?(
  ec2_resource,
  vpc_id,
  cidr_block,
  availability_zone,
  tag_key,
  tag_value
)
  puts "Subnet created and tagged."
else
  puts "Subnet not created or not tagged."
end

run_me if $PROGRAM_NAME == __FILE__
```

- 如需 API 詳細資訊，請參閱 AWS SDK for Ruby API 參考[CreateSubnet](#)中的。

描述區域

下列程式碼範例顯示如何描述 Amazon EC2 區域。

適用於 Ruby 的開發套件

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫](#)。

```
require "aws-sdk-ec2"

# @param ec2_client [Aws::EC2::Client] An initialized EC2 client.
# @example
#   list_regions_endpoints(Aws::EC2::Client.new(region: 'us-west-2'))
def list_regions_endpoints(ec2_client)
  result = ec2_client.describe_regions
  # Enable pretty printing.
  max_region_string_length = 16
  max_endpoint_string_length = 33
  # Print header.
  print "Region"
  print " " * (max_region_string_length - "Region".length)
  print "  Endpoint\n"
  print "-" * max_region_string_length
  print " "
  print "-" * max_endpoint_string_length
  print "\n"
  # Print Regions and their endpoints.
  result.regions.each do |region|
    print region.region_name
    print " " * (max_region_string_length - region.region_name.length)
    print " "
    print region.endpoint
    print "\n"
  end
end

# Displays a list of Amazon Elastic Compute Cloud (Amazon EC2)
# Availability Zones available to you depending on the AWS Region
# of the Amazon EC2 client.
#
# @param ec2_client [Aws::EC2::Client] An initialized EC2 client.
# @example
#   list_availability_zones(Aws::EC2::Client.new(region: 'us-west-2'))
def list_availability_zones(ec2_client)
  result = ec2_client.describe_availability_zones
  # Enable pretty printing.
  max_region_string_length = 16
  max_zone_string_length = 18
  max_state_string_length = 9
  # Print header.
  print "Region"
  print " " * (max_region_string_length - "Region".length)
```

```

print " Zone"
print " " * (max_zone_string_length - "Zone".length)
print " State\n"
print "-" * max_region_string_length
print " "
print "-" * max_zone_string_length
print " "
print "-" * max_state_string_length
print "\n"
# Print Regions, Availability Zones, and their states.
result.availability_zones.each do |zone|
  print zone.region_name
  print " " * (max_region_string_length - zone.region_name.length)
  print " "
  print zone.zone_name
  print " " * (max_zone_string_length - zone.zone_name.length)
  print " "
  print zone.state
  # Print any messages for this Availability Zone.
  if zone.messages.count.positive?
    print "\n"
    puts " Messages for this zone:"
    zone.messages.each do |message|
      print "   #{message.message}\n"
    end
  end
  print "\n"
end
end

# Example usage:
def run_me
  region = ""
  # Print usage information and then stop.
  if ARGV[0] == "--help" || ARGV[0] == "-h"
    puts "Usage: ruby ec2-ruby-example-regions-availability-zones.rb REGION"
    # Replace us-west-2 with the AWS Region you're using for Amazon EC2.
    puts "Example: ruby ec2-ruby-example-regions-availability-zones.rb us-west-2"
    exit 1
  # If no values are specified at the command prompt, use these default values.
  # Replace us-west-2 with the AWS Region you're using for Amazon EC2.
  elsif ARGV.count.zero?
    region = "us-west-2"
  # Otherwise, use the values as specified at the command prompt.

```



```

else
  region = ARGV[0]
end

ec2_client = Aws::EC2::Client.new(region: region)

puts "AWS Regions for Amazon EC2 that are available to you:"
list_regions_endpoints(ec2_client)
puts "\n\nAmazon EC2 Availability Zones that are available to you for AWS Region
'#{region}':"
list_availability_zones(ec2_client)
end

run_me if $PROGRAM_NAME == __FILE__

```

- 如需 API 詳細資訊，請參閱 AWS SDK for Ruby API 參考 [DescribeRegions](#) 中的。

描述執行個體

下列程式碼範例說明如何描述 Amazon EC2 執行個體。

適用於 Ruby 的開發套件

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在 [AWS 設定和執行程式碼範例儲存庫](#)。

```

require "aws-sdk-ec2"

# @param ec2_resource [Aws::EC2::Resource] An initialized EC2 resource object.
# @example
# list_instance_ids_states(Aws::EC2::Resource.new(region: 'us-west-2'))
def list_instance_ids_states(ec2_resource)
  response = ec2_resource.instances
  if response.count.zero?
    puts "No instances found."
  else
    puts "Instances -- ID, state:"
    response.each do |instance|

```

```

        puts "#{instance.id}, #{instance.state.name}"
      end
    end
  end
rescue StandardError => e
  puts "Error getting information about instances: #{e.message}"
end

# Example usage:
def run_me
  region = ""
  # Print usage information and then stop.
  if ARGV[0] == "--help" || ARGV[0] == "-h"
    puts "Usage:  ruby ec2-ruby-example-get-all-instance-info.rb REGION"
    # Replace us-west-2 with the AWS Region you're using for Amazon EC2.
    puts "Example: ruby ec2-ruby-example-get-all-instance-info.rb us-west-2"
    exit 1
  # If no values are specified at the command prompt, use these default values.
  # Replace us-west-2 with the AWS Region you're using for Amazon EC2.
  elsif ARGV.count.zero?
    region = "us-west-2"
  # Otherwise, use the values as specified at the command prompt.
  else
    region = ARGV[0]
  end
  ec2_resource = Aws::EC2::Resource.new(region: region)
  list_instance_ids_states(ec2_resource)
end

run_me if $PROGRAM_NAME == __FILE__

```

- 如需 API 詳細資訊，請參閱 AWS SDK for Ruby API 參考 [DescribeInstances](#) 中的。

釋出彈性 IP 地址

下列程式碼範例會示範如何釋放彈性 IP 位址。

適用於 Ruby 的開發套件

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在 [AWS 設定和執行程式碼範例儲存庫](#)。

```

# Releases an Elastic IP address from an
# Amazon Elastic Compute Cloud (Amazon EC2) instance.
#
# Prerequisites:
#
# - An Amazon EC2 instance with an associated Elastic IP address.
#
# @param ec2_client [Aws::EC2::Client] An initialized EC2 client.
# @param allocation_id [String] The ID of the allocation corresponding to
#   the Elastic IP address.
# @return [Boolean] true if the Elastic IP address was released;
#   otherwise, false.
# @example
#   exit 1 unless elastic_ip_address_released?(
#     Aws::EC2::Client.new(region: 'us-west-2'),
#     'eipalloc-04452e528a66279EX'
#   )
def elastic_ip_address_released?(ec2_client, allocation_id)
  ec2_client.release_address(allocation_id: allocation_id)
  return true
rescue StandardError => e
  puts("Error releasing Elastic IP address: #{e.message}")
  return false
end

```

- 如需 API 詳細資訊，請參閱 AWS SDK for Ruby API 參考[ReleaseAddress](#)中的。

啟動執行個體

下列程式碼範例示範如何啟動 Amazon EC2 執行個體。

適用於 Ruby 的開發套件

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫](#)。

```
require "aws-sdk-ec2"
```

```
# Attempts to start an Amazon Elastic Compute Cloud (Amazon EC2) instance.
#
# Prerequisites:
#
# - The Amazon EC2 instance.
#
# @param ec2_client [Aws::EC2::Client] An initialized EC2 client.
# @param instance_id [String] The ID of the instance.
# @return [Boolean] true if the instance was started; otherwise, false.
# @example
#   exit 1 unless instance_started?(
#     Aws::EC2::Client.new(region: 'us-west-2'),
#     'i-123abc'
#   )
def instance_started?(ec2_client, instance_id)
  response = ec2_client.describe_instance_status(instance_ids: [instance_id])

  if response.instance_statuses.count.positive?
    state = response.instance_statuses[0].instance_state.name
    case state
    when "pending"
      puts "Error starting instance: the instance is pending. Try again later."
      return false
    when "running"
      puts "The instance is already running."
      return true
    when "terminated"
      puts "Error starting instance: " \
        "the instance is terminated, so you cannot start it."
      return false
    end
  end
end

ec2_client.start_instances(instance_ids: [instance_id])
ec2_client.wait_until(:instance_running, instance_ids: [instance_id])
puts "Instance started."
return true
rescue StandardError => e
  puts "Error starting instance: #{e.message}"
  return false
end

# Example usage:
```

```
def run_me
  instance_id = ""
  region = ""
  # Print usage information and then stop.
  if ARGV[0] == "--help" || ARGV[0] == "-h"
    puts "Usage:  ruby ec2-ruby-example-start-instance-i-123abc.rb " \
         "INSTANCE_ID REGION "
  # Replace us-west-2 with the AWS Region you're using for Amazon EC2.
  puts "Example: ruby ec2-ruby-example-start-instance-i-123abc.rb " \
       "i-123abc us-west-2"
  exit 1
  # If no values are specified at the command prompt, use these default values.
  # Replace us-west-2 with the AWS Region you're using for Amazon EC2.
  elsif ARGV.count.zero?
    instance_id = "i-123abc"
    region = "us-west-2"
  # Otherwise, use the values as specified at the command prompt.
  else
    instance_id = ARGV[0]
    region = ARGV[1]
  end

  ec2_client = Aws::EC2::Client.new(region: region)

  puts "Attempting to start instance '#{instance_id}' " \
       "(this might take a few minutes)..."
  unless instance_started?(ec2_client, instance_id)
    puts "Could not start instance."
  end
end


run_me if $PROGRAM_NAME == __FILE__
```

- 如需 API 詳細資訊，請參閱 AWS SDK for Ruby API 參考[StartInstances](#)中的。

停止執行個體

下列程式碼範例示範如何停止 Amazon EC2 執行個體。

適用於 Ruby 的開發套件

 Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫](#)。

```
require "aws-sdk-ec2"

# Prerequisites:
#
# - The Amazon EC2 instance.
#
# @param ec2_client [Aws::EC2::Client] An initialized EC2 client.
# @param instance_id [String] The ID of the instance.
# @return [Boolean] true if the instance was stopped; otherwise, false.
# @example
#   exit 1 unless instance_stopped?(
#     Aws::EC2::Client.new(region: 'us-west-2'),
#     'i-123abc'
#   )
def instance_stopped?(ec2_client, instance_id)
  response = ec2_client.describe_instance_status(instance_ids: [instance_id])

  if response.instance_statuses.count.positive?
    state = response.instance_statuses[0].instance_state.name
    case state
    when "stopping"
      puts "The instance is already stopping."
      return true
    when "stopped"
      puts "The instance is already stopped."
      return true
    when "terminated"
      puts "Error stopping instance: " \
        "the instance is terminated, so you cannot stop it."
      return false
    end
  end
end

ec2_client.stop_instances(instance_ids: [instance_id])
ec2_client.wait_until(:instance_stopped, instance_ids: [instance_id])
```

```
    puts "Instance stopped."
    return true
  rescue StandardError => e
    puts "Error stopping instance: #{e.message}"
    return false
  end

# Example usage:
def run_me
  instance_id = ""
  region = ""
  # Print usage information and then stop.
  if ARGV[0] == "--help" || ARGV[0] == "-h"
    puts "Usage:  ruby ec2-ruby-example-stop-instance-i-123abc.rb " \
         "INSTANCE_ID REGION "
    # Replace us-west-2 with the AWS Region you're using for Amazon EC2.
    puts "Example: ruby ec2-ruby-example-start-instance-i-123abc.rb " \
         "i-123abc us-west-2"
    exit 1
  # If no values are specified at the command prompt, use these default values.
  # Replace us-west-2 with the AWS Region you're using for Amazon EC2.
  elsif ARGV.count.zero?
    instance_id = "i-123abc"
    region = "us-west-2"
  # Otherwise, use the values as specified at the command prompt.
  else
    instance_id = ARGV[0]
    region = ARGV[1]
  end

  ec2_client = Aws::EC2::Client.new(region: region)

  puts "Attempting to stop instance '#{instance_id}' " \
       "(this might take a few minutes)..."
  unless instance_stopped?(ec2_client, instance_id)
    puts "Could not stop instance."
  end
end

run_me if $PROGRAM_NAME == __FILE__
```

- 如需 API 詳細資訊，請參閱 AWS SDK for Ruby API 參考[StopInstances](#)中的。

終止執行個體

下列程式碼範例示範如何終止 Amazon EC2 執行個體。

適用於 Ruby 的開發套件

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫](#)。

```
require "aws-sdk-ec2"

# Prerequisites:
#
# - The Amazon EC2 instance.
#
# @param ec2_client [Aws::EC2::Client] An initialized EC2 client.
# @param instance_id [String] The ID of the instance.
# @return [Boolean] true if the instance was terminated; otherwise, false.
# @example
#   exit 1 unless instance_terminated?(
#     Aws::EC2::Client.new(region: 'us-west-2'),
#     'i-123abc'
#   )
def instance_terminated?(ec2_client, instance_id)
  response = ec2_client.describe_instance_status(instance_ids: [instance_id])

  if response.instance_statuses.count.positive? &&
    response.instance_statuses[0].instance_state.name == "terminated"

    puts "The instance is already terminated."
    return true
  end

  ec2_client.terminate_instances(instance_ids: [instance_id])
  ec2_client.wait_until(:instance_terminated, instance_ids: [instance_id])
  puts "Instance terminated."
  return true
rescue StandardError => e
  puts "Error terminating instance: #{e.message}"
  return false
end
```



```

end

# Example usage:
def run_me
  instance_id = ""
  region = ""
  # Print usage information and then stop.
  if ARGV[0] == "--help" || ARGV[0] == "-h"
    puts "Usage:  ruby ec2-ruby-example-terminate-instance-i-123abc.rb " \
         "INSTANCE_ID REGION "
    # Replace us-west-2 with the AWS Region you're using for Amazon EC2.
    puts "Example: ruby ec2-ruby-example-terminate-instance-i-123abc.rb " \
         "i-123abc us-west-2"
    exit 1
  # If no values are specified at the command prompt, use these default values.
  # Replace us-west-2 with the AWS Region you're using for Amazon EC2.
  elsif ARGV.count.zero?
    instance_id = "i-123abc"
    region = "us-west-2"
  # Otherwise, use the values as specified at the command prompt.
  else
    instance_id = ARGV[0]
    region = ARGV[1]
  end

  ec2_client = Aws::EC2::Client.new(region: region)

  puts "Attempting to terminate instance '#{instance_id}' " \
       "(this might take a few minutes)..."
  unless instance_terminated?(ec2_client, instance_id)
    puts "Could not terminate instance."
  end
end

run_me if $PROGRAM_NAME == __FILE__

```

- 如需 API 詳細資訊，請參閱 AWS SDK for Ruby API 參考 [TerminateInstances](#) 中的。

使用 SDK for Ruby Elastic Beanstalk 示例

下列程式碼範例會示範如何使用 Elastic Beanstalk 來執行動作及實作常見案例。AWS SDK for Ruby

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境和跨服務範例中查看內容中的動作。

案例是向您展示如何呼叫相同服務中的多個函數來完成特定任務的程式碼範例。

每個範例都包含一個連結 GitHub，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- [動作](#)

動作

描述應用

下列程式碼範例會示範如何描述 AWS Elastic Beanstalk 應用程式。

適用於 Ruby 的開發套件

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫](#)。

```
# Class to manage Elastic Beanstalk applications
class ElasticBeanstalkManager
  def initialize(eb_client, logger: Logger.new($stdout))
    @eb_client = eb_client
    @logger = logger
  end

  # Lists applications and their environments
  def list_applications
    @eb_client.describe_applications.applications.each do |application|
      log_application_details(application)
      list_environments(application.application_name)
    end
  rescue Aws::ElasticBeanstalk::Errors::ServiceError => e
    @logger.error("Elastic Beanstalk Service Error: #{e.message}")
  end

  private
end
```

```

# Logs application details
def log_application_details(application)
  @logger.info("Name:      #{application.application_name}")
  @logger.info("Description: #{application.description}")
end

# Lists and logs details of environments for a given application
def list_environments(application_name)
  @eb_client.describe_environments(application_name:
application_name).environments.each do |env|
    @logger.info("  Environment:  #{env.environment_name}")
    @logger.info("    URL:      #{env.cname}")
    @logger.info("    Health:   #{env.health}")
  end
  rescue Aws::ElasticBeanstalk::Errors::ServiceError => e
    @logger.error("Error listing environments for application #{application_name}:
#{e.message}")
  end
end
end

```

- 如需 API 詳細資訊，請參閱 AWS SDK for Ruby API 參考[DescribeApplications](#)中的。

列出堆疊

下列程式碼範例會示範如何列出 AWS Elastic Beanstalk 堆疊。

適用於 Ruby 的開發套件

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫](#)。

```

# Manages listing of AWS Elastic Beanstalk solution stacks
# @param [Aws::ElasticBeanstalk::Client] eb_client
# @param [String] filter - Returns subset of results based on match
# @param [Logger] logger
class StackLister
  # Initialize with AWS Elastic Beanstalk client
  def initialize(eb_client, filter, logger: Logger.new($stdout))

```

```
@eb_client = eb_client
@filter = filter.downcase
@logger = logger
end

# Lists and logs Elastic Beanstalk solution stacks
def list_stacks
  stacks = @eb_client.list_available_solution_stacks.solution_stacks
  orig_length = stacks.length
  filtered_length = 0

  stacks.each do |stack|
    if @filter.empty? || stack.downcase.include?(@filter)
      @logger.info(stack)
      filtered_length += 1
    end
  end

  log_summary(filtered_length, orig_length)
rescue Aws::Errors::ServiceError => e
  @logger.error("Error listing solution stacks: #{e.message}")
end

private

# Logs summary of listed stacks
def log_summary(filtered_length, orig_length)
  if @filter.empty?
    @logger.info("Showed #{orig_length} stack(s)")
  else
    @logger.info("Showed #{filtered_length} stack(s) of #{orig_length}")
  end
end
end
```

- 如需 API 詳細資訊，請參閱 AWS SDK for Ruby API 參考 [ListAvailableSolutionStacks](#) 中的。

更新應用

下列程式碼範例会示範如何更新 AWS Elastic Beanstalk 應用程式。

適用於 Ruby 的開發套件

 Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫](#)。

```
# Manages deployment of Rails applications to AWS Elastic Beanstalk
class RailsAppDeployer
  def initialize(eb_client, s3_client, app_name, logger: Logger.new($stdout))
    @eb_client = eb_client
    @s3_client = s3_client
    @app_name = app_name
    @logger = logger
  end

  # Deploys the latest application version to Elastic Beanstalk
  def deploy
    create_storage_location
    zip_file_name = create_zip_file
    upload_zip_to_s3(zip_file_name)
    create_and_deploy_new_application_version(zip_file_name)
  end

  private

  # Creates a new S3 storage location for the application
  def create_storage_location
    resp = @eb_client.create_storage_location
    @logger.info("Created storage location in bucket #{resp.s3_bucket}")
    rescue Aws::Errors::ServiceError => e
      @logger.error("Failed to create storage location: #{e.message}")
  end

  # Creates a ZIP file of the application using git
  def create_zip_file
    zip_file_basename = SecureRandom.urlsafe_base64
    zip_file_name = "#{zip_file_basename}.zip"
    `git archive --format=zip -o #{zip_file_name} HEAD`
    zip_file_name
  end

  # Uploads the ZIP file to the S3 bucket
```

```
def upload_zip_to_s3(zip_file_name)
  zip_contents = File.read(zip_file_name)
  key = "#{@app_name}/#{zip_file_name}"
  @s3_client.put_object(body: zip_contents, bucket: fetch_bucket_name, key: key)
rescue Aws::Errors::ServiceError => e
  @logger.error("Failed to upload ZIP file to S3: #{e.message}")
end

# Fetches the S3 bucket name from Elastic Beanstalk application versions
def fetch_bucket_name
  app_versions = @eb_client.describe_application_versions(application_name:
@app_name)
  av = app_versions.application_versions.first
  av.source_bundle.s3_bucket
rescue Aws::Errors::ServiceError => e
  @logger.error("Failed to fetch bucket name: #{e.message}")
  raise
end

# Creates a new application version and deploys it
def create_and_deploy_new_application_version(zip_file_name)
  version_label = File.basename(zip_file_name, ".zip")
  @eb_client.create_application_version(
    process: false,
    application_name: @app_name,
    version_label: version_label,
    source_bundle: {
      s3_bucket: fetch_bucket_name,
      s3_key: "#{@app_name}/#{zip_file_name}"
    },
    description: "Updated #{Time.now.strftime('%d/%m/%Y')}}"
  )
  update_environment(version_label)
rescue Aws::Errors::ServiceError => e
  @logger.error("Failed to create or deploy application version: #{e.message}")
end

# Updates the environment to the new application version
def update_environment(version_label)
  env_name = fetch_environment_name
  @eb_client.update_environment(
    environment_name: env_name,
    version_label: version_label
  )
end
```

```
rescue Aws::Errors::ServiceError => e
  @logger.error("Failed to update environment: #{e.message}")
end

# Fetches the environment name of the application
def fetch_environment_name
  envs = @eb_client.describe_environments(application_name: @app_name)
  envs.environments.first.environment_name
rescue Aws::Errors::ServiceError => e
  @logger.error("Failed to fetch environment name: #{e.message}")
  raise
end
end
```

- 如需 API 詳細資訊，請參閱 AWS SDK for Ruby API 參考[UpdateApplication](#)中的。

EventBridge 使用 SDK for Ruby 示例

下列程式碼範例說明如何使用 AWS SDK for Ruby 與來執行動作及實作常見案例 EventBridge。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境和跨服務範例中查看內容中的動作。

案例是向您展示如何呼叫相同服務中的多個函數來完成特定任務的程式碼範例。

每個範例都包含一個連結 GitHub，您可以在其中找到如何在內容中設定和執程式碼的指示。

主題


- [案例](#)

案例

建立和觸發規則

下列程式碼範例顯示如何在 Amazon 中建立和觸發規則 EventBridge。

適用於 Ruby 的開發套件

 Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫](#)。

以正確的順序呼叫函數。

```
require "aws-sdk-sns"
require "aws-sdk-iam"
require "aws-sdk-cloudwatchevents"
require "aws-sdk-ec2"
require "aws-sdk-cloudwatch"
require "aws-sdk-cloudwatchlogs"
require "securerandom"
```

檢查提供給此函數的主題中是否存在指定的 Amazon Simple Notification Service (Amazon SNS) 主題。

```
# Checks whether the specified Amazon SNS
# topic exists among those provided to this function.
# This is a helper function that is called by the topic_exists? function.
#
# @param topics [Array] An array of Aws::SNS::Types::Topic objects.
# @param topic_arn [String] The ARN of the topic to find.
# @return [Boolean] true if the topic ARN was found; otherwise, false.
# @example
#   sns_client = Aws::SNS::Client.new(region: 'us-east-1')
#   response = sns_client.list_topics
#   if topic_found?(
#     response.topics,
#     'arn:aws:sns:us-east-1:111111111111:aws-doc-sdk-examples-topic'
#   )
#     puts 'Topic found.'
#   end

def topic_found?(topics, topic_arn)
  topics.each do |topic|
    return true if topic.topic_arn == topic_arn
  end
end
```



```
    return false
  end
end
```

檢查 Amazon SNS 中呼叫者可用的主題中是否存在指定的主題。

```
# Checks whether the specified topic exists among those available to the
# caller in Amazon SNS.
#
# @param sns_client [Aws::SNS::Client] An initialized Amazon SNS client.
# @param topic_arn [String] The ARN of the topic to find.
# @return [Boolean] true if the topic ARN was found; otherwise, false.
# @example
#   exit 1 unless topic_exists?(
#     Aws::SNS::Client.new(region: 'us-east-1'),
#     'arn:aws:sns:us-east-1:111111111111:aws-doc-sdk-examples-topic'
#   )
def topic_exists?(sns_client, topic_arn)
  puts "Searching for topic with ARN '#{topic_arn}'..."
  response = sns_client.list_topics
  if response.topics.count.positive?
    if topic_found?(response.topics, topic_arn)
      puts "Topic found."
      return true
    end
    while response.next_page? do
      response = response.next_page
      if response.topics.count.positive?
        if topic_found?(response.topics, topic_arn)
          puts "Topic found."
          return true
        end
      end
    end
  end
  puts "Topic not found."
  return false
rescue StandardError => e
  puts "Topic not found: #{e.message}"
  return false
end
```

在 Amazon SNS 中建立主題，然後訂閱電子郵件地址以接收該主題的通知。

```
# Creates a topic in Amazon SNS
# and then subscribes an email address to receive notifications to that topic.
#
# @param sns_client [Aws::SNS::Client] An initialized Amazon SNS client.
# @param topic_name [String] The name of the topic to create.
# @param email_address [String] The email address of the recipient to notify.
# @return [String] The ARN of the topic that was created.
# @example
#   puts create_topic(
#     Aws::SNS::Client.new(region: 'us-east-1'),
#     'aws-doc-sdk-examples-topic',
#     'mary@example.com'
#   )
def create_topic(sns_client, topic_name, email_address)
  puts "Creating the topic named '#{topic_name}'..."
  topic_response = sns_client.create_topic(name: topic_name)
  puts "Topic created with ARN '#{topic_response.topic_arn}'."
  subscription_response = sns_client.subscribe(
    topic_arn: topic_response.topic_arn,
    protocol: "email",
    endpoint: email_address,
    return_subscription_arn: true
  )
  puts "Subscription created with ARN " \
    "'#{subscription_response.subscription_arn}'. Have the owner of the " \
    "'email address '#{email_address}' check their inbox in a few minutes " \
    "and confirm the subscription to start receiving notification emails."
  return topic_response.topic_arn
rescue StandardError => e
  puts "Error creating or subscribing to topic: #{e.message}"
  return "Error"
end
```

檢查提供給此函數的角色中是否存在指定的 AWS Identity and Access Management (IAM) 角色。

```
# Checks whether the specified AWS Identity and Access Management (IAM)
# role exists among those provided to this function.
# This is a helper function that is called by the role_exists? function.
#
```

```

# @param roles [Array] An array of Aws::IAM::Role objects.
# @param role_arn [String] The ARN of the role to find.
# @return [Boolean] true if the role ARN was found; otherwise, false.
# @example
#   iam_client = Aws::IAM::Client.new(region: 'us-east-1')
#   response = iam_client.list_roles
#   if role_found?(
#     response.roles,
#     'arn:aws:iam::111111111111:role/aws-doc-sdk-examples-ec2-state-change'
#   )
#     puts 'Role found.'
#   end
def role_found?(roles, role_arn)
  roles.each do |role|
    return true if role.arn == role_arn
  end
  return false
end

```

檢查 IAM 中呼叫者可用的角色中是否存在指定的角色。

```

# Checks whether the specified role exists among those available to the
# caller in AWS Identity and Access Management (IAM).
#
# @param iam_client [Aws::IAM::Client] An initialized IAM client.
# @param role_arn [String] The ARN of the role to find.
# @return [Boolean] true if the role ARN was found; otherwise, false.
# @example
#   exit 1 unless role_exists?(
#     Aws::IAM::Client.new(region: 'us-east-1'),
#     'arn:aws:iam::111111111111:role/aws-doc-sdk-examples-ec2-state-change'
#   )
def role_exists?(iam_client, role_arn)
  puts "Searching for role with ARN '#{role_arn}'..."
  response = iam_client.list_roles
  if response.roles.count.positive?
    if role_found?(response.roles, role_arn)
      puts "Role found."
      return true
    end
  end
  while response.next_page? do
    response = response.next_page
  end
end

```

```

    if response.roles.count.positive?
      if role_found?(response.roles, role_arn)
        puts "Role found."
        return true
      end
    end
  end
end
puts "Role not found."
return false
rescue StandardError => e
  puts "Role not found: #{e.message}"
  return false
end

```

在 IAM 中建立角色。

```

# Creates a role in AWS Identity and Access Management (IAM).
# This role is used by a rule in Amazon EventBridge to allow
# that rule to operate within the caller's account.
# This role is designed to be used specifically by this code example.
#
# @param iam_client [Aws::IAM::Client] An initialized IAM client.
# @param role_name [String] The name of the role to create.
# @return [String] The ARN of the role that was created.
# @example
#   puts create_role(
#     Aws::IAM::Client.new(region: 'us-east-1'),
#     'aws-doc-sdk-examples-ec2-state-change'
#   )
def create_role(iam_client, role_name)
  puts "Creating the role named '#{role_name}'..."
  response = iam_client.create_role(
    assume_role_policy_document: {
      'Version': "2012-10-17",
      'Statement': [
        {
          'Sid': "",
          'Effect': "Allow",
          'Principal': {
            'Service': "events.amazonaws.com"
          }
        }
      ]
    },

```

```

        'Action': "sts:AssumeRole"
      }
    ]
  }.to_json,
  path: "/",
  role_name: role_name
)
puts "Role created with ARN '#{response.role.arn}'."
puts "Adding access policy to role..."
iam_client.put_role_policy(
  policy_document: {
    'Version': "2012-10-17",
    'Statement': [
      {
        'Sid': "CloudWatchEventsFullAccess",
        'Effect': "Allow",
        'Resource': "*",
        'Action': "events:*"
      },
      {
        'Sid': "IAMPassRoleForCloudWatchEvents",
        'Effect': "Allow",
        'Resource': "arn:aws:iam::*:role/AWS_Events_Invoke_Targets",
        'Action': "iam:PassRole"
      }
    ]
  }.to_json,
  policy_name: "CloudWatchEventsPolicy",
  role_name: role_name
)
puts "Access policy added to role."
return response.role.arn
rescue StandardError => e
  puts "Error creating role or adding policy to it: #{e.message}"
  puts "If the role was created, you must add the access policy " \
    "to the role yourself, or delete the role yourself and try again."
  return "Error"
end

```

檢查指定的 EventBridge 規則是否存在於那些提供給此函數之間。

```
# Checks whether the specified Amazon EventBridge rule exists among
```

```

# those provided to this function.
# This is a helper function that is called by the rule_exists? function.
#
# @param rules [Array] An array of Aws::CloudWatchEvents::Types::Rule objects.
# @param rule_arn [String] The name of the rule to find.
# @return [Boolean] true if the name of the rule was found; otherwise, false.
# @example
#   cloudwatchevents_client = Aws::CloudWatch::Client.new(region: 'us-east-1')
#   response = cloudwatchevents_client.list_rules
#   if rule_found?(response.rules, 'aws-doc-sdk-examples-ec2-state-change')
#     puts 'Rule found.'
#   end
def rule_found?(rules, rule_name)
  rules.each do |rule|
    return true if rule.name == rule_name
  end
  return false
end

```

檢查指定的規則是否存在於可供呼叫者使用的規則之間 EventBridge。

```

# Checks whether the specified rule exists among those available to the
# caller in Amazon EventBridge.
#
# @param cloudwatchevents_client [Aws::CloudWatchEvents::Client]
#   An initialized Amazon EventBridge client.
# @param rule_name [String] The name of the rule to find.
# @return [Boolean] true if the rule name was found; otherwise, false.
# @example
#   exit 1 unless rule_exists?(
#     Aws::CloudWatch::Client.new(region: 'us-east-1')
#     'aws-doc-sdk-examples-ec2-state-change'
#   )
def rule_exists?(cloudwatchevents_client, rule_name)
  puts "Searching for rule with name '#{rule_name}'..."
  response = cloudwatchevents_client.list_rules
  if response.rules.count.positive?
    if rule_found?(response.rules, rule_name)
      puts "Rule found."
      return true
    end
  end
  while response.next_page? do

```

```

    response = response.next_page
    if response.rules.count.positive?
      if rule_found?(response.rules, rule_name)
        puts "Rule found."
        return true
      end
    end
  end
end
puts "Rule not found."
return false
rescue StandardError => e
  puts "Rule not found: #{e.message}"
  return false
end

```

在中建立規則 EventBridge。

```

# Creates a rule in Amazon EventBridge.
# This rule is triggered whenever an available instance in
# Amazon EC2 changes to the specified state.
# This rule is designed to be used specifically by this code example.
#
# Prerequisites:
#
# - A role in AWS Identity and Access Management (IAM) that is designed
#   to be used specifically by this code example.
# - A topic in Amazon SNS.
#
# @param cloudwatchevents_client [Aws::CloudWatchEvents::Client]
#   An initialized Amazon EventBridge client.
# @param rule_name [String] The name of the rule to create.
# @param rule_description [String] Some description for this rule.
# @param instance_state [String] The state that available instances in
#   Amazon EC2 must change to, to
#   trigger this rule.
# @param role_arn [String] The Amazon Resource Name (ARN) of the IAM role.
# @param target_id [String] Some identifying string for the rule's target.
# @param topic_arn [String] The ARN of the Amazon SNS topic.
# @return [Boolean] true if the rule was created; otherwise, false.
# @example
#   exit 1 unless rule_created?(

```

```
# Aws::CloudWatch::Client.new(region: 'us-east-1'),
# 'aws-doc-sdk-examples-ec2-state-change',
# 'Triggers when any available EC2 instance starts.',
# 'running',
# 'arn:aws:iam::111111111111:role/aws-doc-sdk-examples-ec2-state-change',
# 'sns-topic',
# 'arn:aws:sns:us-east-1:111111111111:aws-doc-sdk-examples-topic'
# )
def rule_created?(
  cloudwatchevents_client,
  rule_name,
  rule_description,
  instance_state,
  role_arn,
  target_id,
  topic_arn
)
  puts "Creating rule with name '#{rule_name}'..."
  put_rule_response = cloudwatchevents_client.put_rule(
    name: rule_name,
    description: rule_description,
    event_pattern: {
      'source': [
        "aws.ec2"
      ],
      'detail-type': [
        "EC2 Instance State-change Notification"
      ],
      'detail': {
        'state': [
          instance_state
        ]
      }
    }.to_json,
    state: "ENABLED",
    role_arn: role_arn
  )
  puts "Rule created with ARN '#{put_rule_response.rule_arn}'."

  put_targets_response = cloudwatchevents_client.put_targets(
    rule: rule_name,
    targets: [
      {
        id: target_id,
```



```

        arn: topic_arn
      }
    ]
  )
  if put_targets_response.key?(:failed_entry_count) &&
    put_targets_response.failed_entry_count > 0
    puts "Error(s) adding target to rule:"
    put_targets_response.failed_entries.each do |failure|
      puts failure.error_message
    end
    return false
  else
    return true
  end
rescue StandardError => e
  puts "Error creating rule or adding target to rule: #{e.message}"
  puts "If the rule was created, you must add the target " \
    "to the rule yourself, or delete the rule yourself and try again."
  return false
end
end

```

檢查 Amazon CloudWatch Logs 中呼叫者可用的記錄群組中是否存在指定的日誌群組。

```

# Checks to see whether the specified log group exists among those available
# to the caller in Amazon CloudWatch Logs.
#
# @param cloudwatchlogs_client [Aws::CloudWatchLogs::Client] An initialized
#   Amazon CloudWatch Logs client.
# @param log_group_name [String] The name of the log group to find.
# @return [Boolean] true if the log group name was found; otherwise, false.
# @example
#   exit 1 unless log_group_exists?(
#     Aws::CloudWatchLogs::Client.new(region: 'us-east-1'),
#     'aws-doc-sdk-examples-cloudwatch-log'
#   )
def log_group_exists?(cloudwatchlogs_client, log_group_name)
  puts "Searching for log group with name '#{log_group_name}'..."
  response = cloudwatchlogs_client.describe_log_groups(
    log_group_name_prefix: log_group_name
  )
  if response.log_groups.count.positive?
    response.log_groups.each do |log_group|

```

```

    if log_group.log_group_name == log_group_name
      puts "Log group found."
      return true
    end
  end
end
puts "Log group not found."
return false
rescue StandardError => e
  puts "Log group not found: #{e.message}"
  return false
end
end

```

在記錄檔中建立 CloudWatch 記錄群組。

```

# Creates a log group in Amazon CloudWatch Logs.
#
# @param cloudwatchlogs_client [Aws::CloudWatchLogs::Client] An initialized
#   Amazon CloudWatch Logs client.
# @param log_group_name [String] The name of the log group to create.
# @return [Boolean] true if the log group name was created; otherwise, false.
# @example
#   exit 1 unless log_group_created?(
#     Aws::CloudWatchLogs::Client.new(region: 'us-east-1'),
#     'aws-doc-sdk-examples-cloudwatch-log'
#   )
def log_group_created?(cloudwatchlogs_client, log_group_name)
  puts "Attempting to create log group with the name '#{log_group_name}'..."
  cloudwatchlogs_client.create_log_group(log_group_name: log_group_name)
  puts "Log group created."
  return true
rescue StandardError => e
  puts "Error creating log group: #{e.message}"
  return false
end
end

```

在記錄檔中將事件寫入 CloudWatch 記錄資料流。

```

# Writes an event to a log stream in Amazon CloudWatch Logs.
#
# Prerequisites:

```

```
#
# - A log group in Amazon CloudWatch Logs.
# - A log stream within the log group.
#
# @param cloudwatchlogs_client [Aws::CloudWatchLogs::Client] An initialized
#   Amazon CloudWatch Logs client.
# @param log_group_name [String] The name of the log group.
# @param log_stream_name [String] The name of the log stream within
#   the log group.
# @param message [String] The message to write to the log stream.
# @param sequence_token [String] If available, the sequence token from the
#   message that was written immediately before this message. This sequence
#   token is returned by Amazon CloudWatch Logs whenever you programmatically
#   write a message to the log stream.
# @return [String] The sequence token that is returned by
#   Amazon CloudWatch Logs after successfully writing the message to the
#   log stream.
# @example
#   puts log_event(
#     Aws::EC2::Client.new(region: 'us-east-1'),
#     'aws-doc-sdk-examples-cloudwatch-log'
#     '2020/11/19/53f985be-199f-408e-9a45-fc242df41fEX',
#     "Instance 'i-033c48ef067af3dEX' restarted.",
#     '495426724868310740095796045676567882148068632824696073EX'
#   )
def log_event(
  cloudwatchlogs_client,
  log_group_name,
  log_stream_name,
  message,
  sequence_token
)
  puts "Attempting to log '#{message}' to log stream '#{log_stream_name}'..."
  event = {
    log_group_name: log_group_name,
    log_stream_name: log_stream_name,
    log_events: [
      {
        timestamp: (Time.now.utc.to_f.round(3) * 1_000).to_i,
        message: message
      }
    ]
  }
  unless sequence_token.empty?
```

```
    event[:sequence_token] = sequence_token
  end

  response = cloudwatchlogs_client.put_log_events(event)
  puts "Message logged."
  return response.next_sequence_token
rescue StandardError => e
  puts "Message not logged: #{e.message}"
end
```

重新啟動 Amazon Elastic Compute Cloud (Amazon EC2) 執行個體，並將相關活動的相關資訊新增至 CloudWatch 日誌中的日誌串流。

```
# Restarts an Amazon EC2 instance
# and adds information about the related activity to a log stream
# in Amazon CloudWatch Logs.
#
# Prerequisites:
#
# - The Amazon EC2 instance to restart.
# - The log group in Amazon CloudWatch Logs to add related activity
#   information to.
#
# @param ec2_client [Aws::EC2::Client] An initialized Amazon EC2 client.
# @param cloudwatchlogs_client [Aws::CloudWatchLogs::Client]
#   An initialized Amazon CloudWatch Logs client.
# @param instance_id [String] The ID of the instance.
# @param log_group_name [String] The name of the log group.
# @return [Boolean] true if the instance was restarted and the information
#   was written to the log stream; otherwise, false.
# @example
#   exit 1 unless instance_restarted?(
#     Aws::EC2::Client.new(region: 'us-east-1'),
#     Aws::CloudWatchLogs::Client.new(region: 'us-east-1'),
#     'i-033c48ef067af3dEX',
#     'aws-doc-sdk-examples-cloudwatch-log'
#   )
def instance_restarted?(
  ec2_client,
  cloudwatchlogs_client,
  instance_id,
```

```
    log_group_name
  )
  log_stream_name = "#{Time.now.year}/#{Time.now.month}/#{Time.now.day}/" \
    "#{SecureRandom.uuid}"
  cloudwatchlogs_client.create_log_stream(
    log_group_name: log_group_name,
    log_stream_name: log_stream_name
  )
  sequence_token = ""

  puts "Attempting to stop the instance with the ID '#{instance_id}'. " \
    "This might take a few minutes..."
  ec2_client.stop_instances(instance_ids: [instance_id])
  ec2_client.wait_until(:instance_stopped, instance_ids: [instance_id])
  puts "Instance stopped."
  sequence_token = log_event(
    cloudwatchlogs_client,
    log_group_name,
    log_stream_name,
    "Instance '#{instance_id}' stopped.",
    sequence_token
  )

  puts "Attempting to restart the instance. This might take a few minutes..."
  ec2_client.start_instances(instance_ids: [instance_id])
  ec2_client.wait_until(:instance_running, instance_ids: [instance_id])
  puts "Instance restarted."
  sequence_token = log_event(
    cloudwatchlogs_client,
    log_group_name,
    log_stream_name,
    "Instance '#{instance_id}' restarted.",
    sequence_token
  )

  return true
rescue StandardError => e
  puts "Error creating log stream or stopping or restarting the instance: " \
    "#{e.message}"
  log_event(
    cloudwatchlogs_client,
    log_group_name,
    log_stream_name,
    "Error stopping or starting instance '#{instance_id}': #{e.message}",
  )
end
```

```

    sequence_token
  )
  return false
end

```

顯示中規則活動的相關資訊 EventBridge。

```

# Displays information about activity for a rule in Amazon EventBridge.
#
# Prerequisites:
#
# - A rule in Amazon EventBridge.
#
# @param cloudwatch_client [Amazon::CloudWatch::Client] An initialized
#   Amazon CloudWatch client.
# @param rule_name [String] The name of the rule.
# @param start_time [Time] The timestamp that determines the first datapoint
#   to return. Can also be expressed as DateTime, Date, Integer, or String.
# @param end_time [Time] The timestamp that determines the last datapoint
#   to return. Can also be expressed as DateTime, Date, Integer, or String.
# @param period [Integer] The interval, in seconds, to check for activity.
# @example
#   display_rule_activity(
#     Aws::CloudWatch::Client.new(region: 'us-east-1'),
#     'aws-doc-sdk-examples-ec2-state-change',
#     Time.now - 600, # Start checking from 10 minutes ago.
#     Time.now, # Check up until now.
#     60 # Check every minute during those 10 minutes.
#   )
def display_rule_activity(
  cloudwatch_client,
  rule_name,
  start_time,
  end_time,
  period
)
  puts "Attempting to display rule activity..."
  response = cloudwatch_client.get_metric_statistics(
    namespace: "AWS/Events",
    metric_name: "Invocations",
    dimensions: [
      {

```

```

        name: "RuleName",
        value: rule_name
      }
    ],
    start_time: start_time,
    end_time: end_time,
    period: period,
    statistics: ["Sum"],
    unit: "Count"
  )

  if response.key?(:datapoints) && response.datapoints.count.positive?
    puts "The event rule '#{rule_name}' was triggered:"
    response.datapoints.each do |datapoint|
      puts "  #{datapoint.sum} time(s) at #{datapoint.timestamp}"
    end
  else
    puts "The event rule '#{rule_name}' was not triggered during the " \
      "specified time period."
  end
rescue StandardError => e
  puts "Error getting information about event rule activity: #{e.message}"
end

```

顯示記錄檔群組中所有記錄資料流的 CloudWatch 記錄資訊。

```

# Displays log information for all of the log streams in a log group in
# Amazon CloudWatch Logs.
#
# Prerequisites:
#
# - A log group in Amazon CloudWatch Logs.
#
# @param cloudwatchlogs_client [Amazon::CloudWatchLogs::Client] An initialized
#   Amazon CloudWatch Logs client.
# @param log_group_name [String] The name of the log group.
# @example
#   display_log_data(
#     Amazon::CloudWatchLogs::Client.new(region: 'us-east-1'),
#     'aws-doc-sdk-examples-cloudwatch-log'
#   )
def display_log_data(cloudwatchlogs_client, log_group_name)

```

```

puts "Attempting to display log stream data for the log group " \
     "named '#{log_group_name}'..."
describe_log_streams_response = cloudwatchlogs_client.describe_log_streams(
  log_group_name: log_group_name,
  order_by: "LastEventTime",
  descending: true
)
if describe_log_streams_response.key?(:log_streams) &&
  describe_log_streams_response.log_streams.count.positive?
  describe_log_streams_response.log_streams.each do |log_stream|
    get_log_events_response = cloudwatchlogs_client.get_log_events(
      log_group_name: log_group_name,
      log_stream_name: log_stream.log_stream_name
    )
    puts "\nLog messages for '#{log_stream.log_stream_name}':"
    puts "-" * (log_stream.log_stream_name.length + 20)
    if get_log_events_response.key?(:events) &&
      get_log_events_response.events.count.positive?
      get_log_events_response.events.each do |event|
        puts event.message
      end
    else
      puts "No log messages for this log stream."
    end
  end
end
rescue StandardError => e
  puts "Error getting information about the log streams or their messages: " \
       "#{e.message}"
end

```

向來電者顯示提醒，以手動清除他們不再需要的任何相關 AWS 資源。

```

# Displays a reminder to the caller to manually clean up any associated
# AWS resources that they no longer need.
#
# @param topic_name [String] The name of the Amazon SNS topic.
# @param role_name [String] The name of the IAM role.
# @param rule_name [String] The name of the Amazon EventBridge rule.
# @param log_group_name [String] The name of the Amazon CloudWatch Logs log group.
# @param instance_id [String] The ID of the Amazon EC2 instance.

```



```
# @example
#   manual_cleanup_notice(
#     'aws-doc-sdk-examples-topic',
#     'aws-doc-sdk-examples-cloudwatch-events-rule-role',
#     'aws-doc-sdk-examples-ec2-state-change',
#     'aws-doc-sdk-examples-cloudwatch-log',
#     'i-033c48ef067af3dEX'
#   )
def manual_cleanup_notice(
  topic_name, role_name, rule_name, log_group_name, instance_id
)
  puts "-" * 10
  puts "Some of the following AWS resources might still exist in your account."
  puts "If you no longer want to use this code example, then to clean up"
  puts "your AWS account and avoid unexpected costs, you might want to"
  puts "manually delete any of the following resources if they exist:"
  puts "- The Amazon SNS topic named '#{topic_name}'."
  puts "- The IAM role named '#{role_name}'."
  puts "- The Amazon EventBridge rule named '#{rule_name}'."
  puts "- The Amazon CloudWatch Logs log group named '#{log_group_name}'."
  puts "- The Amazon EC2 instance with the ID '#{instance_id}'."
end

# Example usage:
def run_me
  # Properties for the Amazon SNS topic.
  topic_name = "aws-doc-sdk-examples-topic"
  email_address = "mary@example.com"
  # Properties for the IAM role.
  role_name = "aws-doc-sdk-examples-cloudwatch-events-rule-role"
  # Properties for the Amazon EventBridge rule.
  rule_name = "aws-doc-sdk-examples-ec2-state-change"
  rule_description = "Triggers when any available EC2 instance starts."
  instance_state = "running"
  target_id = "sns-topic"
  # Properties for the Amazon EC2 instance.
  instance_id = "i-033c48ef067af3dEX"
  # Properties for displaying the event rule's activity.
  start_time = Time.now - 600 # Go back over the past 10 minutes
                                # (10 minutes * 60 seconds = 600 seconds).
  end_time = Time.now
  period = 60 # Look back every 60 seconds over the past 10 minutes.
  # Properties for the Amazon CloudWatch Logs log group.
  log_group_name = "aws-doc-sdk-examples-cloudwatch-log"
```

```
# AWS service clients for this code example.
region = "us-east-1"
sts_client = Aws::STS::Client.new(region: region)
sns_client = Aws::SNS::Client.new(region: region)
iam_client = Aws::IAM::Client.new(region: region)
cloudwatchevents_client = Aws::CloudWatchEvents::Client.new(region: region)
ec2_client = Aws::EC2::Client.new(region: region)
cloudwatch_client = Aws::CloudWatch::Client.new(region: region)
cloudwatchlogs_client = Aws::CloudWatchLogs::Client.new(region: region)

# Get the caller's account ID for use in forming
# Amazon Resource Names (ARNs) that this code relies on later.
account_id = sts_client.get_caller_identity.account

# If the Amazon SNS topic doesn't exist, create it.
topic_arn = "arn:aws:sns:#{region}:#{account_id}:#{topic_name}"
unless topic_exists?(sns_client, topic_arn)
  topic_arn = create_topic(sns_client, topic_name, email_address)
  if topic_arn == "Error"
    puts "Could not create the Amazon SNS topic correctly. Program stopped."
    manual_cleanup_notice(
      topic_name, role_name, rule_name, log_group_name, instance_id
    )
    exit 1
  end
end

# If the IAM role doesn't exist, create it.
role_arn = "arn:aws:iam:#{account_id}:role/#{role_name}"
unless role_exists?(iam_client, role_arn)
  role_arn = create_role(iam_client, role_name)
  if role_arn == "Error"
    puts "Could not create the IAM role correctly. Program stopped."
    manual_cleanup_notice(
      topic_name, role_name, rule_name, log_group_name, instance_id
    )
  end
end

# If the Amazon EventBridge rule doesn't exist, create it.
unless rule_exists?(cloudwatchevents_client, rule_name)
  unless rule_created?(
    cloudwatchevents_client,
    rule_name,
```

```
    rule_description,
    instance_state,
    role_arn,
    target_id,
    topic_arn
  )
  puts "Could not create the Amazon EventBridge rule correctly. " \
    "Program stopped."
  manual_cleanup_notice(
    topic_name, role_name, rule_name, log_group_name, instance_id
  )
end
end

# If the Amazon CloudWatch Logs log group doesn't exist, create it.
unless log_group_exists?(cloudwatchlogs_client, log_group_name)
  unless log_group_created?(cloudwatchlogs_client, log_group_name)
    puts "Could not create the Amazon CloudWatch Logs log group " \
      "correctly. Program stopped."
    manual_cleanup_notice(
      topic_name, role_name, rule_name, log_group_name, instance_id
    )
  end
end

# Restart the Amazon EC2 instance, which triggers the rule.
unless instance_restarted?(
  ec2_client,
  cloudwatchlogs_client,
  instance_id,
  log_group_name
)
  puts "Could not restart the instance to trigger the rule. " \
    "Continuing anyway to show information about the rule and logs..."
end

# Display how many times the rule was triggered over the past 10 minutes.
display_rule_activity(
  cloudwatch_client,
  rule_name,
  start_time,
  end_time,
  period
)
```

```
# Display related log data in Amazon CloudWatch Logs.
display_log_data(cloudwatchlogs_client, log_group_name)

# Reminder the caller to clean up any AWS resources that are used
# by this code example and are no longer needed.
manual_cleanup_notice(
  topic_name, role_name, rule_name, log_group_name, instance_id
)
end

run_me if $PROGRAM_NAME == __FILE__
```

- 如需 API 詳細資訊，請參閱《AWS SDK for Ruby API 參考》中的下列主題。
 - [PutEvents](#)
 - [PutRule](#)

AWS Glue 使用 SDK for Ruby 示例

下列程式碼範例說明如何使用 AWS SDK for Ruby 與來執行動作及實作常見案例 AWS Glue。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境和跨服務範例中查看內容中的動作。

案例是向您展示如何呼叫相同服務中的多個函數來完成特定任務的程式碼範例。

每個範例都包含一個連結 GitHub，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- [動作](#)
- [案例](#)

動作

建立爬蟲程式

下列程式碼範例會示範如何建立 AWS Glue 爬行者程式。

適用於 Ruby 的開發套件

 Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫](#)。

```
# The `GlueWrapper` class serves as a wrapper around the AWS Glue API, providing a
simplified interface for common operations.
# It encapsulates the functionality of the AWS SDK for Glue and provides methods for
interacting with Glue crawlers, databases, tables, jobs, and S3 resources.
# The class initializes with a Glue client and a logger, allowing it to make API
calls and log any errors or informational messages.
class GlueWrapper
  def initialize(glue_client, logger)
    @glue_client = glue_client
    @logger = logger
  end

  # Creates a new crawler with the specified configuration.
  #
  # @param name [String] The name of the crawler.
  # @param role_arn [String] The ARN of the IAM role to be used by the crawler.
  # @param db_name [String] The name of the database where the crawler stores its
  metadata.
  # @param db_prefix [String] The prefix to be added to the names of tables that the
  crawler creates.
  # @param s3_target [String] The S3 path that the crawler will crawl.
  # @return [void]
  def create_crawler(name, role_arn, db_name, db_prefix, s3_target)
    @glue_client.create_crawler(
      name: name,
      role: role_arn,
      database_name: db_name,
      targets: {
        s3_targets: [
          {
            path: s3_target
          }
        ]
      }
    )
  end
end
```

```
rescue Aws::Glue::Errors::GlueException => e
  @logger.error("Glue could not create crawler: \n#{e.message}")
  raise
end
```

- 如需 API 詳細資訊，請參閱 AWS SDK for Ruby API 參考[CreateCrawler](#)中的。

建立任務定義

下列程式碼範例會示範如何建立 AWS Glue 工作定義。

適用於 Ruby 的開發套件

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫](#)。

```
# The `GlueWrapper` class serves as a wrapper around the AWS Glue API, providing a
# simplified interface for common operations.
# It encapsulates the functionality of the AWS SDK for Glue and provides methods for
# interacting with Glue crawlers, databases, tables, jobs, and S3 resources.
# The class initializes with a Glue client and a logger, allowing it to make API
# calls and log any errors or informational messages.
class GlueWrapper
  def initialize(glue_client, logger)
    @glue_client = glue_client
    @logger = logger
  end

  # Creates a new job with the specified configuration.
  #
  # @param name [String] The name of the job.
  # @param description [String] The description of the job.
  # @param role_arn [String] The ARN of the IAM role to be used by the job.
  # @param script_location [String] The location of the ETL script for the job.
  # @return [void]
  def create_job(name, description, role_arn, script_location)
    @glue_client.create_job(
      name: name,
```

```
description: description,
role: role_arn,
command: {
  name: "glueetl",
  script_location: script_location,
  python_version: "3"
},
glue_version: "3.0"
)
rescue Aws::Glue::Errors::GlueException => e
  @logger.error("Glue could not create job #{name}: \n#{e.message}")
  raise
end
```

- 如需 API 詳細資訊，請參閱 AWS SDK for Ruby API 參考[CreateJob](#)中的。

刪除爬蟲程式

下列程式碼範例顯示如何刪除 AWS Glue 爬行者程式。

適用於 Ruby 的開發套件

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫](#)。

```
# The `GlueWrapper` class serves as a wrapper around the AWS Glue API, providing a
simplified interface for common operations.
# It encapsulates the functionality of the AWS SDK for Glue and provides methods for
interacting with Glue crawlers, databases, tables, jobs, and S3 resources.
# The class initializes with a Glue client and a logger, allowing it to make API
calls and log any errors or informational messages.
class GlueWrapper
  def initialize(glue_client, logger)
    @glue_client = glue_client
    @logger = logger
  end

  # Deletes a crawler with the specified name.
```

```
#
# @param name [String] The name of the crawler to delete.
# @return [void]
def delete_crawler(name)
  @glue_client.delete_crawler(name: name)
rescue Aws::Glue::Errors::ServiceError => e
  @logger.error("Glue could not delete crawler #{name}: \n#{e.message}")
  raise
end
```

- 如需 API 詳細資訊，請參閱 AWS SDK for Ruby API 參考 [DeleteCrawler](#) 中的。

從 Data Catalog 中刪除資料庫

下列程式碼範例示範如何從中刪除資料庫 AWS Glue Data Catalog。

適用於 Ruby 的開發套件

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在 [AWS 設定和執行程式碼範例儲存庫](#)。

```
# The `GlueWrapper` class serves as a wrapper around the AWS Glue API, providing a
simplified interface for common operations.
# It encapsulates the functionality of the AWS SDK for Glue and provides methods for
interacting with Glue crawlers, databases, tables, jobs, and S3 resources.
# The class initializes with a Glue client and a logger, allowing it to make API
calls and log any errors or informational messages.
class GlueWrapper
  def initialize(glue_client, logger)
    @glue_client = glue_client
    @logger = logger
  end

  # Removes a specified database from a Data Catalog.
  #
  # @param database_name [String] The name of the database to delete.
  # @return [void]
  def delete_database(database_name)
```



```
@glue_client.delete_database(name: database_name)
rescue Aws::Glue::Errors::ServiceError => e
  @logger.error("Glue could not delete database: \n#{e.message}")
end
```

- 如需 API 詳細資訊，請參閱 AWS SDK for Ruby API 參考[DeleteDatabase](#)中的。

刪除任務定義

下列程式碼範例顯示如何刪除 AWS Glue 工作定義和所有關聯的執行。

適用於 Ruby 的開發套件

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫](#)。

```
# The `GlueWrapper` class serves as a wrapper around the AWS Glue API, providing a
# simplified interface for common operations.
# It encapsulates the functionality of the AWS SDK for Glue and provides methods for
# interacting with Glue crawlers, databases, tables, jobs, and S3 resources.
# The class initializes with a Glue client and a logger, allowing it to make API
# calls and log any errors or informational messages.
class GlueWrapper
  def initialize(glue_client, logger)
    @glue_client = glue_client
    @logger = logger
  end

  # Deletes a job with the specified name.
  #
  # @param job_name [String] The name of the job to delete.
  # @return [void]
  def delete_job(job_name)
    @glue_client.delete_job(job_name: job_name)
  rescue Aws::Glue::Errors::ServiceError => e
    @logger.error("Glue could not delete job: \n#{e.message}")
  end
end
```

- 如需 API 詳細資訊，請參閱 AWS SDK for Ruby API 參考[DeleteJob](#)中的。

從資料庫中刪除資料表

下列程式碼範例會示範如何從 AWS Glue Data Catalog 資料庫中刪除資料表。

適用於 Ruby 的開發套件

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫](#)。

```
# The `GlueWrapper` class serves as a wrapper around the AWS Glue API, providing a
simplified interface for common operations.
# It encapsulates the functionality of the AWS SDK for Glue and provides methods for
interacting with Glue crawlers, databases, tables, jobs, and S3 resources.
# The class initializes with a Glue client and a logger, allowing it to make API
calls and log any errors or informational messages.
class GlueWrapper
  def initialize(glue_client, logger)
    @glue_client = glue_client
    @logger = logger
  end

  # Deletes a table with the specified name.
  #
  # @param database_name [String] The name of the catalog database in which the
table resides.
  # @param table_name [String] The name of the table to be deleted.
  # @return [void]
  def delete_table(database_name, table_name)
    @glue_client.delete_table(database_name: database_name, name: table_name)
  rescue Aws::Glue::Errors::ServiceError => e
    @logger.error("Glue could not delete job: \n#{e.message}")
  end
end
```

- 如需 API 詳細資訊，請參閱 AWS SDK for Ruby API 參考[DeleteTable](#)中的。

取得爬蟲程式

下列程式碼範例會示範如何取得 AWS Glue 搜尋器。

適用於 Ruby 的開發套件

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫](#)。

```
# The `GlueWrapper` class serves as a wrapper around the AWS Glue API, providing a
# simplified interface for common operations.
# It encapsulates the functionality of the AWS SDK for Glue and provides methods for
# interacting with Glue crawlers, databases, tables, jobs, and S3 resources.
# The class initializes with a Glue client and a logger, allowing it to make API
# calls and log any errors or informational messages.
class GlueWrapper
  def initialize(glue_client, logger)
    @glue_client = glue_client
    @logger = logger
  end

  # Retrieves information about a specific crawler.
  #
  # @param name [String] The name of the crawler to retrieve information about.
  # @return [Aws::Glue::Types::Crawler, nil] The crawler object if found, or nil if
  # not found.
  def get_crawler(name)
    @glue_client.get_crawler(name: name)
  rescue Aws::Glue::Errors::EntityNotFoundException
    @logger.info("Crawler #{name} doesn't exist.")
    false
  rescue Aws::Glue::Errors::GlueException => e
    @logger.error("Glue could not get crawler #{name}: \n#{e.message}")
    raise
  end
end
```

- 如需 API 詳細資訊，請參閱 AWS SDK for Ruby API 參考[GetCrawler](#)中的。

從 Data Catalog 中取得資料庫

下列程式碼範例會示範如何從中取得資料庫 AWS Glue Data Catalog。

適用於 Ruby 的開發套件

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫](#)。

```
# The `GlueWrapper` class serves as a wrapper around the AWS Glue API, providing a
simplified interface for common operations.
# It encapsulates the functionality of the AWS SDK for Glue and provides methods for
interacting with Glue crawlers, databases, tables, jobs, and S3 resources.
# The class initializes with a Glue client and a logger, allowing it to make API
calls and log any errors or informational messages.
class GlueWrapper
  def initialize(glue_client, logger)
    @glue_client = glue_client
    @logger = logger
  end

  # Retrieves information about a specific database.
  #
  # @param name [String] The name of the database to retrieve information about.
  # @return [Aws::Glue::Types::Database, nil] The database object if found, or nil
  if not found.
  def get_database(name)
    response = @glue_client.get_database(name: name)
    response.database
  rescue Aws::Glue::Errors::GlueException => e
    @logger.error("Glue could not get database #{name}: \n#{e.message}")
    raise
  end
end
```

- 如需 API 詳細資訊，請參閱 AWS SDK for Ruby API 參考[GetDatabase](#)中的。

取得任務執行

下列程式碼範例會示範如何取得 AWS Glue 工作執行。

適用於 Ruby 的開發套件

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫](#)。

```
# The `GlueWrapper` class serves as a wrapper around the AWS Glue API, providing a
simplified interface for common operations.
# It encapsulates the functionality of the AWS SDK for Glue and provides methods for
interacting with Glue crawlers, databases, tables, jobs, and S3 resources.
# The class initializes with a Glue client and a logger, allowing it to make API
calls and log any errors or informational messages.
class GlueWrapper
  def initialize(glue_client, logger)
    @glue_client = glue_client
    @logger = logger
  end

  # Retrieves data for a specific job run.
  #
  # @param job_name [String] The name of the job run to retrieve data for.
  # @return [Glue::Types::GetJobRunResponse]
  def get_job_run(job_name, run_id)
    @glue_client.get_job_run(job_name: job_name, run_id: run_id)
  rescue Aws::Glue::Errors::GlueException => e
    @logger.error("Glue could not get job runs: \n#{e.message}")
  end
end
```

- 如需 API 詳細資訊，請參閱 AWS SDK for Ruby API 參考[GetJobRun](#)中的。

取得任務的執行

下列程式碼範例會示範如何取得 AWS Glue 工作的執行。

適用於 Ruby 的開發套件

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫](#)。

```
# The `GlueWrapper` class serves as a wrapper around the AWS Glue API, providing a
simplified interface for common operations.
# It encapsulates the functionality of the AWS SDK for Glue and provides methods for
interacting with Glue crawlers, databases, tables, jobs, and S3 resources.
# The class initializes with a Glue client and a logger, allowing it to make API
calls and log any errors or informational messages.
class GlueWrapper
  def initialize(glue_client, logger)
    @glue_client = glue_client
    @logger = logger
  end

  # Retrieves a list of job runs for the specified job.
  #
  # @param job_name [String] The name of the job to retrieve job runs for.
  # @return [Array<Aws::Glue::Types::JobRun>]
  def get_job_runs(job_name)
    response = @glue_client.get_job_runs(job_name: job_name)
    response.job_runs
  rescue Aws::Glue::Errors::GlueException => e
    @logger.error("Glue could not get job runs: \n#{e.message}")
  end
end
```

- 如需 API 詳細資訊，請參閱 AWS SDK for Ruby API 參考[GetJobRuns](#)中的。

從資料庫中取得資料表

下列程式碼範例示範如何從中的資料庫取得資料表 AWS Glue Data Catalog。

適用於 Ruby 的開發套件

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫](#)。

```
# The `GlueWrapper` class serves as a wrapper around the AWS Glue API, providing a
simplified interface for common operations.
# It encapsulates the functionality of the AWS SDK for Glue and provides methods for
interacting with Glue crawlers, databases, tables, jobs, and S3 resources.
# The class initializes with a Glue client and a logger, allowing it to make API
calls and log any errors or informational messages.
class GlueWrapper
  def initialize(glue_client, logger)
    @glue_client = glue_client
    @logger = logger
  end

  # Retrieves a list of tables in the specified database.
  #
  # @param db_name [String] The name of the database to retrieve tables from.
  # @return [Array<Aws::Glue::Types::Table>]
  def get_tables(db_name)
    response = @glue_client.get_tables(database_name: db_name)
    response.table_list
  rescue Aws::Glue::Errors::GlueException => e
    @logger.error("Glue could not get tables #{db_name}: \n#{e.message}")
    raise
  end
end
```

- 如需 API 詳細資訊，請參閱 AWS SDK for Ruby API 參考[GetTables](#)中的。

列出任務定義

下列程式碼範例顯示如何列出 AWS Glue 工作定義。

適用於 Ruby 的開發套件

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫](#)。

```
# The `GlueWrapper` class serves as a wrapper around the AWS Glue API, providing a
simplified interface for common operations.
# It encapsulates the functionality of the AWS SDK for Glue and provides methods for
interacting with Glue crawlers, databases, tables, jobs, and S3 resources.
# The class initializes with a Glue client and a logger, allowing it to make API
calls and log any errors or informational messages.
class GlueWrapper
  def initialize(glue_client, logger)
    @glue_client = glue_client
    @logger = logger
  end

  # Retrieves a list of jobs in AWS Glue.
  #
  # @return [Aws::Glue::Types::ListJobsResponse]
  def list_jobs
    @glue_client.list_jobs
  rescue Aws::Glue::Errors::GlueException => e
    @logger.error("Glue could not list jobs: \n#{e.message}")
    raise
  end
end
```

- 如需 API 詳細資訊，請參閱 AWS SDK for Ruby API 參考[ListJobs](#)中的。

啟動爬蟲程式

下列程式碼範例會示範如何啟動 AWS Glue 爬行者程式。

適用於 Ruby 的開發套件

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫](#)。

```
# The `GlueWrapper` class serves as a wrapper around the AWS Glue API, providing a
simplified interface for common operations.
# It encapsulates the functionality of the AWS SDK for Glue and provides methods for
interacting with Glue crawlers, databases, tables, jobs, and S3 resources.
# The class initializes with a Glue client and a logger, allowing it to make API
calls and log any errors or informational messages.
class GlueWrapper
  def initialize(glue_client, logger)
    @glue_client = glue_client
    @logger = logger
  end


  # Starts a crawler with the specified name.
  #
  # @param name [String] The name of the crawler to start.
  # @return [void]
  def start_crawler(name)
    @glue_client.start_crawler(name: name)
  rescue Aws::Glue::Errors::ServiceError => e
    @logger.error("Glue could not start crawler #{name}: \n#{e.message}")
    raise
  end
end
```

- 如需 API 詳細資訊，請參閱 AWS SDK for Ruby API 參考[StartCrawler](#)中的。

開始任務執行

下列程式碼範例會示範如何啟動 AWS Glue 工作執行。

適用於 Ruby 的開發套件

 Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫](#)。

```
# The `GlueWrapper` class serves as a wrapper around the AWS Glue API, providing a
simplified interface for common operations.
# It encapsulates the functionality of the AWS SDK for Glue and provides methods for
interacting with Glue crawlers, databases, tables, jobs, and S3 resources.
# The class initializes with a Glue client and a logger, allowing it to make API
calls and log any errors or informational messages.
class GlueWrapper
  def initialize(glue_client, logger)
    @glue_client = glue_client
    @logger = logger
  end

  # Starts a job run for the specified job.
  #
  # @param name [String] The name of the job to start the run for.
  # @param input_database [String] The name of the input database for the job.
  # @param input_table [String] The name of the input table for the job.
  # @param output_bucket_name [String] The name of the output S3 bucket for the job.
  # @return [String] The ID of the started job run.
  def start_job_run(name, input_database, input_table, output_bucket_name)
    response = @glue_client.start_job_run(
      job_name: name,
      arguments: {
        '--input_database': input_database,
        '--input_table': input_table,
        '--output_bucket_url': "s3://#{output_bucket_name}/"
      }
    )
    response.job_run_id
  rescue Aws::Glue::Errors::GlueException => e
    @logger.error("Glue could not start job run #{name}: \n#{e.message}")
    raise
  end
end
```

- 如需 API 詳細資訊，請參閱 AWS SDK for Ruby API 參考[StartJobRun](#)中的。

案例

開始使用爬蟲程式和任務

以下程式碼範例顯示做法：

- 建立網路爬取公有 Amazon S3 儲存貯體的爬蟲程式，以及產生 CSV 格式中繼資料的資料庫。
- 列出有關 AWS Glue Data Catalog。
- 建立從 S3 儲存貯體中擷取 CSV 資料的任務、轉換資料，以及將 JSON 格式的輸出載入至另一個 S3 儲存貯體。
- 列出任務執行的相關資訊、檢視已轉換的資料以及清除資源。

如需詳細資訊，請參閱 < [教學課程：開始使用 AWS Glue Studio](#) >。

適用於 Ruby 的開發套件

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫](#)。

創建一個包裝在場景中使用的 AWS Glue 函數的類。

```
# The `GlueWrapper` class serves as a wrapper around the AWS Glue API, providing a
simplified interface for common operations.
# It encapsulates the functionality of the AWS SDK for Glue and provides methods for
interacting with Glue crawlers, databases, tables, jobs, and S3 resources.
# The class initializes with a Glue client and a logger, allowing it to make API
calls and log any errors or informational messages.
class GlueWrapper
  def initialize(glue_client, logger)
    @glue_client = glue_client
    @logger = logger
  end

  # Retrieves information about a specific crawler.
  #
```

```
# @param name [String] The name of the crawler to retrieve information about.
# @return [Aws::Glue::Types::Crawler, nil] The crawler object if found, or nil if
not found.
def get_crawler(name)
  @glue_client.get_crawler(name: name)
rescue Aws::Glue::Errors::EntityNotFoundException
  @logger.info("Crawler #{name} doesn't exist.")
  false
rescue Aws::Glue::Errors::GlueException => e
  @logger.error("Glue could not get crawler #{name}: \n#{e.message}")
  raise
end

# Creates a new crawler with the specified configuration.
#
# @param name [String] The name of the crawler.
# @param role_arn [String] The ARN of the IAM role to be used by the crawler.
# @param db_name [String] The name of the database where the crawler stores its
metadata.
# @param db_prefix [String] The prefix to be added to the names of tables that the
crawler creates.
# @param s3_target [String] The S3 path that the crawler will crawl.
# @return [void]
def create_crawler(name, role_arn, db_name, db_prefix, s3_target)
  @glue_client.create_crawler(
    name: name,
    role: role_arn,
    database_name: db_name,
    targets: {
      s3_targets: [
        {
          path: s3_target
        }
      ]
    }
  )
rescue Aws::Glue::Errors::GlueException => e
  @logger.error("Glue could not create crawler: \n#{e.message}")
  raise
end

# Starts a crawler with the specified name.
#
# @param name [String] The name of the crawler to start.
```

```
# @return [void]
def start_crawler(name)
  @glue_client.start_crawler(name: name)
rescue Aws::Glue::Errors::ServiceError => e
  @logger.error("Glue could not start crawler #{name}: \n#{e.message}")
  raise
end

# Deletes a crawler with the specified name.
#
# @param name [String] The name of the crawler to delete.
# @return [void]
def delete_crawler(name)
  @glue_client.delete_crawler(name: name)
rescue Aws::Glue::Errors::ServiceError => e
  @logger.error("Glue could not delete crawler #{name}: \n#{e.message}")
  raise
end

# Retrieves information about a specific database.
#
# @param name [String] The name of the database to retrieve information about.
# @return [Aws::Glue::Types::Database, nil] The database object if found, or nil
if not found.
def get_database(name)
  response = @glue_client.get_database(name: name)
  response.database
rescue Aws::Glue::Errors::GlueException => e
  @logger.error("Glue could not get database #{name}: \n#{e.message}")
  raise
end

# Retrieves a list of tables in the specified database.
#
# @param db_name [String] The name of the database to retrieve tables from.
# @return [Array<Aws::Glue::Types::Table>]
def get_tables(db_name)
  response = @glue_client.get_tables(database_name: db_name)
  response.table_list
rescue Aws::Glue::Errors::GlueException => e
  @logger.error("Glue could not get tables #{db_name}: \n#{e.message}")
  raise
end
```

```
# Creates a new job with the specified configuration.
#
# @param name [String] The name of the job.
# @param description [String] The description of the job.
# @param role_arn [String] The ARN of the IAM role to be used by the job.
# @param script_location [String] The location of the ETL script for the job.
# @return [void]
def create_job(name, description, role_arn, script_location)
  @glue_client.create_job(
    name: name,
    description: description,
    role: role_arn,
    command: {
      name: "glueetl",
      script_location: script_location,
      python_version: "3"
    },
    glue_version: "3.0"
  )
rescue Aws::Glue::Errors::GlueException => e
  @logger.error("Glue could not create job #{name}: \n#{e.message}")
  raise
end

# Starts a job run for the specified job.
#
# @param name [String] The name of the job to start the run for.
# @param input_database [String] The name of the input database for the job.
# @param input_table [String] The name of the input table for the job.
# @param output_bucket_name [String] The name of the output S3 bucket for the job.
# @return [String] The ID of the started job run.
def start_job_run(name, input_database, input_table, output_bucket_name)
  response = @glue_client.start_job_run(
    job_name: name,
    arguments: {
      '--input_database': input_database,
      '--input_table': input_table,
      '--output_bucket_url': "s3://#{output_bucket_name}/"
    }
  )
  response.job_run_id
rescue Aws::Glue::Errors::GlueException => e
  @logger.error("Glue could not start job run #{name}: \n#{e.message}")
  raise
end
```

```
end

# Retrieves a list of jobs in AWS Glue.
#
# @return [Aws::Glue::Types::ListJobsResponse]
def list_jobs
  @glue_client.list_jobs
rescue Aws::Glue::Errors::GlueException => e
  @logger.error("Glue could not list jobs: \n#{e.message}")
  raise
end

# Retrieves a list of job runs for the specified job.
#
# @param job_name [String] The name of the job to retrieve job runs for.
# @return [Array<Aws::Glue::Types::JobRun>]
def get_job_runs(job_name)
  response = @glue_client.get_job_runs(job_name: job_name)
  response.job_runs
rescue Aws::Glue::Errors::GlueException => e
  @logger.error("Glue could not get job runs: \n#{e.message}")
end

# Retrieves data for a specific job run.
#
# @param job_name [String] The name of the job run to retrieve data for.
# @return [Glue::Types::GetJobRunResponse]
def get_job_run(job_name, run_id)
  @glue_client.get_job_run(job_name: job_name, run_id: run_id)
rescue Aws::Glue::Errors::GlueException => e
  @logger.error("Glue could not get job runs: \n#{e.message}")
end

# Deletes a job with the specified name.
#
# @param job_name [String] The name of the job to delete.
# @return [void]
def delete_job(job_name)
  @glue_client.delete_job(job_name: job_name)
rescue Aws::Glue::Errors::ServiceError => e
  @logger.error("Glue could not delete job: \n#{e.message}")
end

# Deletes a table with the specified name.
```

```
#
# @param database_name [String] The name of the catalog database in which the
table resides.
# @param table_name [String] The name of the table to be deleted.
# @return [void]
def delete_table(database_name, table_name)
  @glue_client.delete_table(database_name: database_name, name: table_name)
rescue Aws::Glue::Errors::ServiceError => e
  @logger.error("Glue could not delete job: \n#{e.message}")
end

# Removes a specified database from a Data Catalog.
#
# @param database_name [String] The name of the database to delete.
# @return [void]
def delete_database(database_name)
  @glue_client.delete_database(name: database_name)
rescue Aws::Glue::Errors::ServiceError => e
  @logger.error("Glue could not delete database: \n#{e.message}")
end

# Uploads a job script file to an S3 bucket.
#
# @param file_path [String] The local path of the job script file.
# @param bucket_resource [Aws::S3::Bucket] The S3 bucket resource to upload the
file to.
# @return [void]
def upload_job_script(file_path, bucket_resource)
  File.open(file_path) do |file|
    bucket_resource.client.put_object({
      body: file,
      bucket: bucket_resource.name,
      key: file_path
    })
  end
rescue Aws::S3::Errors::S3UploadFailedError => e
  @logger.error("S3 could not upload job script: \n#{e.message}")
  raise
end

end
```


建立可執行案例的類別。

```
class GlueCrawlerJobScenario
  def initialize(glue_client, glue_service_role, glue_bucket, logger)
    @glue_client = glue_client
    @glue_service_role = glue_service_role
    @glue_bucket = glue_bucket
    @logger = logger
  end

  def run(crawler_name, db_name, db_prefix, data_source, job_script, job_name)
    wrapper = GlueWrapper.new(@glue_client, @logger)

    new_step(1, "Create a crawler")
    puts "Checking for crawler #{crawler_name}."
    crawler = wrapper.get_crawler(crawler_name)
    if crawler == false
      puts "Creating crawler #{crawler_name}."
      wrapper.create_crawler(crawler_name, @glue_service_role.arn, db_name,
db_prefix, data_source)
      puts "Successfully created #{crawler_name}:"
      crawler = wrapper.get_crawler(crawler_name)
      puts JSON.pretty_generate(crawler).yellow
    end
    print "\nDone!\n".green

    new_step(2, "Run a crawler to output a database.")
    puts "Location of input data analyzed by crawler: #{data_source}"
    puts "Outputs: a Data Catalog database in CSV format containing metadata on
input."
    wrapper.start_crawler(crawler_name)
    puts "Starting crawler... (this typically takes a few minutes)"
    crawler_state = nil
    while crawler_state != "READY"
      custom_wait(15)
      crawler = wrapper.get_crawler(crawler_name)
      crawler_state = crawler[0]["state"]
      print "Status check: #{crawler_state}.".yellow
    end
    print "\nDone!\n".green

    new_step(3, "Query the database.")
    database = wrapper.get_database(db_name)
    puts "The crawler created database #{db_name}:"
```

```
print "#{database}".yellow
puts "\nThe database contains these tables:"
tables = wrapper.get_tables(db_name)
tables.each_with_index do |table, index|
  print "\t#{index + 1}. #{table['name']}".yellow
end
print "\nDone!\n".green

new_step(4, "Create a job definition that runs an ETL script.")
puts "Uploading Python ETL script to S3..."
wrapper.upload_job_script(job_script, @glue_bucket)
puts "Creating job definition #{job_name}:\n"
response = wrapper.create_job(job_name, "Getting started example job.",
@glue_service_role.arn, "s3://#{@glue_bucket.name}/#{job_script}")
puts JSON.pretty_generate(response).yellow
print "\nDone!\n".green

new_step(5, "Start a new job")
job_run_status = nil
job_run_id = wrapper.start_job_run(
  job_name,
  db_name,
  tables[0]["name"],
  @glue_bucket.name
)
puts "Job #{job_name} started. Let's wait for it to run."
until ["SUCCEEDED", "STOPPED", "FAILED", "TIMEOUT"].include?(job_run_status)
  custom_wait(10)
  job_run = wrapper.get_job_runs(job_name)
  job_run_status = job_run[0]["job_run_state"]
  print "Status check: #{job_name}/#{job_run_id} - #{job_run_status}.".yellow
end
print "\nDone!\n".green

new_step(6, "View results from a successful job run.")
if job_run_status == "SUCCEEDED"
  puts "Data from your job run is stored in your S3 bucket
'#{@glue_bucket.name}'. Files include:"
  begin

    # Print the key name of each object in the bucket.
    @glue_bucket.objects.each do |object_summary|
      if object_summary.key.include?("run-")
        print "#{object_summary.key}".yellow
      end
    end
  end
end
```

```

        end
    end

    # Print the first 256 bytes of a run file
    desired_sample_objects = 1
    @glue_bucket.objects.each do |object_summary|
        if object_summary.key.include?("run-")
            if desired_sample_objects > 0
                sample_object = @glue_bucket.object(object_summary.key)
                sample = sample_object.get(range: "bytes=0-255").body.read
                puts "\nSample run file contents:"
                print "#{sample}".yellow
                desired_sample_objects -= 1
            end
        end
    end

    rescue Aws::S3::Errors::ServiceError => e
        logger.error(
            "Couldn't get job run data. Here's why: %s: %s",
            e.response.error.code, e.response.error.message
        )
        raise
    end

    end
end

print "\nDone!\n".green

new_step(7, "Delete job definition and crawler.")
wrapper.delete_job(job_name)
puts "Job deleted: #{job_name}."
wrapper.delete_crawler(crawler_name)
puts "Crawler deleted: #{crawler_name}."
wrapper.delete_table(db_name, tables[0]["name"])
puts "Table deleted: #{tables[0]["name"]} in #{db_name}."
wrapper.delete_database(db_name)
puts "Database deleted: #{db_name}."
print "\nDone!\n".green

end
end

def main

    banner(".././helpers/banner.txt")
    puts
    "#####"

```

```

puts "#
           #".yellow

puts "#
           #".yellow
           EXAMPLE CODE DEMO:

puts "#
           #".yellow
           AWS Glue

puts "#
           #".yellow

puts
#####
puts ""
puts "You have launched a demo of AWS Glue using the AWS for Ruby v3 SDK. Over the
next 60 seconds, it will"
puts "do the following:"
puts "  1. Create a crawler."
puts "  2. Run a crawler to output a database."
puts "  3. Query the database."
puts "  4. Create a job definition that runs an ETL script."
puts "  5. Start a new job."
puts "  6. View results from a successful job run."
puts "  7. Delete job definition and crawler."
puts ""

confirm_begin
billing
security
puts "\e[H\e[2J"

# Set input file names
job_script_filepath = "job_script.py"
resource_names = YAML.load_file("resource_names.yaml")

# Instantiate existing IAM role.
iam = Aws::IAM::Resource.new(region: "us-east-1")
iam_role_name = resource_names["glue_service_role"]
iam_role = iam.role(iam_role_name)

# Instantiate existing S3 bucket.
s3 = Aws::S3::Resource.new(region: "us-east-1")
s3_bucket_name = resource_names["glue_bucket"]
s3_bucket = s3.bucket(s3_bucket_name)

scenario = GlueCrawlerJobScenario.new(
  Aws::Glue::Client.new(region: "us-east-1"),

```

```

    iam_role,
    s3_bucket,
    @logger
  )

  random_int = rand(10 ** 4)
  scenario.run(
    "doc-example-crawler-#{random_int}",
    "doc-example-database-#{random_int}",
    "doc-example-#{random_int}-",
    "s3://crawler-public-us-east-1/flight/2016/csv",
    job_script_filepath,
    "doc-example-job-#{random_int}"
  )

  puts "-" * 88
  puts "You have reached the end of this tour of AWS Glue."
  puts "To destroy CDK-created resources, run:\n          cdk destroy"
  puts "-" * 88

end

```

建立 ETL 指令碼，AWS Glue 以便在工作執行期間擷取、轉換和載入資料。

```

import sys
from awsglue.transforms import *
from awsglue.utils import getResolvedOptions
from pyspark.context import SparkContext
from awsglue.context import GlueContext
from awsglue.job import Job

"""
These custom arguments must be passed as Arguments to the StartJobRun request.
  --input_database      The name of a metadata database that is contained in your
                        AWS Glue Data Catalog and that contains tables that
describe
                        the data to be processed.
  --input_table        The name of a table in the database that describes the data
to
                        be processed.
  --output_bucket_url  An S3 bucket that receives the transformed output data.
"""

```

```
args = getResolvedOptions(
  sys.argv, ["JOB_NAME", "input_database", "input_table", "output_bucket_url"]
)
sc = SparkContext()
glueContext = GlueContext(sc)
spark = glueContext.spark_session
job = Job(glueContext)
job.init(args["JOB_NAME"], args)

# Script generated for node S3 Flight Data.
S3FlightData_node1 = glueContext.create_dynamic_frame.from_catalog(
  database=args["input_database"],
  table_name=args["input_table"],
  transformation_ctx="S3FlightData_node1",
)

# This mapping performs two main functions:
# 1. It simplifies the output by removing most of the fields from the data.
# 2. It renames some fields. For example, `fl_date` is renamed to `flight_date`.
ApplyMapping_node2 = ApplyMapping.apply(
  frame=S3FlightData_node1,
  mappings=[
    ("year", "long", "year", "long"),
    ("month", "long", "month", "tinyint"),
    ("day_of_month", "long", "day", "tinyint"),
    ("fl_date", "string", "flight_date", "string"),
    ("carrier", "string", "carrier", "string"),
    ("fl_num", "long", "flight_num", "long"),
    ("origin_city_name", "string", "origin_city_name", "string"),
    ("origin_state_abr", "string", "origin_state_abr", "string"),
    ("dest_city_name", "string", "dest_city_name", "string"),
    ("dest_state_abr", "string", "dest_state_abr", "string"),
    ("dep_time", "long", "departure_time", "long"),
    ("wheels_off", "long", "wheels_off", "long"),
    ("wheels_on", "long", "wheels_on", "long"),
    ("arr_time", "long", "arrival_time", "long"),
    ("mon", "string", "mon", "string"),
  ],
  transformation_ctx="ApplyMapping_node2",
)

# Script generated for node Revised Flight Data.
RevisedFlightData_node3 = glueContext.write_dynamic_frame.from_options(
  frame=ApplyMapping_node2,
```

```
connection_type="s3",
format="json",
connection_options={"path": args["output_bucket_url"], "partitionKeys": []},
transformation_ctx="RevisedFlightData_node3",
)

job.commit()
```

- 如需 API 詳細資訊，請參閱《AWS SDK for Ruby API 參考》中的下列主題。
 - [CreateCrawler](#)
 - [CreateJob](#)
 - [DeleteCrawler](#)
 - [DeleteDatabase](#)
 - [DeleteJob](#)
 - [DeleteTable](#)
 - [GetCrawler](#)
 - [GetDatabase](#)
 - [GetDatabases](#)
 - [GetJob](#)
 - [GetJobRun](#)
 - [GetJobRuns](#)
 - [GetTables](#)
 - [ListJobs](#)
 - [StartCrawler](#)
 - [StartJobRun](#)

使用 SDK for Ruby IAM 範例

下列程式碼範例說明如何使用 IAM 來執行動作和實作常見案例。AWS SDK for Ruby

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境和跨服務範例中查看內容中的動作。

案例是向您展示如何呼叫相同服務中的多個函數來完成特定任務的程式碼範例。

每個範例都包含一個連結 GitHub，您可以在其中找到如何在內容中設定和執程式碼的指示。

主題

- [動作](#)
- [案例](#)

動作

將政策連接至角色

下列程式碼範例示範如何將 IAM 政策連接至角色。

適用於 Ruby 的開發套件

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執程式碼範例儲存庫](#)。

此範例模組會列出、建立、附加和解除連結角色原則。

```
# Manages policies in AWS Identity and Access Management (IAM)
class RolePolicyManager
  # Initialize with an AWS IAM client
  #
  # @param iam_client [Aws::IAM::Client] An initialized IAM client
  def initialize(iam_client, logger: Logger.new($stdout))
    @iam_client = iam_client
    @logger = logger
    @logger.progname = "PolicyManager"
  end

  # Creates a policy
  #
  # @param policy_name [String] The name of the policy
  # @param policy_document [Hash] The policy document
  # @return [String] The policy ARN if successful, otherwise nil
  def create_policy(policy_name, policy_document)
    response = @iam_client.create_policy(
      policy_name: policy_name,
      policy_document: policy_document.to_json
    )
  end
end
```



```

    )
    response.policy.arn
  rescue Aws::IAM::Errors::ServiceError => e
    @logger.error("Error creating policy: #{e.message}")
    nil
  end

  # Fetches an IAM policy by its ARN
  # @param policy_arn [String] the ARN of the IAM policy to retrieve
  # @return [Aws::IAM::Types::GetPolicyResponse] the policy object if found
  def get_policy(policy_arn)
    response = @iam_client.get_policy(policy_arn: policy_arn)
    policy = response.policy
    @logger.info("Got policy '#{policy.policy_name}'. Its ID is:
    #{policy.policy_id}.")
    policy
  rescue Aws::IAM::Errors::NoSuchEntity
    @logger.error("Couldn't get policy '#{policy_arn}'. The policy does not exist.")
    raise
  rescue Aws::IAM::Errors::ServiceError => e
    @logger.error("Couldn't get policy '#{policy_arn}'. Here's why: #{e.code}:
    #{e.message}")
    raise
  end

  # Attaches a policy to a role
  #
  # @param role_name [String] The name of the role
  # @param policy_arn [String] The policy ARN
  # @return [Boolean] true if successful, false otherwise
  def attach_policy_to_role(role_name, policy_arn)
    @iam_client.attach_role_policy(
      role_name: role_name,
      policy_arn: policy_arn
    )
    true
  rescue Aws::IAM::Errors::ServiceError => e
    @logger.error("Error attaching policy to role: #{e.message}")
    false
  end

  # Lists policy ARNs attached to a role
  #
  # @param role_name [String] The name of the role

```

```
# @return [Array<String>] List of policy ARNs
def list_attached_policy_arns(role_name)
  response = @iam_client.list_attached_role_policies(role_name: role_name)
  response.attached_policies.map(&:policy_arn)
rescue Aws::IAM::Errors::ServiceError => e
  @logger.error("Error listing policies attached to role: #{e.message}")
  []
end

# Detaches a policy from a role
#
# @param role_name [String] The name of the role
# @param policy_arn [String] The policy ARN
# @return [Boolean] true if successful, false otherwise
def detach_policy_from_role(role_name, policy_arn)
  @iam_client.detach_role_policy(
    role_name: role_name,
    policy_arn: policy_arn
  )
  true
rescue Aws::IAM::Errors::ServiceError => e
  @logger.error("Error detaching policy from role: #{e.message}")
  false
end
end
```

- 如需 API 詳細資訊，請參閱 AWS SDK for Ruby API 參考[AttachRolePolicy](#)中的。

將政策連接至使用者

下列程式碼範例顯示如何將 IAM 政策附加至使用者。

Warning

為避免安全風險，在開發專用軟體或使用真實資料時，請勿使用 IAM 使用者進行身分驗證。相反地，搭配使用聯合功能和身分提供者，例如 [AWS IAM Identity Center](#)。

適用於 Ruby 的開發套件

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫](#)。

```
# Attaches a policy to a user
#
# @param user_name [String] The name of the user
# @param policy_arn [String] The Amazon Resource Name (ARN) of the policy
# @return [Boolean] true if successful, false otherwise
def attach_policy_to_user(user_name, policy_arn)
  @iam_client.attach_user_policy(
    user_name: user_name,
    policy_arn: policy_arn
  )
  true
rescue Aws::IAM::Errors::ServiceError => e
  @logger.error("Error attaching policy to user: #{e.message}")
  false
end
```

- 如需 API 詳細資訊，請參閱 AWS SDK for Ruby API 參考[AttachUserPolicy](#)中的。

建立政策

下列程式碼範例示範如何建立 IAM 政策。

適用於 Ruby 的開發套件

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫](#)。

此範例模組會列出、建立、附加和解除連結角色原則。

```
# Manages policies in AWS Identity and Access Management (IAM)
class RolePolicyManager
```

```
# Initialize with an AWS IAM client
#
# @param iam_client [Aws::IAM::Client] An initialized IAM client
def initialize(iam_client, logger: Logger.new($stdout))
  @iam_client = iam_client
  @logger = logger
  @logger.progname = "PolicyManager"
end

# Creates a policy
#
# @param policy_name [String] The name of the policy
# @param policy_document [Hash] The policy document
# @return [String] The policy ARN if successful, otherwise nil
def create_policy(policy_name, policy_document)
  response = @iam_client.create_policy(
    policy_name: policy_name,
    policy_document: policy_document.to_json
  )
  response.policy.arn
rescue Aws::IAM::Errors::ServiceError => e
  @logger.error("Error creating policy: #{e.message}")
  nil
end

# Fetches an IAM policy by its ARN
# @param policy_arn [String] the ARN of the IAM policy to retrieve
# @return [Aws::IAM::Types::GetPolicyResponse] the policy object if found
def get_policy(policy_arn)
  response = @iam_client.get_policy(policy_arn: policy_arn)
  policy = response.policy
  @logger.info("Got policy '#{policy.policy_name}'. Its ID is:
#{policy.policy_id}.")
  policy
rescue Aws::IAM::Errors::NoSuchEntity
  @logger.error("Couldn't get policy '#{policy_arn}'. The policy does not exist.")
  raise
rescue Aws::IAM::Errors::ServiceError => e
  @logger.error("Couldn't get policy '#{policy_arn}'. Here's why: #{e.code}:
#{e.message}")
  raise
end

# Attaches a policy to a role
```

```
#
# @param role_name [String] The name of the role
# @param policy_arn [String] The policy ARN
# @return [Boolean] true if successful, false otherwise
def attach_policy_to_role(role_name, policy_arn)
  @iam_client.attach_role_policy(
    role_name: role_name,
    policy_arn: policy_arn
  )
  true
rescue Aws::IAM::Errors::ServiceError => e
  @logger.error("Error attaching policy to role: #{e.message}")
  false
end

# Lists policy ARNs attached to a role
#
# @param role_name [String] The name of the role
# @return [Array<String>] List of policy ARNs
def list_attached_policy_arns(role_name)
  response = @iam_client.list_attached_role_policies(role_name: role_name)
  response.attached_policies.map(&:policy_arn)
rescue Aws::IAM::Errors::ServiceError => e
  @logger.error("Error listing policies attached to role: #{e.message}")
  []
end

# Detaches a policy from a role
#
# @param role_name [String] The name of the role
# @param policy_arn [String] The policy ARN
# @return [Boolean] true if successful, false otherwise
def detach_policy_from_role(role_name, policy_arn)
  @iam_client.detach_role_policy(
    role_name: role_name,
    policy_arn: policy_arn
  )
  true
rescue Aws::IAM::Errors::ServiceError => e
  @logger.error("Error detaching policy from role: #{e.message}")
  false
end
end
```

- 如需 API 詳細資訊，請參閱 AWS SDK for Ruby API 參考[CreatePolicy](#)中的。

建立角色

下列程式碼範例示範如何建立 IAM 角色。

適用於 Ruby 的開發套件

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫](#)。

```
# Creates a role and attaches policies to it.
#
# @param role_name [String] The name of the role.
# @param assume_role_policy_document [Hash] The trust relationship policy
document.
# @param policy_arns [Array<String>] The ARNs of the policies to attach.
# @return [String, nil] The ARN of the new role if successful, or nil if an error
occurred.
def create_role(role_name, assume_role_policy_document, policy_arns)
  response = @iam_client.create_role(
    role_name: role_name,
    assume_role_policy_document: assume_role_policy_document.to_json
  )
  role_arn = response.role.arn

  policy_arns.each do |policy_arn|
    @iam_client.attach_role_policy(
      role_name: role_name,
      policy_arn: policy_arn
    )
  end

  role_arn
rescue Aws::IAM::Errors::ServiceError => e
  @logger.error("Error creating role: #{e.message}")
  nil
end
```

- 如需 API 詳細資訊，請參閱 AWS SDK for Ruby API 參考[CreateRole](#)中的。

建立服務連結角色

下列程式碼範例顯示如何建立 IAM 服務連結角色。

適用於 Ruby 的開發套件

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫](#)。

```
# Creates a service-linked role
#
# @param service_name [String] The service name to create the role for.
# @param description [String] The description of the service-linked role.
# @param suffix [String] Suffix for customizing role name.
# @return [String] The name of the created role
def create_service_linked_role(service_name, description, suffix)
  response = @iam_client.create_service_linked_role(
    aws_service_name: service_name, description: description, custom_suffix:
suffix,)
  role_name = response.role.role_name
  @logger.info("Created service-linked role #{role_name}.")
  role_name
rescue Aws::Errors::ServiceError => e
  @logger.error("Couldn't create service-linked role for #{service_name}. Here's
why:")
  @logger.error("\t#{e.code}: #{e.message}")
  raise
end
```

- 如需 API 詳細資訊，請參閱 AWS SDK for Ruby API 參考[CreateServiceLinkedRole](#)中的。

建立使用者

下列程式碼範例示範如何建立 IAM 使用者。

⚠ Warning

為避免安全風險，在開發專用軟體或使用真實資料時，請勿使用 IAM 使用者進行身分驗證。相反地，搭配使用聯合功能和身分提供者，例如 [AWS IAM Identity Center](#)。

適用於 Ruby 的開發套件

i Note

還有更多關於 GitHub。尋找完整範例，並了解如何在 [AWS 設定和執行程式碼範例儲存庫](#)。

```
# Creates a user and their login profile
#
# @param user_name [String] The name of the user
# @param initial_password [String] The initial password for the user
# @return [String, nil] The ID of the user if created, or nil if an error occurred
def create_user(user_name, initial_password)
  response = @iam_client.create_user(user_name: user_name)
  @iam_client.wait_until(:user_exists, user_name: user_name)
  @iam_client.create_login_profile(
    user_name: user_name,
    password: initial_password,
    password_reset_required: true
  )
  @logger.info("User '#{user_name}' created successfully.")
  response.user.user_id
rescue Aws::IAM::Errors::EntityAlreadyExists
  @logger.error("Error creating user '#{user_name}': user already exists.")
  nil
rescue Aws::IAM::Errors::ServiceError => e
  @logger.error("Error creating user '#{user_name}': #{e.message}")
  nil
end
```

- 如需 API 詳細資訊，請參閱 AWS SDK for Ruby API 參考 [CreateUser](#) 中的。

建立存取金鑰

下列程式碼範例示範如何建立 IAM 存取金鑰。

Warning

為避免安全風險，在開發專用軟體或使用真實資料時，請勿使用 IAM 使用者進行身分驗證。相反地，搭配使用聯合功能和身分提供者，例如 [AWS IAM Identity Center](#)。

適用於 Ruby 的開發套件

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在 [AWS 設定和執行程式碼範例儲存庫](#)。

此範例模組會列出、建立、停用及刪除存取金鑰。

```
# Manages access keys for IAM users
class AccessKeyManager
  def initialize(iam_client, logger: Logger.new($stdout))
    @iam_client = iam_client
    @logger = logger
    @logger.progname = "AccessKeyManager"
  end

  # Lists access keys for a user
  #
  # @param user_name [String] The name of the user.
  def list_access_keys(user_name)
    response = @iam_client.list_access_keys(user_name: user_name)
    if response.access_key_metadata.empty?
      @logger.info("No access keys found for user '#{user_name}'.")
    else
      response.access_key_metadata.map(&:access_key_id)
    end
  rescue Aws::IAM::Errors::NoSuchEntity => e
    @logger.error("Error listing access keys: cannot find user '#{user_name}'.")
    []
  rescue StandardError => e
    @logger.error("Error listing access keys: #{e.message}")
  end
end
```

```
    []
  end

  # Creates an access key for a user
  #
  # @param user_name [String] The name of the user.
  # @return [Boolean]
  def create_access_key(user_name)
    response = @iam_client.create_access_key(user_name: user_name)
    access_key = response.access_key
    @logger.info("Access key created for user '#{user_name}':
#{access_key.access_key_id}")
    access_key
  rescue Aws::IAM::Errors::LimitExceeded => e
    @logger.error("Error creating access key: limit exceeded. Cannot create more.")
    nil
  rescue StandardError => e
    @logger.error("Error creating access key: #{e.message}")
    nil
  end

  # Deactivates an access key
  #
  # @param user_name [String] The name of the user.
  # @param access_key_id [String] The ID for the access key.
  # @return [Boolean]
  def deactivate_access_key(user_name, access_key_id)
    @iam_client.update_access_key(
      user_name: user_name,
      access_key_id: access_key_id,
      status: "Inactive"
    )
    true
  rescue StandardError => e
    @logger.error("Error deactivating access key: #{e.message}")
    false
  end

  # Deletes an access key
  #
  # @param user_name [String] The name of the user.
  # @param access_key_id [String] The ID for the access key.
  # @return [Boolean]
  def delete_access_key(user_name, access_key_id)
```

```

    @iam_client.delete_access_key(
      user_name: user_name,
      access_key_id: access_key_id
    )
    true
  rescue StandardError => e
    @logger.error("Error deleting access key: #{e.message}")
    false
  end
end
end

```

- 如需 API 詳細資訊，請參閱 AWS SDK for Ruby API 參考 [CreateAccessKey](#) 中的。

建立帳戶別名

下列程式碼範例顯示如何為 IAM 帳戶建立別名。

適用於 Ruby 的開發套件

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在 [AWS 設定和執行程式碼範例儲存庫](#)。

列出、建立及刪除帳戶別名。

```

class IAMAliasManager
  # Initializes the IAM client and logger
  #
  # @param iam_client [Aws::IAM::Client] An initialized IAM client.
  def initialize(iam_client, logger: Logger.new($stdout))
    @iam_client = iam_client
    @logger = logger
  end

  # Lists available AWS account aliases.
  def list_aliases
    response = @iam_client.list_account_aliases

    if response.account_aliases.count.positive?
      @logger.info("Account aliases are:")
    end
  end
end

```

```
        response.account_aliases.each { |account_alias| @logger.info("
#{account_alias}") }
      else
        @logger.info("No account aliases found.")
      end
    rescue Aws::IAM::Errors::ServiceError => e
      @logger.error("Error listing account aliases: #{e.message}")
    end

    # Creates an AWS account alias.
    #
    # @param account_alias [String] The name of the account alias to create.
    # @return [Boolean] true if the account alias was created; otherwise, false.
    def create_account_alias(account_alias)
      @iam_client.create_account_alias(account_alias: account_alias)
      true
    rescue Aws::IAM::Errors::ServiceError => e
      @logger.error("Error creating account alias: #{e.message}")
      false
    end

    # Deletes an AWS account alias.
    #
    # @param account_alias [String] The name of the account alias to delete.
    # @return [Boolean] true if the account alias was deleted; otherwise, false.
    def delete_account_alias(account_alias)
      @iam_client.delete_account_alias(account_alias: account_alias)
      true
    rescue Aws::IAM::Errors::ServiceError => e
      @logger.error("Error deleting account alias: #{e.message}")
      false
    end
  end
end
```

- 如需 API 詳細資訊，請參閱 AWS SDK for Ruby API 參考[CreateAccountAlias](#)中的。

為使用者建立內嵌政策

以下程式碼範例示範如何為使用者建立內嵌 IAM 政策。

⚠ Warning

為避免安全風險，在開發專用軟體或使用真實資料時，請勿使用 IAM 使用者進行身分驗證。相反地，搭配使用聯合功能和身分提供者，例如 [AWS IAM Identity Center](#)。

適用於 Ruby 的開發套件

i Note

還有更多關於 GitHub。尋找完整範例，並了解如何在 [AWS 設定和執行程式碼範例儲存庫](#)。

```
# Creates an inline policy for a specified user.
# @param username [String] The name of the IAM user.
# @param policy_name [String] The name of the policy to create.
# @param policy_document [String] The JSON policy document.
# @return [Boolean]
def create_user_policy(username, policy_name, policy_document)
  @iam_client.put_user_policy({
    user_name: username,
    policy_name: policy_name,
    policy_document: policy_document
  })
  @logger.info("Policy #{policy_name} created for user #{username}.")
  true
rescue Aws::IAM::Errors::ServiceError => e
  @logger.error("Couldn't create policy #{policy_name} for user #{username}.
Here's why:")
  @logger.error("\t#{e.code}: #{e.message}")
  false
end
```

- 如需 API 詳細資訊，請參閱 AWS SDK for Ruby API 參考 [PutUserPolicy](#) 中的。

刪除角色

下列程式碼範例示範如何刪除 IAM 角色。

適用於 Ruby 的開發套件

 Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫](#)。

```
# Deletes a role and its attached policies.
#
# @param role_name [String] The name of the role to delete.
def delete_role(role_name)
  begin
    # Detach and delete attached policies
    @iam_client.list_attached_role_policies(role_name: role_name).each do |
response|
      response.attached_policies.each do |policy|
        @iam_client.detach_role_policy({
          role_name: role_name,
          policy_arn: policy.policy_arn
        })
        # Check if the policy is a customer managed policy (not AWS managed)
        unless policy.policy_arn.include?("aws:policy/")
          @iam_client.delete_policy({ policy_arn: policy.policy_arn })
          @logger.info("Deleted customer managed policy #{policy.policy_name}.")
        end
      end
    end
  end

  # Delete the role
  @iam_client.delete_role({ role_name: role_name })
  @logger.info("Deleted role #{role_name}.")
  rescue Aws::IAM::Errors::ServiceError => e
    @logger.error("Couldn't detach policies and delete role #{role_name}. Here's
why:")
    @logger.error("\t#{e.code}: #{e.message}")
    raise
  end
end
```

- 如需 API 詳細資訊，請參閱 AWS SDK for Ruby API 參考[DeleteRole](#)中的。

刪除伺服器憑證

下列程式碼範例示範如何刪除 IAM 伺服器憑證。

適用於 Ruby 的開發套件

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫](#)。

列出、更新和刪除伺服器憑證。

```
class ServerCertificateManager
  def initialize(iam_client, logger: Logger.new($stdout))
    @iam_client = iam_client
    @logger = logger
    @logger.progname = "ServerCertificateManager"
  end

  # Creates a new server certificate.
  # @param name [String] the name of the server certificate
  # @param certificate_body [String] the contents of the certificate
  # @param private_key [String] the private key contents
  # @return [Boolean] returns true if the certificate was successfully created
  def create_server_certificate(name, certificate_body, private_key)
    @iam_client.upload_server_certificate({
      server_certificate_name: name,
      certificate_body: certificate_body,
      private_key: private_key,
    })

    true
  rescue Aws::IAM::Errors::ServiceError => e
    puts "Failed to create server certificate: #{e.message}"
    false
  end

  # Lists available server certificate names.
  def list_server_certificate_names
    response = @iam_client.list_server_certificates

    if response.server_certificate_metadata_list.empty?
      @logger.info("No server certificates found.")
    end
  end
end
```

```

    return
  end

  response.server_certificate_metadata_list.each do |certificate_metadata|
    @logger.info("Certificate Name:
#{certificate_metadata.server_certificate_name}")
    end
  rescue Aws::IAM::Errors::ServiceError => e
    @logger.error("Error listing server certificates: #{e.message}")
  end

  # Updates the name of a server certificate.
  def update_server_certificate_name(current_name, new_name)
    @iam_client.update_server_certificate(
      server_certificate_name: current_name,
      new_server_certificate_name: new_name
    )
    @logger.info("Server certificate name updated from '#{current_name}' to
'#{new_name}'.")
    true
  rescue Aws::IAM::Errors::ServiceError => e
    @logger.error("Error updating server certificate name: #{e.message}")
    false
  end

  # Deletes a server certificate.
  def delete_server_certificate(name)
    @iam_client.delete_server_certificate(server_certificate_name: name)
    @logger.info("Server certificate '#{name}' deleted.")
    true
  rescue Aws::IAM::Errors::ServiceError => e
    @logger.error("Error deleting server certificate: #{e.message}")
    false
  end
end
end

```

- 如需 API 詳細資訊，請參閱 AWS SDK for Ruby API 參考[DeleteServerCertificate](#)中的。

刪除服務連結角色

下列程式碼範例顯示如何刪除 IAM 服務連結角色。

適用於 Ruby 的開發套件

 Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫](#)。

```
# Deletes a service-linked role.
#
# @param role_name [String] The name of the role to delete.
def delete_service_linked_role(role_name)
  response = @iam_client.delete_service_linked_role(role_name: role_name)
  task_id = response.deletion_task_id
  check_deletion_status(role_name, task_id)
rescue Aws::Errors::ServiceError => e
  handle_deletion_error(e, role_name)
end

private

# Checks the deletion status of a service-linked role
#
# @param role_name [String] The name of the role being deleted
# @param task_id [String] The task ID for the deletion process
def check_deletion_status(role_name, task_id)
  loop do
    response = @iam_client.get_service_linked_role_deletion_status(
      deletion_task_id: task_id)
    status = response.status
    @logger.info("Deletion of #{role_name} #{status}.")
    break if %w[SUCCEEDED FAILED].include?(status)
    sleep(3)
  end
end

# Handles deletion error
#
# @param e [Aws::Errors::ServiceError] The error encountered during deletion
# @param role_name [String] The name of the role attempted to delete
def handle_deletion_error(e, role_name)
  unless e.code == "NoSuchEntity"
    @logger.error("Couldn't delete #{role_name}. Here's why:")
    @logger.error("\t#{e.code}: #{e.message}")
  end
end
```

```
    raise
  end
end
```

- 如需 API 詳細資訊，請參閱 AWS SDK for Ruby API 參考 [DeleteServiceLinkedRole](#) 中的。

刪除使用者

下列程式碼範例示範如何刪除 IAM 使用者。

Warning

為避免安全風險，在開發專用軟體或使用真實資料時，請勿使用 IAM 使用者進行身分驗證。相反地，搭配使用聯合功能和身分提供者，例如 [AWS IAM Identity Center](#)。

適用於 Ruby 的開發套件

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在 [AWS 設定和執行程式碼範例儲存庫](#)。

```
# Deletes a user and their associated resources
#
# @param user_name [String] The name of the user to delete
def delete_user(user_name)
  user = @iam_client.list_access_keys(user_name: user_name).access_key_metadata
  user.each do |key|
    @iam_client.delete_access_key({ access_key_id: key.access_key_id, user_name:
user_name })
    @logger.info("Deleted access key #{key.access_key_id} for user
'#{user_name}'.")
  end

  @iam_client.delete_user(user_name: user_name)
  @logger.info("Deleted user '#{user_name}'.")
rescue Aws::IAM::Errors::ServiceError => e
  @logger.error("Error deleting user '#{user_name}': #{e.message}")
end
```

- 如需 API 詳細資訊，請參閱 AWS SDK for Ruby API 參考 [DeleteUser](#) 中的。

刪除存取金鑰

下列程式碼範例示範如何刪除 IAM 存取金鑰。

Warning

為避免安全風險，在開發專用軟體或使用真實資料時，請勿使用 IAM 使用者進行身分驗證。相反地，搭配使用聯合功能和身分提供者，例如 [AWS IAM Identity Center](#)。

適用於 Ruby 的開發套件

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在 [AWS 設定和執行程式碼範例儲存庫](#)。

此範例模組會列出、建立、停用及刪除存取金鑰。

```
# Manages access keys for IAM users
class AccessKeyManager
  def initialize(iam_client, logger: Logger.new($stdout))
    @iam_client = iam_client
    @logger = logger
    @logger.progname = "AccessKeyManager"
  end

  # Lists access keys for a user
  #
  # @param user_name [String] The name of the user.
  def list_access_keys(user_name)
    response = @iam_client.list_access_keys(user_name: user_name)
    if response.access_key_metadata.empty?
      @logger.info("No access keys found for user '#{user_name}'.")
    else
      response.access_key_metadata.map(&:access_key_id)
    end
  end
end
```

```
rescue Aws::IAM::Errors::NoSuchEntity => e
  @logger.error("Error listing access keys: cannot find user '#{user_name}'.")
  []
rescue StandardError => e
  @logger.error("Error listing access keys: #{e.message}")
  []
end

# Creates an access key for a user
#
# @param user_name [String] The name of the user.
# @return [Boolean]
def create_access_key(user_name)
  response = @iam_client.create_access_key(user_name: user_name)
  access_key = response.access_key
  @logger.info("Access key created for user '#{user_name}':
#{access_key.access_key_id}")
  access_key
rescue Aws::IAM::Errors::LimitExceeded => e
  @logger.error("Error creating access key: limit exceeded. Cannot create more.")
  nil
rescue StandardError => e
  @logger.error("Error creating access key: #{e.message}")
  nil
end

# Deactivates an access key
#
# @param user_name [String] The name of the user.
# @param access_key_id [String] The ID for the access key.
# @return [Boolean]
def deactivate_access_key(user_name, access_key_id)
  @iam_client.update_access_key(
    user_name: user_name,
    access_key_id: access_key_id,
    status: "Inactive"
  )
  true
rescue StandardError => e
  @logger.error("Error deactivating access key: #{e.message}")
  false
end

# Deletes an access key
```

```
#
# @param user_name [String] The name of the user.
# @param access_key_id [String] The ID for the access key.
# @return [Boolean]
def delete_access_key(user_name, access_key_id)
  @iam_client.delete_access_key(
    user_name: user_name,
    access_key_id: access_key_id
  )
  true
rescue StandardError => e
  @logger.error("Error deleting access key: #{e.message}")
  false
end
end
```

- 如需 API 詳細資訊，請參閱 AWS SDK for Ruby API 參考 [DeleteAccessKey](#) 中的。

刪除帳戶別名

下列程式碼範例顯示如何刪除 IAM 帳戶別名。

適用於 Ruby 的開發套件

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在 [AWS 設定和執行程式碼範例儲存庫](#)。

列出、建立及刪除帳戶別名。

```
class IAMAliasManager
  # Initializes the IAM client and logger
  #
  # @param iam_client [Aws::IAM::Client] An initialized IAM client.
  def initialize(iam_client, logger: Logger.new($stdout))
    @iam_client = iam_client
    @logger = logger
  end

  # Lists available AWS account aliases.
```

```
def list_aliases
  response = @iam_client.list_account_aliases

  if response.account_aliases.count.positive?
    @logger.info("Account aliases are:")
    response.account_aliases.each { |account_alias| @logger.info("#{account_alias}") }
  else
    @logger.info("No account aliases found.")
  end
rescue Aws::IAM::Errors::ServiceError => e
  @logger.error("Error listing account aliases: #{e.message}")
end

# Creates an AWS account alias.
#
# @param account_alias [String] The name of the account alias to create.
# @return [Boolean] true if the account alias was created; otherwise, false.
def create_account_alias(account_alias)
  @iam_client.create_account_alias(account_alias: account_alias)
  true
rescue Aws::IAM::Errors::ServiceError => e
  @logger.error("Error creating account alias: #{e.message}")
  false
end

# Deletes an AWS account alias.
#
# @param account_alias [String] The name of the account alias to delete.
# @return [Boolean] true if the account alias was deleted; otherwise, false.
def delete_account_alias(account_alias)
  @iam_client.delete_account_alias(account_alias: account_alias)
  true
rescue Aws::IAM::Errors::ServiceError => e
  @logger.error("Error deleting account alias: #{e.message}")
  false
end
end
```

- 如需 API 詳細資訊，請參閱 AWS SDK for Ruby API 參考 [DeleteAccountAlias](#) 中的。

從使用者中刪除內嵌政策

下列程式碼範例顯示如何從使用者刪除內嵌 IAM 政策。

Warning

為避免安全風險，在開發專用軟體或使用真實資料時，請勿使用 IAM 使用者進行身分驗證。相反地，搭配使用聯合功能和身分提供者，例如 [AWS IAM Identity Center](#)。

適用於 Ruby 的開發套件

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在 [AWS 設定和執行程式碼範例儲存庫](#)。

```
# Deletes a user and their associated resources
#
# @param user_name [String] The name of the user to delete
def delete_user(user_name)
  user = @iam_client.list_access_keys(user_name: user_name).access_key_metadata
  user.each do |key|
    @iam_client.delete_access_key({ access_key_id: key.access_key_id, user_name:
user_name })
    @logger.info("Deleted access key #{key.access_key_id} for user
'#{user_name}'.")
  end

  @iam_client.delete_user(user_name: user_name)
  @logger.info("Deleted user '#{user_name}'.")
rescue Aws::IAM::Errors::ServiceError => e
  @logger.error("Error deleting user '#{user_name}': #{e.message}")
end
```

- 如需 API 詳細資訊，請參閱 AWS SDK for Ruby API 參考 [DeleteUserPolicy](#) 中的。

將政策與角色分離

下列程式碼範例示範如何將 IAM 政策與角色分離。

適用於 Ruby 的開發套件

 Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫](#)。

此範例模組會列出、建立、附加和解除連結角色原則。

```
# Manages policies in AWS Identity and Access Management (IAM)
class RolePolicyManager
  # Initialize with an AWS IAM client
  #
  # @param iam_client [Aws::IAM::Client] An initialized IAM client
  def initialize(iam_client, logger: Logger.new($stdout))
    @iam_client = iam_client
    @logger = logger
    @logger.progname = "PolicyManager"
  end

  # Creates a policy
  #
  # @param policy_name [String] The name of the policy
  # @param policy_document [Hash] The policy document
  # @return [String] The policy ARN if successful, otherwise nil
  def create_policy(policy_name, policy_document)
    response = @iam_client.create_policy(
      policy_name: policy_name,
      policy_document: policy_document.to_json
    )
    response.policy.arn
  rescue Aws::IAM::Errors::ServiceError => e
    @logger.error("Error creating policy: #{e.message}")
    nil
  end

  # Fetches an IAM policy by its ARN
  # @param policy_arn [String] the ARN of the IAM policy to retrieve
  # @return [Aws::IAM::Types::GetPolicyResponse] the policy object if found
  def get_policy(policy_arn)
    response = @iam_client.get_policy(policy_arn: policy_arn)
    policy = response.policy
  end
end
```



```
@logger.info("Got policy '#{policy.policy_name}'. Its ID is:
#{policy.policy_id}.")
  policy
rescue Aws::IAM::Errors::NoSuchEntity
  @logger.error("Couldn't get policy '#{policy_arn}'. The policy does not exist.")
  raise
rescue Aws::IAM::Errors::ServiceError => e
  @logger.error("Couldn't get policy '#{policy_arn}'. Here's why: #{e.code}:
#{e.message}")
  raise
end

# Attaches a policy to a role
#
# @param role_name [String] The name of the role
# @param policy_arn [String] The policy ARN
# @return [Boolean] true if successful, false otherwise
def attach_policy_to_role(role_name, policy_arn)
  @iam_client.attach_role_policy(
    role_name: role_name,
    policy_arn: policy_arn
  )
  true
rescue Aws::IAM::Errors::ServiceError => e
  @logger.error("Error attaching policy to role: #{e.message}")
  false
end

# Lists policy ARNs attached to a role
#
# @param role_name [String] The name of the role
# @return [Array<String>] List of policy ARNs
def list_attached_policy_arns(role_name)
  response = @iam_client.list_attached_role_policies(role_name: role_name)
  response.attached_policies.map(&:policy_arn)
rescue Aws::IAM::Errors::ServiceError => e
  @logger.error("Error listing policies attached to role: #{e.message}")
  []
end

# Detaches a policy from a role
#
# @param role_name [String] The name of the role
# @param policy_arn [String] The policy ARN
```

```
# @return [Boolean] true if successful, false otherwise
def detach_policy_from_role(role_name, policy_arn)
  @iam_client.detach_role_policy(
    role_name: role_name,
    policy_arn: policy_arn
  )
  true
rescue Aws::IAM::Errors::ServiceError => e
  @logger.error("Error detaching policy from role: #{e.message}")
  false
end
end
```

- 如需 API 詳細資訊，請參閱 AWS SDK for Ruby API 參考 [DetachRolePolicy](#) 中的。

將政策與使用者分離

下列程式碼範例顯示如何從使用者中斷 IAM 政策。

Warning

為避免安全風險，在開發專用軟體或使用真實資料時，請勿使用 IAM 使用者進行身分驗證。相反地，搭配使用聯合功能和身分提供者，例如 [AWS IAM Identity Center](#)。

適用於 Ruby 的開發套件

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在 [AWS 設定和執行程式碼範例儲存庫](#)。

```
# Detaches a policy from a user
#
# @param user_name [String] The name of the user
# @param policy_arn [String] The ARN of the policy to detach
# @return [Boolean] true if the policy was successfully detached, false otherwise
def detach_user_policy(user_name, policy_arn)
  @iam_client.detach_user_policy(
    user_name: user_name,
```

```

    policy_arn: policy_arn
  )
  @logger.info("Policy '#{policy_arn}' detached from user '#{user_name}'
successfully.")
  true
rescue Aws::IAM::Errors::NoSuchEntity
  @logger.error("Error detaching policy: Policy or user does not exist.")
  false
rescue Aws::IAM::Errors::ServiceError => e
  @logger.error("Error detaching policy from user '#{user_name}': #{e.message}")
  false
end

```

- 如需 API 詳細資訊，請參閱 AWS SDK for Ruby API 參考 [DetachUserPolicy](#) 中的。

取得政策

下列程式碼範例顯示如何取得 IAM 政策。

適用於 Ruby 的開發套件

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在 [AWS 設定和執行程式碼範例儲存庫](#)。

```

# Fetches an IAM policy by its ARN
# @param policy_arn [String] the ARN of the IAM policy to retrieve
# @return [Aws::IAM::Types::GetPolicyResponse] the policy object if found
def get_policy(policy_arn)
  response = @iam_client.get_policy(policy_arn: policy_arn)
  policy = response.policy
  @logger.info("Got policy '#{policy.policy_name}'. Its ID is:
#{policy.policy_id}.")
  policy
rescue Aws::IAM::Errors::NoSuchEntity
  @logger.error("Couldn't get policy '#{policy_arn}'. The policy does not exist.")
  raise
rescue Aws::IAM::Errors::ServiceError => e
  @logger.error("Couldn't get policy '#{policy_arn}'. Here's why: #{e.code}:
#{e.message}")

```

```
    raise
  end
```

- 如需 API 詳細資訊，請參閱 AWS SDK for Ruby API 參考[GetPolicy](#)中的。

取得角色

下列程式碼範例顯示如何取得 IAM 角色。

適用於 Ruby 的開發套件

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫](#)。

```
# Gets data about a role.
#
# @param name [String] The name of the role to look up.
# @return [Aws::IAM::Role] The retrieved role.
def get_role(name)
  role = @iam_client.get_role({
    role_name: name,
  }).role
  puts("Got data for role '#{role.role_name}'. Its ARN is '#{role.arn}'.")
rescue Aws::Errors::ServiceError => e
  puts("Couldn't get data for role '#{name}' Here's why:")
  puts("\t#{e.code}: #{e.message}")
  raise
else
  role
end
```

- 如需 API 詳細資訊，請參閱 AWS SDK for Ruby API 參考[GetRole](#)中的。

取得使用者

下列程式碼範例示範如何獲取 IAM 使用者。

⚠ Warning

為避免安全風險，在開發專用軟體或使用真實資料時，請勿使用 IAM 使用者進行身分驗證。相反地，搭配使用聯合功能和身分提供者，例如 [AWS IAM Identity Center](#)。

適用於 Ruby 的開發套件

i Note

還有更多關於 GitHub。尋找完整範例，並了解如何在 [AWS 設定和執行程式碼範例儲存庫](#)。

```
# Retrieves a user's details
#
# @param user_name [String] The name of the user to retrieve
# @return [Aws::IAM::Types::User, nil] The user object if found, or nil if an
error occurred
def get_user(user_name)
  response = @iam_client.get_user(user_name: user_name)
  response.user
rescue Aws::IAM::Errors::NoSuchEntity
  @logger.error("User '#{user_name}' not found.")
  nil
rescue Aws::IAM::Errors::ServiceError => e
  @logger.error("Error retrieving user '#{user_name}': #{e.message}")
  nil
end
```

- 如需 API 詳細資訊，請參閱 AWS SDK for Ruby API 參考 [GetUser](#) 中的。

取得帳戶密碼政策

下列程式碼範例顯示如何取得 IAM 帳戶密碼政策。

適用於 Ruby 的開發套件

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫](#)。

```
# Class to manage IAM account password policies
class PasswordPolicyManager
  attr_accessor :iam_client, :logger

  def initialize(iam_client, logger: Logger.new($stdout))
    @iam_client = iam_client
    @logger = logger
    @logger.progname = "IAMPolicyManager"
  end

  # Retrieves and logs the account password policy
  def print_account_password_policy
    begin
      response = @iam_client.get_account_password_policy
      @logger.info("The account password policy is:
#{response.password_policy.to_h}")
      rescue Aws::IAM::Errors::NoSuchEntity
        @logger.info("The account does not have a password policy.")
      rescue Aws::Errors::ServiceError => e
        @logger.error("Couldn't print the account password policy. Error: #{e.code} -
#{e.message}")
        raise
      end
    end
  end
end
```

- 如需 API 詳細資訊，請參閱 AWS SDK for Ruby API 參考[GetAccountPasswordPolicy](#)中的。

列出 SAML 供應商

下列程式碼範例顯示如何列出 IAM 的 SAML 提供者。

適用於 Ruby 的開發套件

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫](#)。

```
class SamlProviderLister
  # Initializes the SamlProviderLister with IAM client and a logger.
  # @param iam_client [Aws::IAM::Client] The IAM client object.
  # @param logger [Logger] The logger object for logging output.
  def initialize(iam_client, logger = Logger.new($stdout))
    @iam_client = iam_client
    @logger = logger
  end

  # Lists up to a specified number of SAML providers for the account.
  # @param count [Integer] The maximum number of providers to list.
  # @return [Aws::IAM::Client::Response]
  def list_saml_providers(count)
    response = @iam_client.list_saml_providers
    response.saml_provider_list.take(count).each do |provider|
      @logger.info("\t#{provider.arn}")
    end
    response
  rescue Aws::Errors::ServiceError => e
    @logger.error("Couldn't list SAML providers. Here's why:")
    @logger.error("\t#{e.code}: #{e.message}")
    raise
  end
end
```

- 如需 API 詳細資訊，請參閱《AWS SDK for Ruby API 參考》中的 [ListSAMLProviders](#)。

列出使用者的存取金鑰

下列程式碼範例示範如何列出使用者的 IAM 存取金鑰。

⚠ Warning

為避免安全風險，在開發專用軟體或使用真實資料時，請勿使用 IAM 使用者進行身分驗證。相反地，搭配使用聯合功能和身分提供者，例如 [AWS IAM Identity Center](#)。

適用於 Ruby 的開發套件

i Note

還有更多關於 GitHub。尋找完整範例，並了解如何在 [AWS 設定和執行程式碼範例儲存庫](#)。

此範例模組會列出、建立、停用及刪除存取金鑰。

```
# Manages access keys for IAM users
class AccessKeyManager
  def initialize(iam_client, logger: Logger.new($stdout))
    @iam_client = iam_client
    @logger = logger
    @logger.progname = "AccessKeyManager"
  end

  # Lists access keys for a user
  #
  # @param user_name [String] The name of the user.
  def list_access_keys(user_name)
    response = @iam_client.list_access_keys(user_name: user_name)
    if response.access_key_metadata.empty?
      @logger.info("No access keys found for user '#{user_name}'.")
    else
      response.access_key_metadata.map(&:access_key_id)
    end
  rescue Aws::IAM::Errors::NoSuchEntity => e
    @logger.error("Error listing access keys: cannot find user '#{user_name}'.")
    []
  rescue StandardError => e
    @logger.error("Error listing access keys: #{e.message}")
    []
  end

  # Creates an access key for a user
```



```
#
# @param user_name [String] The name of the user.
# @return [Boolean]
def create_access_key(user_name)
  response = @iam_client.create_access_key(user_name: user_name)
  access_key = response.access_key
  @logger.info("Access key created for user '#{user_name}':
#{access_key.access_key_id}")
  access_key
rescue Aws::IAM::Errors::LimitExceeded => e
  @logger.error("Error creating access key: limit exceeded. Cannot create more.")
  nil
rescue StandardError => e
  @logger.error("Error creating access key: #{e.message}")
  nil
end

# Deactivates an access key
#
# @param user_name [String] The name of the user.
# @param access_key_id [String] The ID for the access key.
# @return [Boolean]
def deactivate_access_key(user_name, access_key_id)
  @iam_client.update_access_key(
    user_name: user_name,
    access_key_id: access_key_id,
    status: "Inactive"
  )
  true
rescue StandardError => e
  @logger.error("Error deactivating access key: #{e.message}")
  false
end

# Deletes an access key
#
# @param user_name [String] The name of the user.
# @param access_key_id [String] The ID for the access key.
# @return [Boolean]
def delete_access_key(user_name, access_key_id)
  @iam_client.delete_access_key(
    user_name: user_name,
    access_key_id: access_key_id
  )
end
```

```
    true
  rescue StandardError => e
    @logger.error("Error deleting access key: #{e.message}")
    false
  end
end
```

- 如需 API 詳細資訊，請參閱 AWS SDK for Ruby API 參考 [ListAccessKeys](#) 中的。

列出帳戶別名

下列程式碼範例顯示如何列出 IAM 帳戶別名。

適用於 Ruby 的開發套件

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在 [AWS 設定和執行程式碼範例儲存庫](#)。

列出、建立及刪除帳戶別名。

```
class IAMAliasManager
  # Initializes the IAM client and logger
  #
  # @param iam_client [Aws::IAM::Client] An initialized IAM client.
  def initialize(iam_client, logger: Logger.new($stdout))
    @iam_client = iam_client
    @logger = logger
  end

  # Lists available AWS account aliases.
  def list_aliases
    response = @iam_client.list_account_aliases

    if response.account_aliases.count.positive?
      @logger.info("Account aliases are:")
      response.account_aliases.each { |account_alias| @logger.info("#{account_alias}") }
    else
      @logger.info("No account aliases found.")
    end
  end
end
```

```
    end
  rescue Aws::IAM::Errors::ServiceError => e
    @logger.error("Error listing account aliases: #{e.message}")
  end

  # Creates an AWS account alias.
  #
  # @param account_alias [String] The name of the account alias to create.
  # @return [Boolean] true if the account alias was created; otherwise, false.
  def create_account_alias(account_alias)
    @iam_client.create_account_alias(account_alias: account_alias)
    true
  rescue Aws::IAM::Errors::ServiceError => e
    @logger.error("Error creating account alias: #{e.message}")
    false
  end

  # Deletes an AWS account alias.
  #
  # @param account_alias [String] The name of the account alias to delete.
  # @return [Boolean] true if the account alias was deleted; otherwise, false.
  def delete_account_alias(account_alias)
    @iam_client.delete_account_alias(account_alias: account_alias)
    true
  rescue Aws::IAM::Errors::ServiceError => e
    @logger.error("Error deleting account alias: #{e.message}")
    false
  end
end
end
```

- 如需 API 詳細資訊，請參閱 AWS SDK for Ruby API 參考 [ListAccountAliases](#) 中的。

列出群組

下列程式碼範例顯示如何列出 IAM 群組。

適用於 Ruby 的開發套件

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在 [AWS 設定和執行程式碼範例儲存庫](#)。

```
# A class to manage IAM operations via the AWS SDK client
class IamGroupManager
  # Initializes the IamGroupManager class
  # @param iam_client [Aws::IAM::Client] An instance of the IAM client
  def initialize(iam_client, logger: Logger.new($stdout))
    @iam_client = iam_client
    @logger = logger
  end

  # Lists up to a specified number of groups for the account.
  # @param count [Integer] The maximum number of groups to list.
  # @return [Aws::IAM::Client::Response]
  def list_groups(count)
    response = @iam_client.list_groups(max_items: count)
    response.groups.each do |group|
      @logger.info("\t#{group.group_name}")
    end
    response
  rescue Aws::Errors::ServiceError => e
    @logger.error("Couldn't list groups for the account. Here's why:")
    @logger.error("\t#{e.code}: #{e.message}")
    raise
  end
end
```

- 如需 API 詳細資訊，請參閱 AWS SDK for Ruby API 參考 [ListGroups](#) 中的。

列出角色的內嵌政策

下列程式碼範例顯示如何列出 IAM 角色的內嵌政策。

適用於 Ruby 的開發套件

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在 [AWS 設定和執行程式碼範例儲存庫](#)。

```
# Lists policy ARNs attached to a role
#
```

```
# @param role_name [String] The name of the role
# @return [Array<String>] List of policy ARNs
def list_attached_policy_arns(role_name)
  response = @iam_client.list_attached_role_policies(role_name: role_name)
  response.attached_policies.map(&:policy_arn)
rescue Aws::IAM::Errors::ServiceError => e
  @logger.error("Error listing policies attached to role: #{e.message}")
  []
end
```

- 如需 API 詳細資訊，請參閱 AWS SDK for Ruby API 參考[ListRolePolicies](#)中的。

列出政策

下列程式碼範例顯示如何列出 IAM 政策。

適用於 Ruby 的開發套件

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫](#)。

此範例模組會列出、建立、附加和解除連結角色原則。

```
# Manages policies in AWS Identity and Access Management (IAM)
class RolePolicyManager
  # Initialize with an AWS IAM client
  #
  # @param iam_client [Aws::IAM::Client] An initialized IAM client
  def initialize(iam_client, logger: Logger.new($stdout))
    @iam_client = iam_client
    @logger = logger
    @logger.progname = "PolicyManager"
  end

  # Creates a policy
  #
  # @param policy_name [String] The name of the policy
  # @param policy_document [Hash] The policy document
  # @return [String] The policy ARN if successful, otherwise nil
```

```
def create_policy(policy_name, policy_document)
  response = @iam_client.create_policy(
    policy_name: policy_name,
    policy_document: policy_document.to_json
  )
  response.policy.arn
rescue Aws::IAM::Errors::ServiceError => e
  @logger.error("Error creating policy: #{e.message}")
  nil
end

# Fetches an IAM policy by its ARN
# @param policy_arn [String] the ARN of the IAM policy to retrieve
# @return [Aws::IAM::Types::GetPolicyResponse] the policy object if found
def get_policy(policy_arn)
  response = @iam_client.get_policy(policy_arn: policy_arn)
  policy = response.policy
  @logger.info("Got policy '#{policy.policy_name}'. Its ID is:
#{policy.policy_id}.")
  policy
rescue Aws::IAM::Errors::NoSuchEntity
  @logger.error("Couldn't get policy '#{policy_arn}'. The policy does not exist.")
  raise
rescue Aws::IAM::Errors::ServiceError => e
  @logger.error("Couldn't get policy '#{policy_arn}'. Here's why: #{e.code}:
#{e.message}")
  raise
end

# Attaches a policy to a role
#
# @param role_name [String] The name of the role
# @param policy_arn [String] The policy ARN
# @return [Boolean] true if successful, false otherwise
def attach_policy_to_role(role_name, policy_arn)
  @iam_client.attach_role_policy(
    role_name: role_name,
    policy_arn: policy_arn
  )
  true
rescue Aws::IAM::Errors::ServiceError => e
  @logger.error("Error attaching policy to role: #{e.message}")
  false
end
```

```
# Lists policy ARNs attached to a role
#
# @param role_name [String] The name of the role
# @return [Array<String>] List of policy ARNs
def list_attached_policy_arns(role_name)
  response = @iam_client.list_attached_role_policies(role_name: role_name)
  response.attached_policies.map(&:policy_arn)
rescue Aws::IAM::Errors::ServiceError => e
  @logger.error("Error listing policies attached to role: #{e.message}")
  []
end

# Detaches a policy from a role
#
# @param role_name [String] The name of the role
# @param policy_arn [String] The policy ARN
# @return [Boolean] true if successful, false otherwise
def detach_policy_from_role(role_name, policy_arn)
  @iam_client.detach_role_policy(
    role_name: role_name,
    policy_arn: policy_arn
  )
  true
rescue Aws::IAM::Errors::ServiceError => e
  @logger.error("Error detaching policy from role: #{e.message}")
  false
end
end
```

- 如需 API 詳細資訊，請參閱 AWS SDK for Ruby API 參考[ListPolicies](#)中的。

列出連接至角色的政策

下列程式碼範例顯示如何列出附加至 IAM 角色的政策。

適用於 Ruby 的開發套件

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫](#)。

此範例模組會列出、建立、附加和解除連結角色原則。

```
# Manages policies in AWS Identity and Access Management (IAM)
class RolePolicyManager
  # Initialize with an AWS IAM client
  #
  # @param iam_client [Aws::IAM::Client] An initialized IAM client
  def initialize(iam_client, logger: Logger.new($stdout))
    @iam_client = iam_client
    @logger = logger
    @logger.progname = "PolicyManager"
  end

  # Creates a policy
  #
  # @param policy_name [String] The name of the policy
  # @param policy_document [Hash] The policy document
  # @return [String] The policy ARN if successful, otherwise nil
  def create_policy(policy_name, policy_document)
    response = @iam_client.create_policy(
      policy_name: policy_name,
      policy_document: policy_document.to_json
    )
    response.policy.arn
  rescue Aws::IAM::Errors::ServiceError => e
    @logger.error("Error creating policy: #{e.message}")
    nil
  end

  # Fetches an IAM policy by its ARN
  # @param policy_arn [String] the ARN of the IAM policy to retrieve
  # @return [Aws::IAM::Types::GetPolicyResponse] the policy object if found
  def get_policy(policy_arn)
    response = @iam_client.get_policy(policy_arn: policy_arn)
    policy = response.policy
    @logger.info("Got policy '#{policy.policy_name}'. Its ID is:
    #{policy.policy_id}.")
    policy
  rescue Aws::IAM::Errors::NoSuchEntity
    @logger.error("Couldn't get policy '#{policy_arn}'. The policy does not exist.")
    raise
  rescue Aws::IAM::Errors::ServiceError => e
    @logger.error("Couldn't get policy '#{policy_arn}'. Here's why: #{e.code}:
    #{e.message}")
  end
end
```



```
    raise
  end

  # Attaches a policy to a role
  #
  # @param role_name [String] The name of the role
  # @param policy_arn [String] The policy ARN
  # @return [Boolean] true if successful, false otherwise
  def attach_policy_to_role(role_name, policy_arn)
    @iam_client.attach_role_policy(
      role_name: role_name,
      policy_arn: policy_arn
    )
    true
  rescue Aws::IAM::Errors::ServiceError => e
    @logger.error("Error attaching policy to role: #{e.message}")
    false
  end

  # Lists policy ARNs attached to a role
  #
  # @param role_name [String] The name of the role
  # @return [Array<String>] List of policy ARNs
  def list_attached_policy_arns(role_name)
    response = @iam_client.list_attached_role_policies(role_name: role_name)
    response.attached_policies.map(&:policy_arn)
  rescue Aws::IAM::Errors::ServiceError => e
    @logger.error("Error listing policies attached to role: #{e.message}")
    []
  end

  # Detaches a policy from a role
  #
  # @param role_name [String] The name of the role
  # @param policy_arn [String] The policy ARN
  # @return [Boolean] true if successful, false otherwise
  def detach_policy_from_role(role_name, policy_arn)
    @iam_client.detach_role_policy(
      role_name: role_name,
      policy_arn: policy_arn
    )
    true
  rescue Aws::IAM::Errors::ServiceError => e
    @logger.error("Error detaching policy from role: #{e.message}")
  end
end
```

```
    false
  end
end
```

- 如需 API 詳細資訊，請參閱 AWS SDK for Ruby API 參考[ListAttachedRolePolicies](#)中的。

列出角色

下列程式碼範例顯示如何列出 IAM 角色。

適用於 Ruby 的開發套件

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫](#)。

```
# Lists IAM roles up to a specified count.
# @param count [Integer] the maximum number of roles to list.
# @return [Array<String>] the names of the roles.
def list_roles(count)
  role_names = []
  roles_counted = 0

  @iam_client.list_roles.each_page do |page|
    page.roles.each do |role|
      break if roles_counted >= count
      @logger.info("\t#{roles_counted + 1}: #{role.role_name}")
      role_names << role.role_name
      roles_counted += 1
    end
    break if roles_counted >= count
  end

  role_names
rescue Aws::IAM::Errors::ServiceError => e
  @logger.error("Couldn't list roles for the account. Here's why:")
  @logger.error("\t#{e.code}: #{e.message}")
  raise
end
```

- 如需 API 詳細資訊，請參閱 AWS SDK for Ruby API 參考 [ListRoles](#) 中的。

列出伺服器憑證

下列程式碼範例示範如何列出 IAM 伺服器憑證。

適用於 Ruby 的開發套件

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在 [AWS 設定和執行程式碼範例儲存庫](#)。

列出、更新和刪除伺服器憑證。

```
class ServerCertificateManager
  def initialize(iam_client, logger: Logger.new($stdout))
    @iam_client = iam_client
    @logger = logger
    @logger.progname = "ServerCertificateManager"
  end

  # Creates a new server certificate.
  # @param name [String] the name of the server certificate
  # @param certificate_body [String] the contents of the certificate
  # @param private_key [String] the private key contents
  # @return [Boolean] returns true if the certificate was successfully created
  def create_server_certificate(name, certificate_body, private_key)
    @iam_client.upload_server_certificate({
      server_certificate_name: name,
      certificate_body: certificate_body,
      private_key: private_key,
    })

    true
  rescue Aws::IAM::Errors::ServiceError => e
    puts "Failed to create server certificate: #{e.message}"
    false
  end

  # Lists available server certificate names.
```

```
def list_server_certificate_names
  response = @iam_client.list_server_certificates

  if response.server_certificate_metadata_list.empty?
    @logger.info("No server certificates found.")
    return
  end

  response.server_certificate_metadata_list.each do |certificate_metadata|
    @logger.info("Certificate Name:
#{certificate_metadata.server_certificate_name}")
  end
rescue Aws::IAM::Errors::ServiceError => e
  @logger.error("Error listing server certificates: #{e.message}")
end

# Updates the name of a server certificate.
def update_server_certificate_name(current_name, new_name)
  @iam_client.update_server_certificate(
    server_certificate_name: current_name,
    new_server_certificate_name: new_name
  )
  @logger.info("Server certificate name updated from '#{current_name}' to
'#{new_name}'.")
  true
rescue Aws::IAM::Errors::ServiceError => e
  @logger.error("Error updating server certificate name: #{e.message}")
  false
end

# Deletes a server certificate.
def delete_server_certificate(name)
  @iam_client.delete_server_certificate(server_certificate_name: name)
  @logger.info("Server certificate '#{name}' deleted.")
  true
rescue Aws::IAM::Errors::ServiceError => e
  @logger.error("Error deleting server certificate: #{e.message}")
  false
end
end
```

- 如需 API 詳細資訊，請參閱 AWS SDK for Ruby API 參考 [ListServerCertificates](#) 中的。

列出使用者

下列程式碼範例示範如何列出 IAM 使用者。

Warning

為避免安全風險，在開發專用軟體或使用真實資料時，請勿使用 IAM 使用者進行身分驗證。相反地，搭配使用聯合功能和身分提供者，例如 [AWS IAM Identity Center](#)。

適用於 Ruby 的開發套件

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在 [AWS 設定和執行程式碼範例儲存庫](#)。


```
# Lists all users in the AWS account
#
# @return [Array<Aws::IAM::Types::User>] An array of user objects
def list_users
  users = []
  @iam_client.list_users.each_page do |page|
    page.users.each do |user|
      users << user
    end
  end
  users
rescue Aws::IAM::Errors::ServiceError => e
  @logger.error("Error listing users: #{e.message}")
  []
end
```

- 如需 API 詳細資訊，請參閱 AWS SDK for Ruby API 參考 [ListUsers](#) 中的。

更新伺服器憑證

下列程式碼範例示範如何更新 IAM 伺服器憑證。

適用於 Ruby 的開發套件

 Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫](#)。

列出、更新和刪除伺服器憑證。

```
class ServerCertificateManager
  def initialize(iam_client, logger: Logger.new($stdout))
    @iam_client = iam_client
    @logger = logger
    @logger.progname = "ServerCertificateManager"
  end

  # Creates a new server certificate.
  # @param name [String] the name of the server certificate
  # @param certificate_body [String] the contents of the certificate
  # @param private_key [String] the private key contents
  # @return [Boolean] returns true if the certificate was successfully created
  def create_server_certificate(name, certificate_body, private_key)
    @iam_client.upload_server_certificate({
      server_certificate_name: name,
      certificate_body: certificate_body,
      private_key: private_key,
    })

    true
  rescue Aws::IAM::Errors::ServiceError => e
    puts "Failed to create server certificate: #{e.message}"
    false
  end

  # Lists available server certificate names.
  def list_server_certificate_names
    response = @iam_client.list_server_certificates

    if response.server_certificate_metadata_list.empty?
      @logger.info("No server certificates found.")
      return
    end

    response.server_certificate_metadata_list.each do |certificate_metadata|
```

```
@logger.info("Certificate Name:
#{certificate_metadata.server_certificate_name}")
end
rescue Aws::IAM::Errors::ServiceError => e
  @logger.error("Error listing server certificates: #{e.message}")
end

# Updates the name of a server certificate.
def update_server_certificate_name(current_name, new_name)
  @iam_client.update_server_certificate(
    server_certificate_name: current_name,
    new_server_certificate_name: new_name
  )
  @logger.info("Server certificate name updated from '#{current_name}' to
'#{new_name}'.")
  true
rescue Aws::IAM::Errors::ServiceError => e
  @logger.error("Error updating server certificate name: #{e.message}")
  false
end

# Deletes a server certificate.
def delete_server_certificate(name)
  @iam_client.delete_server_certificate(server_certificate_name: name)
  @logger.info("Server certificate '#{name}' deleted.")
  true
rescue Aws::IAM::Errors::ServiceError => e
  @logger.error("Error deleting server certificate: #{e.message}")
  false
end
end
end
```

- 如需 API 詳細資訊，請參閱 AWS SDK for Ruby API 參考[UpdateServerCertificate](#)中的。

更新使用者

下列程式碼範例顯示如何更新 IAM 使用者。

⚠ Warning

為避免安全風險，在開發專用軟體或使用真實資料時，請勿使用 IAM 使用者進行身分驗證。相反地，搭配使用聯合功能和身分提供者，例如 [AWS IAM Identity Center](#)。

適用於 Ruby 的開發套件

i Note

還有更多關於 GitHub。尋找完整範例，並了解如何在 [AWS 設定和執程式碼範例儲存庫](#)。

```
# Updates an IAM user's name
#
# @param current_name [String] The current name of the user
# @param new_name [String] The new name of the user
def update_user_name(current_name, new_name)
  @iam_client.update_user(user_name: current_name, new_user_name: new_name)
  true
rescue StandardError => e
  @logger.error("Error updating user name from '#{current_name}' to '#{new_name}':
#{e.message}")
  false
end
```

- 如需 API 詳細資訊，請參閱 AWS SDK for Ruby API 參考 [UpdateUser](#) 中的。

案例

建立使用者並擔任角色


下列程式碼範例示範如何建立使用者並擔任角色。

⚠ Warning

為避免安全風險，在開發專用軟體或使用真實資料時，請勿使用 IAM 使用者進行身分驗證。相反地，搭配使用聯合功能和身分提供者，例如 [AWS IAM Identity Center](#)。

- 建立沒有許可的使用者。
- 建立一個可授予許可的角色，以列出帳戶的 Amazon S3 儲存貯體。
- 新增政策，讓使用者擔任該角色。
- 使用暫時憑證，擔任角色並列出 Amazon S3 儲存貯體，然後清理資源。

適用於 Ruby 的開發套件

 Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫](#)。

建立一個可授予許可的 IAM 使用者和角色，以列出 Amazon S3 儲存貯體。使用者只有擔任該角色的權利。擔任角色後，請使用暫時性憑證列出該帳戶的儲存貯體。

```
# Wraps the scenario actions.
class ScenarioCreateUserAssumeRole
  attr_reader :iam_client

  # @param [Aws::IAM::Client] iam_client: The AWS IAM client.
  def initialize(iam_client, logger: Logger.new($stdout))
    @iam_client = iam_client
    @logger = logger
  end

  # Waits for the specified number of seconds.
  #
  # @param duration [Integer] The number of seconds to wait.
  def wait(duration)
    puts("Give AWS time to propagate resources...")
    sleep(duration)
  end

  # Creates a user.
  #
  # @param user_name [String] The name to give the user.
  # @return [Aws::IAM::User] The newly created user.
  def create_user(user_name)
    user = @iam_client.create_user(user_name: user_name).user
    @logger.info("Created demo user named #{user.user_name}.")
    rescue Aws::Errors::ServiceError => e
```

```
@logger.info("Tried and failed to create demo user.")
@logger.info("\t#{e.code}: #{e.message}")
@logger.info("\nCan't continue the demo without a user!")
raise
else
  user
end

# Creates an access key for a user.
#
# @param user [Aws::IAM::User] The user that owns the key.
# @return [Aws::IAM::AccessKeyPair] The newly created access key.
def create_access_key_pair(user)
  user_key = @iam_client.create_access_key(user_name: user.user_name).access_key
  @logger.info("Created accesskey pair for user #{user.user_name}.")
rescue Aws::Errors::ServiceError => e
  @logger.info("Couldn't create access keys for user #{user.user_name}.")
  @logger.info("\t#{e.code}: #{e.message}")
  raise
else
  user_key
end

# Creates a role that can be assumed by a user.
#
# @param role_name [String] The name to give the role.
# @param user [Aws::IAM::User] The user who is granted permission to assume the
role.
# @return [Aws::IAM::Role] The newly created role.
def create_role(role_name, user)
  trust_policy = {
    Version: "2012-10-17",
    Statement: [{
      Effect: "Allow",
      Principal: {'AWS': user.arn},
      Action: "sts:AssumeRole"
    }]
  }.to_json
  role = @iam_client.create_role(
    role_name: role_name,
    assume_role_policy_document: trust_policy
  ).role
  @logger.info("Created role #{role.role_name}.")
rescue Aws::Errors::ServiceError => e
```

```
@logger.info("Couldn't create a role for the demo. Here's why: ")
@logger.info("\t#{e.code}: #{e.message}")
  raise
else
  role
end

# Creates a policy that grants permission to list S3 buckets in the account, and
# then attaches the policy to a role.
#
# @param policy_name [String] The name to give the policy.
# @param role [Aws::IAM::Role] The role that the policy is attached to.
# @return [Aws::IAM::Policy] The newly created policy.
def create_and_attach_role_policy(policy_name, role)
  policy_document = {
    Version: "2012-10-17",
    Statement: [{
      Effect: "Allow",
      Action: "s3:ListAllMyBuckets",
      Resource: "arn:aws:s3:::*"
    }]
  }.to_json
  policy = @iam_client.create_policy(
    policy_name: policy_name,
    policy_document: policy_document
  ).policy
  @iam_client.attach_role_policy(
    role_name: role.role_name,
    policy_arn: policy.arn
  )
  @logger.info("Created policy #{policy.policy_name} and attached it to role
#{role.role_name}.")
  rescue Aws::Errors::ServiceError => e
    @logger.info("Couldn't create a policy and attach it to role #{role.role_name}.
Here's why: ")
    @logger.info("\t#{e.code}: #{e.message}")
    raise
  end

# Creates an inline policy for a user that lets the user assume a role.
#
# @param policy_name [String] The name to give the policy.
# @param user [Aws::IAM::User] The user that owns the policy.
# @param role [Aws::IAM::Role] The role that can be assumed.
```

```
# @return [Aws::IAM::UserPolicy] The newly created policy.
def create_user_policy(policy_name, user, role)
  policy_document = {
    Version: "2012-10-17",
    Statement: [{
      Effect: "Allow",
      Action: "sts:AssumeRole",
      Resource: role.arn
    }]
  }.to_json
  @iam_client.put_user_policy(
    user_name: user.user_name,
    policy_name: policy_name,
    policy_document: policy_document
  )
  puts("Created an inline policy for #{user.user_name} that lets the user assume
role #{role.role_name}.")
  rescue Aws::Errors::ServiceError => e
    @logger.info("Couldn't create an inline policy for user #{user.user_name}.
Here's why: ")
    @logger.info("\t#{e.code}: #{e.message}")
    raise
  end

  # Creates an Amazon S3 resource with specified credentials. This is separated into
a
  # factory function so that it can be mocked for unit testing.
  #
  # @param credentials [Aws::Credentials] The credentials used by the Amazon S3
resource.
  def create_s3_resource(credentials)
    Aws::S3::Resource.new(client: Aws::S3::Client.new(credentials: credentials))
  end

  # Lists the S3 buckets for the account, using the specified Amazon S3 resource.
  # Because the resource uses credentials with limited access, it may not be able to
  # list the S3 buckets.
  #
  # @param s3_resource [Aws::S3::Resource] An Amazon S3 resource.
  def list_buckets(s3_resource)
    count = 10
    s3_resource.buckets.each do |bucket|
      @logger.info "\t#{bucket.name}"
      count -= 1
    end
  end
end
```

```
        break if count.zero?
      end
    rescue Aws::Errors::ServiceError => e
      if e.code == "AccessDenied"
        puts("Attempt to list buckets with no permissions: AccessDenied.")
      else
        @logger.info("Couldn't list buckets for the account. Here's why: ")
        @logger.info("\t#{e.code}: #{e.message}")
        raise
      end
    end
  end

  # Creates an AWS Security Token Service (AWS STS) client with specified
  # credentials.
  # This is separated into a factory function so that it can be mocked for unit
  # testing.
  #
  # @param key_id [String] The ID of the access key used by the STS client.
  # @param key_secret [String] The secret part of the access key used by the STS
  # client.
  def create_sts_client(key_id, key_secret)
    Aws::STS::Client.new(access_key_id: key_id, secret_access_key: key_secret)
  end

  # Gets temporary credentials that can be used to assume a role.
  #
  # @param role_arn [String] The ARN of the role that is assumed when these
  # credentials
  #           are used.
  # @param sts_client [AWS::STS::Client] An AWS STS client.
  # @return [Aws::AssumeRoleCredentials] The credentials that can be used to assume
  # the role.
  def assume_role(role_arn, sts_client)
    credentials = Aws::AssumeRoleCredentials.new(
      client: sts_client,
      role_arn: role_arn,
      role_session_name: "create-use-assume-role-scenario"
    )
    @logger.info("Assumed role '#{role_arn}', got temporary credentials.")
    credentials
  end

  # Deletes a role. If the role has policies attached, they are detached and
  # deleted before the role is deleted.
```

```
#
# @param role_name [String] The name of the role to delete.
def delete_role(role_name)
  @iam_client.list_attached_role_policies(role_name:
role_name).attached_policies.each do |policy|
    @iam_client.detach_role_policy(role_name: role_name, policy_arn:
policy.policy_arn)
    @iam_client.delete_policy(policy_arn: policy.policy_arn)
    @logger.info("Detached and deleted policy #{policy.policy_name}.")
  end
  @iam_client.delete_role({ role_name: role_name })
  @logger.info("Role deleted: #{role_name}.")
rescue Aws::Errors::ServiceError => e
  @logger.info("Couldn't detach policies and delete role #{role.name}. Here's
why:")
  @logger.info("\t#{e.code}: #{e.message}")
  raise
end

# Deletes a user. If the user has inline policies or access keys, they are deleted
# before the user is deleted.
#
# @param user [Aws::IAM::User] The user to delete.
def delete_user(user_name)
  user = @iam_client.list_access_keys(user_name: user_name).access_key_metadata
  user.each do |key|
    @iam_client.delete_access_key({ access_key_id: key.access_key_id, user_name:
user_name })
    @logger.info("Deleted access key #{key.access_key_id} for user
'#{user_name}'.")
  end

  @iam_client.delete_user(user_name: user_name)
  @logger.info("Deleted user '#{user_name}'.")
rescue Aws::IAM::Errors::ServiceError => e
  @logger.error("Error deleting user '#{user_name}': #{e.message}")
end
end

# Runs the IAM create a user and assume a role scenario.
def run_scenario(scenario)
  puts("-" * 88)
  puts("Welcome to the IAM create a user and assume a role demo!")
  puts("-" * 88)
```

```

user = scenario.create_user("doc-example-user-#{Random.uuid}")
user_key = scenario.create_access_key_pair(user)
scenario.wait(10)
role = scenario.create_role("doc-example-role-#{Random.uuid}", user)
scenario.create_and_attach_role_policy("doc-example-role-policy-#{Random.uuid}",
role)
scenario.create_user_policy("doc-example-user-policy-#{Random.uuid}", user, role)
scenario.wait(10)
puts("Try to list buckets with credentials for a user who has no permissions.")
puts("Expect AccessDenied from this call.")
scenario.list_buckets(
  scenario.create_s3_resource(Aws::Credentials.new(user_key.access_key_id,
user_key.secret_access_key)))
puts("Now, assume the role that grants permission.")
temp_credentials = scenario.assume_role(
  role.arn, scenario.create_sts_client(user_key.access_key_id,
user_key.secret_access_key))
puts("Here are your buckets:")
scenario.list_buckets(scenario.create_s3_resource(temp_credentials))
puts("Deleting role '#{role.role_name}' and attached policies.")
scenario.delete_role(role.role_name)
puts("Deleting user '#{user.user_name}', policies, and keys.")
scenario.delete_user(user.user_name)
puts("Thanks for watching!")
puts("-" * 88)
rescue Aws::Errors::ServiceError => e
  puts("Something went wrong with the demo.")
  puts("\t#{e.code}: #{e.message}")
end

run_scenario(ScenarioCreateUserAssumeRole.new(Aws::IAM::Client.new)) if
$PROGRAM_NAME == __FILE__

```

- 如需 API 詳細資訊，請參閱《AWS SDK for Ruby API 參考》中的下列主題。
 - [AttachRolePolicy](#)
 - [CreateAccessKey](#)
 - [CreatePolicy](#)
 - [CreateRole](#)
 - [CreateUser](#)
 - [DeleteAccessKey](#)

- [DeletePolicy](#)
- [DeleteRole](#)
- [DeleteUser](#)
- [DeleteUserPolicy](#)
- [DetachRolePolicy](#)
- [PutUserPolicy](#)

Kinesis 示例使用 SDK for Ruby

下列程式碼範例說明如何使用 Kinesis 來執行動作和實作常見案例。AWS SDK for Ruby

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境和跨服務範例中查看內容中的動作。

案例是向您展示如何呼叫相同服務中的多個函數來完成特定任務的程式碼範例。

每個範例都包含一個連結 GitHub，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- [無伺服器範例](#)

無伺服器範例

使用 Kinesis 觸發條件調用 Lambda 函數

下列程式碼範例示範如何實作 Lambda 函數，該函數會接收從 Kinesis 串流接收記錄而觸發的事件。此函數會擷取 Kinesis 承載、從 Base64 解碼，並記錄記錄內容。

適用於 Ruby 的開發套件

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[無伺服器範例](#)儲存庫中設定和執行。

使用紅寶石使用 Lambda 消耗 Kinesis 事件。

```
require 'aws-sdk'
```



```
def lambda_handler(event:, context:)
  event['Records'].each do |record|
    begin
      puts "Processed Kinesis Event - EventID: #{record['eventID']}"
      record_data = get_record_data_async(record['kinesis'])
      puts "Record Data: #{record_data}"
      # TODO: Do interesting work based on the new data
    rescue => err
      $stderr.puts "An error occurred #{err}"
      raise err
    end
  end
  puts "Successfully processed #{event['Records'].length} records."
end

def get_record_data_async(payload)
  data = Base64.decode64(payload['data']).force_encoding('UTF-8')
  # Placeholder for actual async work
  # You can use Ruby's asynchronous programming tools like async/await or fibers
  here.
  return data
end
```

使用 Kinesis 觸發條件報告 Lambda 函數的批次項目失敗

下列程式碼範例顯示如何針對接收來自 Kinesis 串流之事件的 Lambda 函數實作部分批次回應。此函數會在回應中報告批次項目失敗，指示 Lambda 稍後重試這些訊息。

適用於 Ruby 的開發套件

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[無伺服器範例](#)儲存庫中設定和執行。

使用 Ruby 使用 Lambda 報告 Kinesis 批次項目失敗。

```
require 'aws-sdk'

def lambda_handler(event:, context:)
  batch_item_failures = []
```

```
event['Records'].each do |record|
  begin
    puts "Processed Kinesis Event - EventID: #{record['eventID']}"
    record_data = get_record_data_async(record['kinesis'])
    puts "Record Data: #{record_data}"
    # TODO: Do interesting work based on the new data
  rescue StandardError => err
    puts "An error occurred #{err}"
    # Since we are working with streams, we can return the failed item
    immediately.
    # Lambda will immediately begin to retry processing from this failed item
    onwards.
    return { batchItemFailures: [{ itemIdentifier: record['kinesis']
['sequenceNumber'] }] }
  end
end

puts "Successfully processed #{event['Records'].length} records."
{ batchItemFailures: batch_item_failures }
end

def get_record_data_async(payload)
  data = Base64.decode64(payload['data']).force_encoding('utf-8')
  # Placeholder for actual async work
  sleep(1)
  data
end
```

AWS KMS 使用 SDK for Ruby 示例

下列程式碼範例說明如何使用 AWS SDK for Ruby 與來執行動作及實作常見案例 AWS KMS。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境和跨服務範例中查看內容中的動作。

案例是向您展示如何呼叫相同服務中的多個函數來完成特定任務的程式碼範例。

每個範例都包含一個連結 GitHub，您可以在其中找到如何在內容中設定和執程式碼的指示。

主題

- [動作](#)

動作

建立 金鑰

下列程式碼範例會示範如何建立 AWS KMS key.

適用於 Ruby 的開發套件

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫](#)。

```
require "aws-sdk-kms" # v2: require 'aws-sdk'

# Create a AWS KMS key.
# As long we are only encrypting small amounts of data (4 KiB or less) directly,
# a KMS key is fine for our purposes.
# For larger amounts of data,
# use the KMS key to encrypt a data encryption key (DEK).

client = Aws::KMS::Client.new

resp = client.create_key({
  tags: [
    {
      tag_key: "CreatedBy",
      tag_value: "ExampleUser"
    }
  ]
})

puts resp.key_metadata.key_id
```

- 如需 API 詳細資訊，請參閱 AWS SDK for Ruby API 參考[CreateKey](#)中的。

解密密文

下列程式碼範例顯示如何解密由 KMS 金鑰加密的加密文字。

適用於 Ruby 的開發套件

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫](#)。

```
require "aws-sdk-kms" # v2: require 'aws-sdk'

# Decrypted blob

blob =
  "01020200785d68faeec386af1057904926253051eb2919d3c16078badf65b808b26dd057c101747cadf3593596"
blob_packed = [blob].pack("H*")

client = Aws::KMS::Client.new(region: "us-west-2")

resp = client.decrypt({
  ciphertext_blob: blob_packed
})

puts "Raw text: "
puts resp.plaintext
```

- 如需 API 詳細資訊，請參閱在 AWS SDK for Ruby API 參考中[解密](#)。

使用金鑰加密文字

下列程式碼範例顯示如何使用 KMS 金鑰加密文字。

適用於 Ruby 的開發套件

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫](#)。

```
require "aws-sdk-kms" # v2: require 'aws-sdk'
```

```
# ARN of the AWS KMS key.
#
# Replace the fictitious key ARN with a valid key ID

keyId = "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"

text = "1234567890"

client = Aws::KMS::Client.new(region: "us-west-2")

resp = client.encrypt({
  key_id: keyId,
  plaintext: text,
})


# Display a readable version of the resulting encrypted blob.
puts "Blob:"
puts resp.ciphertext_blob.unpack("H*")
```

- 如需 API 詳細資訊，請參閱 AWS SDK for Ruby API 參考中的[加密](#)。

將密文從一個密鑰重新加密到另一個密鑰

下列程式碼範例顯示如何將密文從一個 KMS 金鑰重新加密到另一個 KMS 金鑰。

適用於 Ruby 的開發套件

 Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫](#)。

```
require "aws-sdk-kms" # v2: require 'aws-sdk'

# Human-readable version of the ciphertext of the data to reencrypt.

blob =
  "01020200785d68faeec386af1057904926253051eb2919d3c16078badf65b808b26dd057c101747cadf3593596"
sourceCiphertextBlob = [blob].pack("H*")
```

```
# Replace the fictitious key ARN with a valid key ID

destinationKeyId = "arn:aws:kms:us-west-2:111122223333:key/0987dcba-09fe-87dc-65ba-
ab0987654321"

client = Aws::KMS::Client.new(region: "us-west-2")

resp = client.re_encrypt({
  ciphertext_blob: sourceCiphertextBlob,
  destination_key_id: destinationKeyId
})

# Display a readable version of the resulting re-encrypted blob.
puts "Blob:"
puts resp.ciphertext_blob.unpack("H*")
```

- 如需 API 詳細資訊，請參閱 AWS SDK for Ruby API 參考[ReEncrypt](#)中的。

使用 SDK for Ruby Lambda

下列程式碼範例說明如何使用 Lambda 來執行動作和實作常見案例。AWS SDK for Ruby

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境和跨服務範例中查看內容中的動作。

案例是向您展示如何呼叫相同服務中的多個函數來完成特定任務的程式碼範例。

每個範例都包含一個連結 GitHub，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- [動作](#)
- [案例](#)
- [無伺服器範例](#)

動作

建立函數

下列程式碼範例會示範如何建立 Lambda 函數。

適用於 Ruby 的開發套件

 Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫](#)。

```
class LambdaWrapper
  attr_accessor :lambda_client

  def initialize
    @lambda_client = Aws::Lambda::Client.new
    @logger = Logger.new($stdout)
    @logger.level = Logger::WARN
  end

  # Deploys a Lambda function.
  #
  # @param function_name: The name of the Lambda function.
  # @param handler_name: The fully qualified name of the handler function. This
  #                       must include the file name and the function name.
  # @param role_arn: The IAM role to use for the function.
  # @param deployment_package: The deployment package that contains the function
  #                             code in .zip format.
  # @return: The Amazon Resource Name (ARN) of the newly created function.
  def create_function(function_name, handler_name, role_arn, deployment_package)
    response = @lambda_client.create_function({
      role: role_arn.to_s,
      function_name: function_name,
      handler: handler_name,
      runtime: "ruby2.7",
      code: {
        zip_file: deployment_package
      },
      environment: {
        variables: {
          "LOG_LEVEL" => "info"
        }
      }
    })
    @lambda_client.wait_until(:function_active_v2, { function_name: function_name})
  end
end

w = LambdaWrapper.new
w.create_function("my_function", "my_handler", "arn:aws:iam::123456789012:role/my_role", "my_package.zip")

w.max_attempts = 5
```

```

    w.delay = 5
  end
  response
rescue Aws::Lambda::Errors::ServiceException => e
  @logger.error("There was an error creating #{function_name}:\n #{e.message}")
rescue Aws::Waiters::Errors::WaiterFailed => e
  @logger.error("Failed waiting for #{function_name} to activate:\n #{e.message}")
end

```

- 如需 API 詳細資訊，請參閱 AWS SDK for Ruby API 參考[CreateFunction](#)中的。

刪除函數

下列程式碼範例會示範如何刪除 Lambda 函數。

適用於 Ruby 的開發套件

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫](#)。

```

class LambdaWrapper
  attr_accessor :lambda_client

  def initialize
    @lambda_client = Aws::Lambda::Client.new
    @logger = Logger.new($stdout)
    @logger.level = Logger::WARN
  end

  # Deletes a Lambda function.
  # @param function_name: The name of the function to delete.
  def delete_function(function_name)
    print "Deleting function: #{function_name}..."
    @lambda_client.delete_function(
      function_name: function_name
    )
    print "Done!".green
  rescue Aws::Lambda::Errors::ServiceException => e
    @logger.error("There was an error deleting #{function_name}:\n #{e.message}")
  end
end

```



```
end
```

- 如需 API 詳細資訊，請參閱 AWS SDK for Ruby API 參考[DeleteFunction](#)中的。

取得函數

下列程式碼範例顯示如何取得 Lambda 函數。

適用於 Ruby 的開發套件

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫](#)。

```
class LambdaWrapper
  attr_accessor :lambda_client

  def initialize
    @lambda_client = Aws::Lambda::Client.new
    @logger = Logger.new($stdout)
    @logger.level = Logger::WARN
  end

  # Gets data about a Lambda function.
  #
  # @param function_name: The name of the function.
  # @return response: The function data, or nil if no such function exists.
  def get_function(function_name)
    @lambda_client.get_function(
      {
        function_name: function_name
      }
    )
  rescue Aws::Lambda::Errors::ResourceNotFoundException => e
    @logger.debug("Could not find function: #{function_name}:\n #{e.message}")
    nil
  end
end
```

- 如需 API 詳細資訊，請參閱 AWS SDK for Ruby API 參考[GetFunction](#)中的。

呼叫函數

下列程式碼範例會示範如何叫用 Lambda 函數。

適用於 Ruby 的開發套件

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫](#)。

```
class LambdaWrapper
  attr_accessor :lambda_client

  def initialize
    @lambda_client = Aws::Lambda::Client.new
    @logger = Logger.new($stdout)
    @logger.level = Logger::WARN
  end

  # Invokes a Lambda function.
  # @param function_name [String] The name of the function to invoke.
  # @param payload [nil] Payload containing runtime parameters.
  # @return [Object] The response from the function invocation.
  def invoke_function(function_name, payload = nil)
    params = { function_name: function_name }
    params[:payload] = payload unless payload.nil?
    @lambda_client.invoke(params)
  rescue Aws::Lambda::Errors::ServiceException => e
    @logger.error("There was an error executing #{function_name}:\n #{e.message}")
  end
end
```

- 如需 API 的詳細資訊，請參閱《AWS SDK for Ruby API 參考》中的「[Invoke](#)」。

列出函數

下列程式碼範例顯示如何列出 Lambda 函數。

適用於 Ruby 的開發套件

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫](#)。

```
class LambdaWrapper
  attr_accessor :lambda_client

  def initialize
    @lambda_client = Aws::Lambda::Client.new
    @logger = Logger.new($stdout)
    @logger.level = Logger::WARN
  end

  # Lists the Lambda functions for the current account.
  def list_functions
    functions = []
    @lambda_client.list_functions.each do |response|
      response["functions"].each do |function|
        functions.append(function["function_name"])
      end
    end
    functions
  rescue Aws::Lambda::Errors::ServiceException => e
    @logger.error("There was an error executing #{function_name}:\n #{e.message}")
  end
end
```

- 如需 API 詳細資訊，請參閱 AWS SDK for Ruby API 參考[ListFunctions](#)中的。

更新函數程式碼

下列程式碼範例顯示如何更新 Lambda 函數程式碼。

適用於 Ruby 的開發套件

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫](#)。

```
class LambdaWrapper
  attr_accessor :lambda_client

  def initialize
    @lambda_client = Aws::Lambda::Client.new
    @logger = Logger.new($stdout)
    @logger.level = Logger::WARN
  end

  # Updates the code for a Lambda function by submitting a .zip archive that
  # contains
  # the code for the function.

  # @param function_name: The name of the function to update.
  # @param deployment_package: The function code to update, packaged as bytes in
  #                               .zip format.
  # @return: Data about the update, including the status.
  def update_function_code(function_name, deployment_package)
    @lambda_client.update_function_code(
      function_name: function_name,
      zip_file: deployment_package
    )
    @lambda_client.wait_until(:function_updated_v2, { function_name: function_name})
  do |w|
    w.max_attempts = 5
    w.delay = 5
  end
  rescue Aws::Lambda::Errors::ServiceException => e
    @logger.error("There was an error updating function code for: #{function_name}:
  \n #{e.message}")
    nil
  rescue Aws::Waiters::Errors::WaiterFailed => e
    @logger.error("Failed waiting for #{function_name} to update:\n #{e.message}")
  end
end
```

- 如需 API 詳細資訊，請參閱 AWS SDK for Ruby API 參考[UpdateFunctionCode](#)中的。

更新函數 URL 組態

下列程式碼範例顯示如何更新 Lambda 函數組態。

適用於 Ruby 的開發套件

 Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫](#)。

```
class LambdaWrapper
  attr_accessor :lambda_client

  def initialize
    @lambda_client = Aws::Lambda::Client.new
    @logger = Logger.new($stdout)
    @logger.level = Logger::WARN
  end

  # Updates the environment variables for a Lambda function.
  # @param function_name: The name of the function to update.
  # @param log_level: The log level of the function.
  # @return: Data about the update, including the status.
  def update_function_configuration(function_name, log_level)
    @lambda_client.update_function_configuration({
      function_name: function_name,
      environment: {
        variables: {
          "LOG_LEVEL" => log_level
        }
      }
    })

    @lambda_client.wait_until(:function_updated_v2, { function_name: function_name})
  end

  do |w|
    w.max_attempts = 5
    w.delay = 5
  end

  rescue Aws::Lambda::Errors::ServiceException => e
    @logger.error("There was an error updating configurations for #{function_name}:
  \n #{e.message}")
  rescue Aws::Waiters::Errors::WaiterFailed => e
    @logger.error("Failed waiting for #{function_name} to activate:\n #{e.message}")
  end
end
```

- 如需 API 詳細資訊，請參閱 AWS SDK for Ruby API 參考 [UpdateFunctionConfiguration](#) 中的。

案例

開始使用函數

以下程式碼範例顯示做法：

- 建立 IAM 角色和 Lambda 函數，然後上傳處理常式程式碼。
- 調用具有單一參數的函數並取得結果。
- 更新函數程式碼並使用環境變數進行設定。
- 調用具有新參數的函數並取得結果。顯示傳回的執行日誌。
- 列出您帳戶的函數，然後清理相關資源。

如需詳細資訊，請參閱 [使用主控台建立 Lambda 函數](#)。

適用於 Ruby 的開發套件

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在 [AWS 設定和執行程式碼範例儲存庫](#)。

為能夠寫入日誌的 Lambda 函數設定先決條件 IAM 許可。

```
# Get an AWS Identity and Access Management (IAM) role.
#
# @param iam_role_name: The name of the role to retrieve.
# @param action: Whether to create or destroy the IAM apparatus.
# @return: The IAM role.
def manage_iam(iam_role_name, action)
  role_policy = {
    'Version': "2012-10-17",
    'Statement': [
      {
        'Effect': "Allow",
        'Principal': {
          'Service': "lambda.amazonaws.com"
        },
        'Action': "sts:AssumeRole"
      }
    ]
  }
```

```

    }
  ]
}
case action
when "create"
  role = $iam_client.create_role(
    role_name: iam_role_name,
    assume_role_policy_document: role_policy.to_json
  )
  $iam_client.attach_role_policy(
    {
      policy_arn: "arn:aws:iam::aws:policy/service-role/
AWSLambdaBasicExecutionRole",
      role_name: iam_role_name
    }
  )
  $iam_client.wait_until(:role_exists, { role_name: iam_role_name }) do |w|
    w.max_attempts = 5
    w.delay = 5
  end
  @logger.debug("Successfully created IAM role: #{role['role']['arn']}")
  @logger.debug("Enforcing a 10-second sleep to allow IAM role to activate
fully.")
  sleep(10)
  return role, role_policy.to_json
when "destroy"
  $iam_client.detach_role_policy(
    {
      policy_arn: "arn:aws:iam::aws:policy/service-role/
AWSLambdaBasicExecutionRole",
      role_name: iam_role_name
    }
  )
  $iam_client.delete_role(
    role_name: iam_role_name
  )
  @logger.debug("Detached policy & deleted IAM role: #{iam_role_name}")
else
  raise "Incorrect action provided. Must provide 'create' or 'destroy'"
end
rescue Aws::Lambda::Errors::ServiceException => e
  @logger.error("There was an error creating role or attaching policy:\n
#{e.message}")
end

```

定義 Lambda 處理常式，該處理常式以作為調用參數提供的數字遞增。

```
require "logger"

# A function that increments a whole number by one (1) and logs the result.
# Requires a manually-provided runtime parameter, 'number', which must be Int
#
# @param event [Hash] Parameters sent when the function is invoked
# @param context [Hash] Methods and properties that provide information
# about the invocation, function, and execution environment.
# @return incremented_number [String] The incremented number.
def lambda_handler(event:, context:)
  logger = Logger.new($stdout)
  log_level = ENV["LOG_LEVEL"]
  logger.level = case log_level
                 when "debug"
                   Logger::DEBUG
                 when "info"
                   Logger::INFO
                 else
                   Logger::ERROR
                 end

  logger.debug("This is a debug log message.")
  logger.info("This is an info log message. Code executed successfully!")
  number = event["number"].to_i
  incremented_number = number + 1
  logger.info("You provided #{number.round} and it was incremented to
#{incremented_number.round}")
  incremented_number.round.to_s
end
```

將您的 Lambda 函數壓縮到部署套件中。

```
# Creates a Lambda deployment package in .zip format.
# This zip can be passed directly as a string to Lambda when creating the
function.
#
# @param source_file: The name of the object, without suffix, for the Lambda file
and zip.
# @return: The deployment package.
```



```

def create_deployment_package(source_file)
  Dir.chdir(File.dirname(__FILE__))
  if File.exist?("lambda_function.zip")
    File.delete("lambda_function.zip")
    @logger.debug("Deleting old zip: lambda_function.zip")
  end
  Zip::File.open("lambda_function.zip", create: true) {
    |zipfile|
    zipfile.add("lambda_function.rb", "#{source_file}.rb")
  }
  @logger.debug("Zipping #{source_file}.rb into: lambda_function.zip.")
  File.read("lambda_function.zip").to_s
rescue StandardError => e
  @logger.error("There was an error creating deployment package:\n #{e.message}")
end

```

建立新 Lambda 函數。

```

# Deploys a Lambda function.
#
# @param function_name: The name of the Lambda function.
# @param handler_name: The fully qualified name of the handler function. This
#                       must include the file name and the function name.
# @param role_arn: The IAM role to use for the function.
# @param deployment_package: The deployment package that contains the function
#                             code in .zip format.
# @return: The Amazon Resource Name (ARN) of the newly created function.
def create_function(function_name, handler_name, role_arn, deployment_package)
  response = @lambda_client.create_function({
    role: role_arn.to_s,
    function_name: function_name,
    handler: handler_name,
    runtime: "ruby2.7",
    code: {
      zip_file: deployment_package
    },
    environment: {
      variables: {
        "LOG_LEVEL" => "info"
      }
    }
  })
end

```

```

    @lambda_client.wait_until(:function_active_v2, { function_name: function_name})
  do |w|
    w.max_attempts = 5
    w.delay = 5
  end
  response
rescue Aws::Lambda::Errors::ServiceException => e
  @logger.error("There was an error creating #{function_name}:\n #{e.message}")
rescue Aws::Waiters::Errors::WaiterFailed => e
  @logger.error("Failed waiting for #{function_name} to activate:\n #{e.message}")
end

```

使用選用的執行階段參數調用 Lambda 函數。

```

# Invokes a Lambda function.
# @param function_name [String] The name of the function to invoke.
# @param payload [nil] Payload containing runtime parameters.
# @return [Object] The response from the function invocation.
def invoke_function(function_name, payload = nil)
  params = { function_name: function_name }
  params[:payload] = payload unless payload.nil?
  @lambda_client.invoke(params)
rescue Aws::Lambda::Errors::ServiceException => e
  @logger.error("There was an error executing #{function_name}:\n #{e.message}")
end

```

更新 Lambda 函數的組態以注入新的環境變數。

```

# Updates the environment variables for a Lambda function.
# @param function_name: The name of the function to update.
# @param log_level: The log level of the function.
# @return: Data about the update, including the status.
def update_function_configuration(function_name, log_level)
  @lambda_client.update_function_configuration({
    function_name: function_name,
    environment: {
      variables: {
        "LOG_LEVEL" => log_level
      }
    }
  })
end

```

```

    @lambda_client.wait_until(:function_updated_v2, { function_name: function_name})
  do |w|
    w.max_attempts = 5
    w.delay = 5
  end
  rescue Aws::Lambda::Errors::ServiceException => e
    @logger.error("There was an error updating configurations for #{function_name}:
\n #{e.message}")
  rescue Aws::Writers::Errors::WaiterFailed => e
    @logger.error("Failed waiting for #{function_name} to activate:\n #{e.message}")
  end

```

使用包含不同程式碼的不同部署套件來更新 Lambda 函數的程式碼。

```

# Updates the code for a Lambda function by submitting a .zip archive that
contains
# the code for the function.

# @param function_name: The name of the function to update.
# @param deployment_package: The function code to update, packaged as bytes in
#                               .zip format.
# @return: Data about the update, including the status.
def update_function_code(function_name, deployment_package)
  @lambda_client.update_function_code(
    function_name: function_name,
    zip_file: deployment_package
  )
  @lambda_client.wait_until(:function_updated_v2, { function_name: function_name})
do |w|
  w.max_attempts = 5
  w.delay = 5
end
  rescue Aws::Lambda::Errors::ServiceException => e
    @logger.error("There was an error updating function code for: #{function_name}:
\n #{e.message}")
    nil
  rescue Aws::Writers::Errors::WaiterFailed => e
    @logger.error("Failed waiting for #{function_name} to update:\n #{e.message}")
  end

```

使用內建分頁程式列出所有現有的 Lambda 函數。

```
# Lists the Lambda functions for the current account.
def list_functions
  functions = []
  @lambda_client.list_functions.each do |response|
    response["functions"].each do |function|
      functions.append(function["function_name"])
    end
  end
  functions
rescue Aws::Lambda::Errors::ServiceException => e
  @logger.error("There was an error executing #{function_name}:\n #{e.message}")
end
```

刪除特定 Lambda 函數。

```
# Deletes a Lambda function.
# @param function_name: The name of the function to delete.
def delete_function(function_name)
  print "Deleting function: #{function_name}..."
  @lambda_client.delete_function(
    function_name: function_name
  )
  print "Done!".green
rescue Aws::Lambda::Errors::ServiceException => e
  @logger.error("There was an error deleting #{function_name}:\n #{e.message}")
end
```

- 如需 API 詳細資訊，請參閱《AWS SDK for Ruby API 參考》中的下列主題。
 - [CreateFunction](#)
 - [DeleteFunction](#)
 - [GetFunction](#)
 - [Invoke](#)
 - [ListFunctions](#)
 - [UpdateFunctionCode](#)
 - [UpdateFunctionConfiguration](#)

無伺服器範例

使用 Kinesis 觸發條件調用 Lambda 函數

下列程式碼範例示範如何實作 Lambda 函數，該函數會接收從 Kinesis 串流接收記錄而觸發的事件。此函數會擷取 Kinesis 承載、從 Base64 解碼，並記錄記錄內容。

適用於 Ruby 的開發套件

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[無伺服器範例](#)儲存庫中設定和執行。

使用紅寶石使用 Lambda 消耗 Kinesis 事件。

```
require 'aws-sdk'

def lambda_handler(event:, context:)
  event['Records'].each do |record|
    begin
      puts "Processed Kinesis Event - EventID: #{record['eventID']}"
      record_data = get_record_data_async(record['kinesis'])
      puts "Record Data: #{record_data}"
      # TODO: Do interesting work based on the new data
    rescue => err
      $stderr.puts "An error occurred #{err}"
      raise err
    end
  end
  puts "Successfully processed #{event['Records'].length} records."
end

def get_record_data_async(payload)
  data = Base64.decode64(payload['data']).force_encoding('UTF-8')
  # Placeholder for actual async work
  # You can use Ruby's asynchronous programming tools like async/await or fibers
  here.
  return data
end
```

使用 Amazon SNS 觸發條件調用 Lambda 函數

下列程式碼範例示範如何實作 Lambda 函數，該函數會接收來自 SNS 主題的訊息而觸發的事件。函數會從事件參數擷取訊息，並記錄每一則訊息的內容。

適用於 Ruby 的開發套件

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[無伺服器範例](#)儲存庫中設定和執行。

使用紅寶石使用與 Lambda 的 SNS 事件。

```
def lambda_handler(event:, context:)  
  event['Records'].map { |record| process_message(record) }  
end  
  
def process_message(record)  
  message = record['Sns']['Message']  
  puts("Processing message: #{message}")  
rescue StandardError => e  
  puts("Error processing message: #{e}")  
  raise  
end
```

使用 Amazon SQS 觸發條件調用 Lambda 函數

下列程式碼範例示範如何實作 Lambda 函數，此函數會接收由 SQS 佇列接收訊息而觸發的事件。函數會從事件參數擷取訊息，並記錄每一則訊息的內容。

適用於 Ruby 的開發套件

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[無伺服器範例](#)儲存庫中設定和執行。

使用 Ruby 搭配 Lambda 來使用 SQS 事件。

```
def lambda_handler(event:, context:)
  event['Records'].each do |message|
    process_message(message)
  end
  puts "done"
end

def process_message(message)
  begin
    puts "Processed message #{message['body']}"
    # TODO: Do interesting work based on the new message
  rescue StandardError => err
    puts "An error occurred"
    raise err
  end
end
```

使用 Kinesis 觸發條件報告 Lambda 函數的批次項目失敗

下列程式碼範例顯示如何針對接收來自 Kinesis 串流之事件的 Lambda 函數實作部分批次回應。此函數會在回應中報告批次項目失敗，指示 Lambda 稍後重試這些訊息。

適用於 Ruby 的開發套件

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[無伺服器範例](#)儲存庫中設定和執行。

使用 Ruby 使用 Lambda 報告 Kinesis 批次項目失敗。

```
require 'aws-sdk'

def lambda_handler(event:, context:)
  batch_item_failures = []

  event['Records'].each do |record|
    begin
      puts "Processed Kinesis Event - EventID: #{record['eventID']}"
      record_data = get_record_data_async(record['kinesis'])
      puts "Record Data: #{record_data}"
    end
  end
end
```

```

    # TODO: Do interesting work based on the new data
    rescue StandardError => err
      puts "An error occurred #{err}"
      # Since we are working with streams, we can return the failed item
      immediately.
      # Lambda will immediately begin to retry processing from this failed item
      onwards.
      return { batchItemFailures: [{ itemIdentifier: record['kinesis']
['sequenceNumber'] }] }
    end
  end

  puts "Successfully processed #{event['Records'].length} records."
  { batchItemFailures: batch_item_failures }
end

def get_record_data_async(payload)
  data = Base64.decode64(payload['data']).force_encoding('utf-8')
  # Placeholder for actual async work
  sleep(1)
  data
end

```

使用 Amazon SQS 觸發條件報告 Lambda 函數的批次項目失敗

下列程式碼範例示範如何針對接收來自 SQS 佇列之事件的 Lambda 函數實作部分批次回應。此函數會在回應中報告批次項目失敗，指示 Lambda 稍後重試這些訊息。

適用於 Ruby 的開發套件

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[無伺服器範例](#)儲存庫中設定和執行。

使用 Ruby 搭配 Lambda 報告 SQS 批次項目失敗。

```

require 'json'

def lambda_handler(event:, context:)
  if event

```



```
batch_item_failures = []
sqs_batch_response = {}

event["Records"].each do |record|
  begin
    # process message
    rescue StandardError => e
      batch_item_failures << {"itemIdentifier" => record['messageId']}
    end
  end

  sqs_batch_response["batchItemFailures"] = batch_item_failures
  return sqs_batch_response
end
end
```

Amazon Polly 示例使用 SDK for Ruby

下列程式碼範例說明如何透過 AWS SDK for Ruby 搭配 Amazon Polly 使用來執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境和跨服務範例中查看內容中的動作。

案例是向您展示如何呼叫相同服務中的多個函數來完成特定任務的程式碼範例。

每個範例都包含一個連結 GitHub，您可以在其中找到如何在內容中設定和執程式碼的指示。

主題

- [動作](#)

動作

取得可用於合成的聲音

下面的代碼示例演示如何獲得 Amazon Polly 聲音可用於合成。

適用於 Ruby 的開發套件

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫](#)。

```
require "aws-sdk-polly" # In v2: require 'aws-sdk'

begin
  # Create an Amazon Polly client using
  # credentials from the shared credentials file ~/.aws/credentials
  # and the configuration (region) from the shared configuration file ~/.aws/config
  polly = Aws::Polly::Client.new

  # Get US English voices
  resp = polly.describe_voices(language_code: "en-US")

  resp.voices.each do |v|
    puts v.name
    puts "  " + v.gender
    puts
  end
rescue StandardError => ex
  puts "Could not get voices"
  puts "Error message:"
  puts ex.message
end
```

- 如需 API 詳細資訊，請參閱 AWS SDK for Ruby API 參考[DescribeVoices](#)中的。

列表發音詞典

下面的代碼示例演示了如何列出 Amazon Polly 發音詞典。

適用於 Ruby 的開發套件

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫](#)。

```
require "aws-sdk-polly" # In v2: require 'aws-sdk'

begin
  # Create an Amazon Polly client using
  # credentials from the shared credentials file ~/.aws/credentials
  # and the configuration (region) from the shared configuration file ~/.aws/config
  polly = Aws::Polly::Client.new

  resp = polly.list_lexicons


  resp.lexicons.each do |l|
    puts l.name
    puts "  Alphabet:" + l.attributes.alphabet
    puts "  Language:" + l.attributes.language
    puts
  end
rescue StandardError => ex
  puts "Could not get lexicons"
  puts "Error message:"
  puts ex.message
end
```

- 如需 API 詳細資訊，請參閱 AWS SDK for Ruby API 參考[ListLexicons](#)中的。

從文本合成語音

下列程式碼範例示範如何使用 Amazon Polly 合成文字中的語音。

適用於 Ruby 的開發套件

 Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫](#)。

```
require "aws-sdk-polly" # In v2: require 'aws-sdk'

begin
  # Get the filename from the command line
  if ARGV.empty?
    puts "You must supply a filename"
    exit 1
  end

  filename = ARGV[0]

  # Open file and get the contents as a string
  if File.exist?(filename)
    contents = IO.read(filename)
  else
    puts "No such file: " + filename
    exit 1
  end

  # Create an Amazon Polly client using
  # credentials from the shared credentials file ~/.aws/credentials
  # and the configuration (region) from the shared configuration file ~/.aws/config
  polly = Aws::Polly::Client.new

  resp = polly.synthesize_speech({
    output_format: "mp3",
    text: contents,
    voice_id: "Joanna",
  })

  # Save output
  # Get just the file name
  # abc/xyz.txt -> xyx.txt
  name = File.basename(filename)
```

```
# Split up name so we get just the xyz part
parts = name.split(".")
first_part = parts[0]
mp3_file = first_part + ".mp3"

IO.copy_stream(resp.audio_stream, mp3_file)

puts "Wrote MP3 content to: " + mp3_file
rescue StandardError => ex
  puts "Got error:"
  puts "Error message:"
  puts ex.message
end
```

- 如需 API 詳細資訊，請參閱 AWS SDK for Ruby API 參考[SynthesizeSpeech](#)中的。

Amazon RDS 示例使用 SDK for Ruby

下列程式碼範例說明如何透過 AWS SDK for Ruby 搭配 Amazon RDS 使用來執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境和跨服務範例中查看內容中的動作。

案例是向您展示如何呼叫相同服務中的多個函數來完成特定任務的程式碼範例。

每個範例都包含一個連結 GitHub，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題


- [動作](#)

動作

建立資料庫執行個體的快照

下列程式碼範例顯示如何建立 Amazon RDS 資料庫執行個體的快照。

適用於 Ruby 的開發套件

 Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫](#)。

```
require "aws-sdk-rds" # v2: require 'aws-sdk'


# Create a snapshot for an Amazon Relational Database Service (Amazon RDS)
# DB instance.
#
# @param rds_resource [Aws::RDS::Resource] The resource containing SDK logic.
# @param db_instance_name [String] The name of the Amazon RDS DB instance.
# @return [Aws::RDS::DBSnapshot, nil] The snapshot created, or nil if error.
def create_snapshot(rds_resource, db_instance_name)
  id = "snapshot-#{rand(10**6)}"
  db_instance = rds_resource.db_instance(db_instance_name)
  db_instance.create_snapshot({
    db_snapshot_identifier: id
  })
rescue Aws::Errors::ServiceError => e
  puts "Couldn't create DB instance snapshot #{id}:\n #{e.message}"
end
```

- 如需 API 詳細資訊，請參閱《AWS SDK for Ruby API 參考》中的 [CreateDBSnapshot](#)。

描述資料庫執行個體

下列程式碼範例顯示如何描述 Amazon RDS 資料庫執行個體。

適用於 Ruby 的開發套件

 Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫](#)。

```
require "aws-sdk-rds" # v2: require 'aws-sdk'
```

```
# List all Amazon Relational Database Service (Amazon RDS) DB instances.
#
# @param rds_resource [Aws::RDS::Resource] An SDK for Ruby Amazon RDS resource.
# @return [Array, nil] List of all DB instances, or nil if error.
def list_instances(rds_resource)
  db_instances = []
  rds_resource.db_instances.each do |i|
    db_instances.append({
      "name": i.id,
      "status": i.db_instance_status
    })
  end
  db_instances
rescue Aws::Errors::ServiceError => e
  puts "Couldn't list instances:\n#{e.message}"
end
```

- 如需 API 詳細資訊，請參閱《AWS SDK for Ruby API 參考》中的 [DescribeDBInstances](#)。

描述資料庫參數群組

下列程式碼範例顯示如何描述 Amazon RDS 資料庫參數群組。

適用於 Ruby 的開發套件

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在 [AWS 設定和執行程式碼範例儲存庫](#)。

```
require "aws-sdk-rds" # v2: require 'aws-sdk'

# List all Amazon Relational Database Service (Amazon RDS) parameter groups.
#
# @param rds_resource [Aws::RDS::Resource] An SDK for Ruby Amazon RDS resource.
# @return [Array, nil] List of all parameter groups, or nil if error.
def list_parameter_groups(rds_resource)
  parameter_groups = []
  rds_resource.db_parameter_groups.each do |p|
    parameter_groups.append({
      "name": p.db_parameter_group_name,
```

```

        "description": p.description
      })
    end
    parameter_groups
  rescue Aws::Errors::ServiceError => e
    puts "Couldn't list parameter groups:\n #{e.message}"
  end
end

```

- 如需 API 詳細資訊，請參閱 AWS SDK for Ruby API 參考資料 [ParameterGroups](#) 中的說明 B。

描述資料庫參數群組中的參數

下列程式碼範例顯示如何描述 Amazon RDS 資料庫參數群組中的參數。

適用於 Ruby 的開發套件

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在 [AWS 設定和執行程式碼範例儲存庫](#)。

```

require "aws-sdk-rds" # v2: require 'aws-sdk'

# List all Amazon Relational Database Service (Amazon RDS) parameter groups.
#
# @param rds_resource [Aws::RDS::Resource] An SDK for Ruby Amazon RDS resource.
# @return [Array, nil] List of all parameter groups, or nil if error.
def list_parameter_groups(rds_resource)
  parameter_groups = []
  rds_resource.db_parameter_groups.each do |p|
    parameter_groups.append({
      "name": p.db_parameter_group_name,
      "description": p.description
    })
  end
  parameter_groups
rescue Aws::Errors::ServiceError => e
  puts "Couldn't list parameter groups:\n #{e.message}"
end

```


- 如需 API 詳細資訊，請參閱《AWS SDK for Ruby API 參考》中的 [DescribeDBParameters](#)。

描述資料庫執行個體的快照

下列程式碼範例顯示如何描述 Amazon RDS 資料庫執行個體的快照。

適用於 Ruby 的開發套件

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在 [AWS 設定和執行程式碼範例儲存庫](#)。

```
require "aws-sdk-rds" # v2: require 'aws-sdk'

# List all Amazon Relational Database Service (Amazon RDS) DB instance
# snapshots.
#
# @param rds_resource [Aws::RDS::Resource] An SDK for Ruby Amazon RDS resource.
# @return instance_snapshots [Array, nil] All instance snapshots, or nil if error.
def list_instance_snapshots(rds_resource)
  instance_snapshots = []
  rds_resource.db_snapshots.each do |s|
    instance_snapshots.append({
      "id": s.snapshot_id,
      "status": s.status
    })
  end
  instance_snapshots
rescue Aws::Errors::ServiceError => e
  puts "Couldn't list instance snapshots:\n #{e.message}"
end
```

- 如需 API 詳細資訊，請參閱《AWS SDK for Ruby API 參考》中的 [DescribeDBSnapshots](#)。

Amazon S3 示例使用 SDK for Ruby

下列程式碼範例說明如何透過 AWS SDK for Ruby 搭配 Amazon S3 使用來執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境和跨服務範例中查看內容中的動作。

案例是向您展示如何呼叫相同服務中的多個函數來完成特定任務的程式碼範例。

每個範例都包含一個連結 GitHub，您可以在其中找到如何在內容中設定和執程式碼的指示。

主題

- [動作](#)
- [案例](#)

動作

將 CORS 規則新增至儲存貯體

下列程式碼範例顯示如何將跨來源資源共用 (CORS) 規則新增至 S3 儲存貯體。

適用於 Ruby 的開發套件

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執程式碼範例儲存庫](#)。

```
require "aws-sdk-s3"

# Wraps Amazon S3 bucket CORS configuration.
class BucketCorsWrapper
  attr_reader :bucket_cors

  # @param bucket_cors [Aws::S3::BucketCors] A bucket CORS object configured with an
  # existing bucket.
  def initialize(bucket_cors)
    @bucket_cors = bucket_cors
  end

  # Sets CORS rules on a bucket.
  #
  # @param allowed_methods [Array<String>] The types of HTTP requests to allow.
  # @param allowed_origins [Array<String>] The origins to allow.
  # @returns [Boolean] True if the CORS rules were set; otherwise, false.
```

```

def set_cors(allowed_methods, allowed_origins)
  @bucket_cors.put(
    cors_configuration: {
      cors_rules: [
        {
          allowed_methods: allowed_methods,
          allowed_origins: allowed_origins,
          allowed_headers: %w[*],
          max_age_seconds: 3600
        }
      ]
    }
  )
  true
rescue Aws::Errors::ServiceError => e
  puts "Couldn't set CORS rules for #{@bucket_cors.bucket.name}. Here's why:
#{e.message}"
  false
end
end

```

- 如需 API 詳細資訊，請參閱 AWS SDK for Ruby API 參考[PutBucketCors](#)中的。

新增儲存貯體政策

下列程式碼範例顯示如何將政策新增至 S3 儲存貯體。

適用於 Ruby 的開發套件

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫](#)。

```

# Wraps an Amazon S3 bucket policy.
class BucketPolicyWrapper
  attr_reader :bucket_policy

  # @param bucket_policy [Aws::S3::BucketPolicy] A bucket policy object configured
  with an existing bucket.

```



```

def initialize(source_object)
  @source_object = source_object
end

# Copy the source object to the specified target bucket and rename it with the
target key.
#
# @param target_bucket [Aws::S3::Bucket] An existing Amazon S3 bucket where the
object is copied.
# @param target_object_key [String] The key to give the copy of the object.
# @return [Aws::S3::Object, nil] The copied object when successful; otherwise,
nil.
def copy_object(target_bucket, target_object_key)
  @source_object.copy_to(bucket: target_bucket.name, key: target_object_key)
  target_bucket.object(target_object_key)
rescue Aws::Errors::ServiceError => e
  puts "Couldn't copy #{@source_object.key} to #{target_object_key}. Here's why:
#{e.message}"
end

# Example usage:
def run_demo
  source_bucket_name = "doc-example-bucket1"
  source_key = "my-source-file.txt"
  target_bucket_name = "doc-example-bucket2"
  target_key = "my-target-file.txt"

  source_bucket = Aws::S3::Bucket.new(source_bucket_name)
  wrapper = ObjectCopyWrapper.new(source_bucket.object(source_key))
  target_bucket = Aws::S3::Bucket.new(target_bucket_name)
  target_object = wrapper.copy_object(target_bucket, target_key)
  return unless target_object

  puts "Copied #{source_key} from #{source_bucket_name} to
#{target_object.bucket_name}:#{target_object.key}."
end

run_demo if $PROGRAM_NAME == __FILE__

```

複製物件，然後將伺服器端加密新增至目標物件。

```
require "aws-sdk-s3"

# Wraps Amazon S3 object actions.
class ObjectCopyEncryptWrapper
  attr_reader :source_object

  # @param source_object [Aws::S3::Object] An existing Amazon S3 object. This is
  # used as the source object for
  #                               copy actions.
  def initialize(source_object)
    @source_object = source_object
  end

  # Copy the source object to the specified target bucket, rename it with the target
  # key, and encrypt it.
  # @param target_bucket [Aws::S3::Bucket] An existing Amazon S3 bucket where the
  # object is copied.
  # @param target_object_key [String] The key to give the copy of the object.
  # @return [Aws::S3::Object, nil] The copied object when successful; otherwise,
  # nil.
  def copy_object(target_bucket, target_object_key, encryption)
    @source_object.copy_to(bucket: target_bucket.name, key: target_object_key,
server_side_encryption: encryption)
    target_bucket.object(target_object_key)
  rescue Aws::Errors::ServiceError => e
    puts "Couldn't copy #{@source_object.key} to #{target_object_key}. Here's why:
#{e.message}"
  end
end

# Example usage:
def run_demo
  source_bucket_name = "doc-example-bucket1"
  source_key = "my-source-file.txt"
  target_bucket_name = "doc-example-bucket2"
  target_key = "my-target-file.txt"
  target_encryption = "AES256"

  source_bucket = Aws::S3::Bucket.new(source_bucket_name)
  wrapper = ObjectCopyEncryptWrapper.new(source_bucket.object(source_key))
  target_bucket = Aws::S3::Bucket.new(target_bucket_name)
  target_object = wrapper.copy_object(target_bucket, target_key, target_encryption)
end
```

```

return unless target_object

puts "Copied #{source_key} from #{source_bucket_name} to
#{target_object.bucket_name}:#{target_object.key} and "\
    "encrypted the target with #{target_object.server_side_encryption}
encryption."
end

run_demo if $PROGRAM_NAME == __FILE__

```

- 如需 API 詳細資訊，請參閱 AWS SDK for Ruby API 參考[CopyObject](#)中的。

建立 儲存貯體

下列程式碼範例示範如何建立 S3 儲存貯體。

適用於 Ruby 的開發套件

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫](#)。

```

require "aws-sdk-s3"

# Wraps Amazon S3 bucket actions.
class BucketCreateWrapper
  attr_reader :bucket

  # @param bucket [Aws::S3::Bucket] An Amazon S3 bucket initialized with a name.
  # This is a client-side object until
  # create is called.
  def initialize(bucket)
    @bucket = bucket
  end

  # Creates an Amazon S3 bucket in the specified AWS Region.
  #
  # @param region [String] The Region where the bucket is created.
  # @return [Boolean] True when the bucket is created; otherwise, false.
  def create?(region)

```

```

    @bucket.create(create_bucket_configuration: { location_constraint: region })
    true
  rescue Aws::Errors::ServiceError => e
    puts "Couldn't create bucket. Here's why: #{e.message}"
    false
  end

  # Gets the Region where the bucket is located.
  #
  # @return [String] The location of the bucket.
  def location
    if @bucket.nil?
      "None. You must create a bucket before you can get its location!"
    else
      @bucket.client.get_bucket_location(bucket: @bucket.name).location_constraint
    end
  rescue Aws::Errors::ServiceError => e
    "Couldn't get the location of #{@bucket.name}. Here's why: #{e.message}"
  end
end

# Example usage:
def run_demo
  region = "us-west-2"
  wrapper = BucketCreateWrapper.new(Aws::S3::Bucket.new("doc-example-bucket-
#{Random.uuid}"))
  return unless wrapper.create?(region)

  puts "Created bucket #{wrapper.bucket.name}."
  puts "Your bucket's region is: #{wrapper.location}"
end

run_demo if $PROGRAM_NAME == __FILE__

```

- 如需 API 詳細資訊，請參閱 AWS SDK for Ruby API 參考[CreateBucket](#)中的。

從儲存貯體刪除 CORS 規則

下列程式碼範例顯示如何從 S3 儲存貯體刪除 CORS 規則。

適用於 Ruby 的開發套件

 Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫](#)。

```
require "aws-sdk-s3"

# Wraps Amazon S3 bucket CORS configuration.
class BucketCorsWrapper
  attr_reader :bucket_cors

  # @param bucket_cors [Aws::S3::BucketCors] A bucket CORS object configured with an
  # existing bucket.
  def initialize(bucket_cors)
    @bucket_cors = bucket_cors
  end

  # Deletes the CORS configuration of a bucket.
  #
  # @return [Boolean] True if the CORS rules were deleted; otherwise, false.
  def delete_cors
    @bucket_cors.delete
    true
  rescue Aws::Errors::ServiceError => e
    puts "Couldn't delete CORS rules for #{@bucket_cors.bucket.name}. Here's why:
    #{e.message}"
    false
  end
end
```

- 如需 API 詳細資訊，請參閱 AWS SDK for Ruby API 參考[DeleteBucketCors](#)中的。

從儲存貯體刪除政策

下列程式碼範例顯示如何從 S3 儲存貯體刪除政策。

適用於 Ruby 的開發套件

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫](#)。

```
# Wraps an Amazon S3 bucket policy.
class BucketPolicyWrapper
  attr_reader :bucket_policy

  # @param bucket_policy [Aws::S3::BucketPolicy] A bucket policy object configured
  # with an existing bucket.
  def initialize(bucket_policy)
    @bucket_policy = bucket_policy
  end

  def delete_policy
    @bucket_policy.delete
    true
  rescue Aws::Errors::ServiceError => e
    puts "Couldn't delete the policy from #{@bucket_policy.bucket.name}. Here's why:
    #{e.message}"
    false
  end
end

end
```

- 如需 API 詳細資訊，請參閱 AWS SDK for Ruby API 參考[DeleteBucketPolicy](#)中的。

刪除空的儲存貯體

下列程式碼範例示範如何刪除空的 S3 儲存貯體。

適用於 Ruby 的開發套件

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫](#)。

```
# Deletes the objects in an Amazon S3 bucket and deletes the bucket.
#
# @param bucket [Aws::S3::Bucket] The bucket to empty and delete.
def delete_bucket(bucket)
  puts("\nDo you want to delete all of the objects as well as the bucket (y/n)? ")
  answer = gets.chomp.downcase
  if answer == "y"
    bucket.objects.batch_delete!
    bucket.delete
    puts("Emptied and deleted bucket #{bucket.name}.\n")
  end
rescue Aws::Errors::ServiceError => e
  puts("Couldn't empty and delete bucket #{bucket.name}.")
  puts("\t#{e.code}: #{e.message}")
  raise
end
```

- 如需 API 詳細資訊，請參閱 AWS SDK for Ruby API 參考 [DeleteBucket](#) 中的。

刪除多個物件

下列程式碼範例示範如何從 S3 儲存貯體中刪除多個物件。

適用於 Ruby 的開發套件

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在 [AWS 設定和執行程式碼範例儲存庫](#)。

```
# Deletes the objects in an Amazon S3 bucket and deletes the bucket.
#
# @param bucket [Aws::S3::Bucket] The bucket to empty and delete.
def delete_bucket(bucket)
  puts("\nDo you want to delete all of the objects as well as the bucket (y/n)? ")
  answer = gets.chomp.downcase
  if answer == "y"
    bucket.objects.batch_delete!
    bucket.delete
    puts("Emptied and deleted bucket #{bucket.name}.\n")
  end
end
```

```
end
rescue Aws::Errors::ServiceError => e
  puts("Couldn't empty and delete bucket #{bucket.name}.")
  puts("\t#{e.code}: #{e.message}")
  raise
end
```

- 如需 API 詳細資訊，請參閱 AWS SDK for Ruby API 參考 [DeleteObjects](#) 中的。

判斷物件是否存在和內容類型

下列程式碼範例顯示如何判斷 S3 儲存貯體中物件的存在和內容類型。

適用於 Ruby 的開發套件

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在 [AWS 設定和執行程式碼範例儲存庫](#)。

```
require "aws-sdk-s3"

# Wraps Amazon S3 object actions.
class ObjectExistsWrapper
  attr_reader :object

  # @param object [Aws::S3::Object] An Amazon S3 object.
  def initialize(object)
    @object = object
  end

  # Checks whether the object exists.
  #
  # @return [Boolean] True if the object exists; otherwise false.
  def exists?
    @object.exists?
  rescue Aws::Errors::ServiceError => e
    puts "Couldn't check existence of object #{@object.bucket.name}:#{@object.key}.
    Here's why: #{e.message}"
    false
  end
end
```

```

end

# Example usage:
def run_demo
  bucket_name = "doc-example-bucket"
  object_key = "my-object.txt"

  wrapper = ObjectExistsWrapper.new(Aws::S3::Object.new(bucket_name, object_key))
  exists = wrapper.exists?

  puts "Object #{object_key} #{exists ? 'does' : 'does not'} exist."
end

run_demo if $PROGRAM_NAME == __FILE__

```

- 如需 API 詳細資訊，請參閱 AWS SDK for Ruby API 參考[HeadObject](#)中的。

取得儲存貯體的 CORS 規則

下列程式碼範例顯示如何取得 S3 儲存貯體的跨來源資源共用 (CORS) 規則。

適用於 Ruby 的開發套件

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫](#)。

```

require "aws-sdk-s3"

# Wraps Amazon S3 bucket CORS configuration.
class BucketCorsWrapper
  attr_reader :bucket_cors

  # @param bucket_cors [Aws::S3::BucketCors] A bucket CORS object configured with an
  # existing bucket.
  def initialize(bucket_cors)
    @bucket_cors = bucket_cors
  end

  # Gets the CORS configuration of a bucket.

```

```
#
# @return [Aws::S3::Type::GetBucketCorsOutput, nil] The current CORS configuration
for the bucket.
def get_cors
  @bucket_cors.data
rescue Aws::Errors::ServiceError => e
  puts "Couldn't get CORS configuration for #{@bucket_cors.bucket.name}. Here's
why: #{e.message}"
  nil
end
end
```

- 如需 API 詳細資訊，請參閱 AWS SDK for Ruby API 參考[GetBucketCors](#)中的。

取得儲存貯體的物件

下列程式碼範例示範如何從 S3 儲存貯體中讀取物件的資料。

適用於 Ruby 的開發套件

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫](#)。

取得物件。

```
require "aws-sdk-s3"

# Wraps Amazon S3 object actions.
class ObjectGetWrapper
  attr_reader :object

  # @param object [Aws::S3::Object] An existing Amazon S3 object.
  def initialize(object)
    @object = object
  end

  # Gets the object directly to a file.
  #
```

```

# @param target_path [String] The path to the file where the object is downloaded.
# @return [Aws::S3::Types::GetObjectOutput, nil] The retrieved object data if
successful; otherwise nil.
def get_object(target_path)
  @object.get(response_target: target_path)
rescue Aws::Errors::ServiceError => e
  puts "Couldn't get object #{@object.key}. Here's why: #{e.message}"
end
end

# Example usage:
def run_demo
  bucket_name = "doc-example-bucket"
  object_key = "my-object.txt"
  target_path = "my-object-as-file.txt"

  wrapper = ObjectGetWrapper.new(Aws::S3::Object.new(bucket_name, object_key))
  obj_data = wrapper.get_object(target_path)
  return unless obj_data

  puts "Object #{object_key} (#{obj_data.content_length} bytes) downloaded to
#{target_path}."
end

run_demo if $PROGRAM_NAME == __FILE__

```

取得物件並報告其伺服器端的加密狀態。

```

require "aws-sdk-s3"

# Wraps Amazon S3 object actions.
class ObjectGetEncryptionWrapper
  attr_reader :object

  # @param object [Aws::S3::Object] An existing Amazon S3 object.
  def initialize(object)
    @object = object
  end

  # Gets the object into memory.
  #

```

```

# @return [Aws::S3::Types::GetObjectOutput, nil] The retrieved object data if
successful; otherwise nil.
def get_object
  @object.get
  rescue Aws::Errors::ServiceError => e
    puts "Couldn't get object #{@object.key}. Here's why: #{e.message}"
  end
end

# Example usage:
def run_demo
  bucket_name = "doc-example-bucket"
  object_key = "my-object.txt"

  wrapper = ObjectGetEncryptionWrapper.new(Aws::S3::Object.new(bucket_name,
object_key))
  obj_data = wrapper.get_object
  return unless obj_data

  encryption = obj_data.server_side_encryption.nil? ? "no" :
obj_data.server_side_encryption
  puts "Object #{object_key} uses #{encryption} encryption."
end

run_demo if $PROGRAM_NAME == __FILE__

```

- 如需 API 詳細資訊，請參閱 AWS SDK for Ruby API 參考[GetObject](#)中的。

取得儲存貯體的政策

下列程式碼範例顯示如何取得 S3 儲存貯體的政策。

適用於 Ruby 的開發套件

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫](#)。

```

# Wraps an Amazon S3 bucket policy.
class BucketPolicyWrapper

```



```
attr_reader :bucket_policy

# @param bucket_policy [Aws::S3::BucketPolicy] A bucket policy object configured
with an existing bucket.
def initialize(bucket_policy)
  @bucket_policy = bucket_policy
end

# Gets the policy of a bucket.
#
# @return [Aws::S3::GetBucketPolicyOutput, nil] The current bucket policy.
def get_policy
  policy = @bucket_policy.data.policy
  policy.respond_to?(:read) ? policy.read : policy
rescue Aws::Errors::ServiceError => e
  puts "Couldn't get the policy for #{@bucket_policy.bucket.name}. Here's why:
#{e.message}"
  nil
end

end
```

- 如需 API 詳細資訊，請參閱 AWS SDK for Ruby API 參考[GetBucketPolicy](#)中的。

列出儲存貯體

下列程式碼範例顯示如何列出 S3 儲存貯體。

適用於 Ruby 的開發套件

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫](#)。

```
require "aws-sdk-s3"

# Wraps Amazon S3 resource actions.
class BucketListWrapper
  attr_reader :s3_resource
```

```
# @param s3_resource [Aws::S3::Resource] An Amazon S3 resource.
def initialize(s3_resource)
  @s3_resource = s3_resource
end

# Lists buckets for the current account.
#
# @param count [Integer] The maximum number of buckets to list.
def list_buckets(count)
  puts "Found these buckets:"
  @s3_resource.buckets.each do |bucket|
    puts "\t#{bucket.name}"
    count -= 1
    break if count.zero?
  end
  true
rescue Aws::Errors::ServiceError => e
  puts "Couldn't list buckets. Here's why: #{e.message}"
  false
end

# Example usage:
def run_demo
  wrapper = BucketListWrapper.new(Aws::S3::Resource.new)
  wrapper.list_buckets(25)
end

run_demo if $PROGRAM_NAME == __FILE__
```

- 如需 API 詳細資訊，請參閱 AWS SDK for Ruby API 參考[ListBuckets](#)中的。

列出儲存貯體中的物件

下列程式碼範例示範如何列出 S3 儲存貯體中的物件。

適用於 Ruby 的開發套件

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫](#)。

```
require "aws-sdk-s3"

# Wraps Amazon S3 bucket actions.
class BucketListObjectsWrapper
  attr_reader :bucket

  # @param bucket [Aws::S3::Bucket] An existing Amazon S3 bucket.
  def initialize(bucket)
    @bucket = bucket
  end

  # Lists object in a bucket.
  #
  # @param max_objects [Integer] The maximum number of objects to list.
  # @return [Integer] The number of objects listed.
  def list_objects(max_objects)
    count = 0
    puts "The objects in #{@bucket.name} are:"
    @bucket.objects.each do |obj|
      puts "\t#{obj.key}"
      count += 1
      break if count == max_objects
    end
    count
  rescue Aws::Errors::ServiceError => e
    puts "Couldn't list objects in bucket #{bucket.name}. Here's why: #{e.message}"
    0
  end
end

# Example usage:
def run_demo
  bucket_name = "doc-example-bucket"

  wrapper = BucketListObjectsWrapper.new(Aws::S3::Bucket.new(bucket_name))
  count = wrapper.list_objects(25)
  puts "Listed #{count} objects."
end

run_demo if $PROGRAM_NAME == __FILE__
```

- 有關 API 的詳細信息，請參閱 AWS SDK for Ruby API 參考中的 [ListObjectsV2](#)。

設定儲存貯體網站組態

下列程式碼範例顯示如何設定 S3 儲存貯體的網站組態。

適用於 Ruby 的開發套件

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫](#)。

```
require "aws-sdk-s3"

# Wraps Amazon S3 bucket website actions.
class BucketWebsiteWrapper
  attr_reader :bucket_website

  # @param bucket_website [Aws::S3::BucketWebsite] A bucket website object
  # configured with an existing bucket.
  def initialize(bucket_website)
    @bucket_website = bucket_website
  end

  # Sets a bucket as a static website.
  #
  # @param index_document [String] The name of the index document for the website.
  # @param error_document [String] The name of the error document to show for 4XX
  # errors.
  # @return [Boolean] True when the bucket is configured as a website; otherwise,
  # false.
  def set_website(index_document, error_document)
    @bucket_website.put(
      website_configuration: {
        index_document: { suffix: index_document },
        error_document: { key: error_document }
      }
    )
    true
  rescue Aws::Errors::ServiceError => e
    puts "Couldn't configure #{@bucket_website.bucket.name} as a website. Here's
  why: #{e.message}"
    false
  end
end
```

```
end

# Example usage:
def run_demo
  bucket_name = "doc-example-bucket"
  index_document = "index.html"
  error_document = "404.html"

  wrapper = BucketWebsiteWrapper.new(Aws::S3::BucketWebsite.new(bucket_name))
  return unless wrapper.set_website(index_document, error_document)

  puts "Successfully configured bucket #{bucket_name} as a static website."
end

run_demo if $PROGRAM_NAME == __FILE__
```

- 如需 API 詳細資訊，請參閱 AWS SDK for Ruby API 參考[PutBucketWebsite](#)中的。

將物件上傳至儲存貯體

下列程式碼範例示範如何將物件上傳至 S3 儲存貯體。

適用於 Ruby 的開發套件

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫](#)。

使用受管的上傳工具上傳檔案 (Object.upload_file)。

```
require "aws-sdk-s3"

# Wraps Amazon S3 object actions.
class ObjectUploadFileWrapper
  attr_reader :object

  # @param object [Aws::S3::Object] An existing Amazon S3 object.
  def initialize(object)
    @object = object
  end
end
```

```

# Uploads a file to an Amazon S3 object by using a managed uploader.
#
# @param file_path [String] The path to the file to upload.
# @return [Boolean] True when the file is uploaded; otherwise false.
def upload_file(file_path)
  @object.upload_file(file_path)
  true
rescue Aws::Errors::ServiceError => e
  puts "Couldn't upload file #{file_path} to #{@object.key}. Here's why:
#{e.message}"
  false
end
end

# Example usage:
def run_demo
  bucket_name = "doc-example-bucket"
  object_key = "my-uploaded-file"
  file_path = "object_upload_file.rb"

  wrapper = ObjectUploadFileWrapper.new(Aws::S3::Object.new(bucket_name,
object_key))
  return unless wrapper.upload_file(file_path)

  puts "File #{file_path} successfully uploaded to #{bucket_name}:#{object_key}."
end

run_demo if $PROGRAM_NAME == __FILE__

```

使用 `Object.put` 上傳檔案。

```

require "aws-sdk-s3"

# Wraps Amazon S3 object actions.
class ObjectPutWrapper
  attr_reader :object

  # @param object [Aws::S3::Object] An existing Amazon S3 object.
  def initialize(object)
    @object = object
  end
end

```

```

def put_object(source_file_path)
  File.open(source_file_path, "rb") do |file|
    @object.put(body: file)
  end
  true
rescue Aws::Errors::ServiceError => e
  puts "Couldn't put #{source_file_path} to #{@object.key}. Here's why:
#{e.message}"
  false
end
end

# Example usage:
def run_demo
  bucket_name = "doc-example-bucket"
  object_key = "my-object-key"
  file_path = "my-local-file.txt"

  wrapper = ObjectPutWrapper.new(Aws::S3::Object.new(bucket_name, object_key))
  success = wrapper.put_object(file_path)
  return unless success

  puts "Put file #{file_path} into #{object_key} in #{bucket_name}."
end

run_demo if $PROGRAM_NAME == __FILE__

```

使用 Object.Pet 上傳檔案並新增伺服器端加密。

```

require "aws-sdk-s3"

# Wraps Amazon S3 object actions.
class ObjectPutSseWrapper
  attr_reader :object

  # @param object [Aws::S3::Object] An existing Amazon S3 object.
  def initialize(object)
    @object = object
  end

  def put_object_encrypted(object_content, encryption)

```

```
@object.put(body: object_content, server_side_encryption: encryption)
  true
rescue Aws::Errors::ServiceError => e
  puts "Couldn't put your content to #{object.key}. Here's why: #{e.message}"
  false
end
end

# Example usage:
def run_demo
  bucket_name = "doc-example-bucket"
  object_key = "my-encrypted-content"
  object_content = "This is my super-secret content."
  encryption = "AES256"

  wrapper = ObjectPutSseWrapper.new(Aws::S3::Object.new(bucket_name,
object_content))
  return unless wrapper.put_object_encrypted(object_content, encryption)

  puts "Put your content into #{bucket_name}:#{object_key} and encrypted it with
#{encryption}."
end

run_demo if $PROGRAM_NAME == __FILE__
```

- 如需 API 詳細資訊，請參閱 AWS SDK for Ruby API 參考[PutObject](#)中的。

案例

建立預先簽章 URL

下列程式碼範例示範如何為 Amazon S3 建立預先簽署的 URL 並上傳物件。

適用於 Ruby 的開發套件

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫](#)。

```
require "aws-sdk-s3"
```



```
require "net/http"

# Creates a presigned URL that can be used to upload content to an object.
#
# @param bucket [Aws::S3::Bucket] An existing Amazon S3 bucket.
# @param object_key [String] The key to give the uploaded object.
# @return [URI, nil] The parsed URI if successful; otherwise nil.
def get_presigned_url(bucket, object_key)
  url = bucket.object(object_key).presigned_url(:put)
  puts "Created presigned URL: #{url}"
  URI(url)
rescue Aws::Errors::ServiceError => e
  puts "Couldn't create presigned URL for #{bucket.name}:#{object_key}. Here's why:
#{e.message}"
end

# Example usage:
def run_demo
  bucket_name = "doc-example-bucket"
  object_key = "my-file.txt"
  object_content = "This is the content of my-file.txt."

  bucket = Aws::S3::Bucket.new(bucket_name)
  presigned_url = get_presigned_url(bucket, object_key)
  return unless presigned_url

  response = Net::HTTP.start(presigned_url.host) do |http|
    http.send_request("PUT", presigned_url.request_uri, object_content,
"content_type" => "")
  end

  case response
  when Net::HTTPSuccess
    puts "Content uploaded!"
  else
    puts response.value
  end
end

run_demo if $PROGRAM_NAME == __FILE__
```

開始使用儲存貯體和物件

以下程式碼範例顯示做法：

- 建立儲存貯體並上傳檔案到該儲存貯體。
- 從儲存貯體下載物件。
- 將物件複製至儲存貯體中的子文件夾。
- 列出儲存貯體中的物件。
- 刪除儲存貯體物件和該儲存貯體。

適用於 Ruby 的開發套件

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫](#)。

```
require "aws-sdk-s3"

# Wraps the getting started scenario actions.
class ScenarioGettingStarted
  attr_reader :s3_resource

  # @param s3_resource [Aws::S3::Resource] An Amazon S3 resource.
  def initialize(s3_resource)
    @s3_resource = s3_resource
  end

  # Creates a bucket with a random name in the currently configured account and
  # AWS Region.
  #
  # @return [Aws::S3::Bucket] The newly created bucket.
  def create_bucket
    bucket = @s3_resource.create_bucket(
      bucket: "doc-example-bucket-#{Random.uuid}",
      create_bucket_configuration: {
        location_constraint: "us-east-1" # Note: only certain regions permitted
      }
    )
    puts("Created demo bucket named #{bucket.name}.")
  end
end
```

```
rescue Aws::Errors::ServiceError => e
  puts("Tried and failed to create demo bucket.")
  puts("\t#{e.code}: #{e.message}")
  puts("\nCan't continue the demo without a bucket!")
  raise
else
  bucket
end

# Requests a file name from the user.
#
# @return The name of the file.
def create_file
  File.open("demo.txt", w) { |f| f.write("This is a demo file.") }
end

# Uploads a file to an Amazon S3 bucket.
#
# @param bucket [Aws::S3::Bucket] The bucket object representing the upload
destination
# @return [Aws::S3::Object] The Amazon S3 object that contains the uploaded file.
def upload_file(bucket)
  File.open("demo.txt", "w+") { |f| f.write("This is a demo file.") }
  s3_object = bucket.object(File.basename("demo.txt"))
  s3_object.upload_file("demo.txt")
  puts("Uploaded file demo.txt into bucket #{bucket.name} with key
#{s3_object.key}.")
rescue Aws::Errors::ServiceError => e
  puts("Couldn't upload file demo.txt to #{bucket.name}.")
  puts("\t#{e.code}: #{e.message}")
  raise
else
  s3_object
end

# Downloads an Amazon S3 object to a file.
#
# @param s3_object [Aws::S3::Object] The object to download.
def download_file(s3_object)
  puts("\nDo you want to download #{s3_object.key} to a local file (y/n)? ")
  answer = gets.chomp.downcase
  if answer == "y"
    puts("Enter a name for the downloaded file: ")
    file_name = gets.chomp
```

```
s3_object.download_file(file_name)
puts("Object #{s3_object.key} successfully downloaded to #{file_name}.")
end
rescue Aws::Errors::ServiceError => e
  puts("Couldn't download #{s3_object.key}.")
  puts("\t#{e.code}: #{e.message}")
  raise
end

# Copies an Amazon S3 object to a subfolder within the same bucket.
#
# @param source_object [Aws::S3::Object] The source object to copy.
# @return [Aws::S3::Object, nil] The destination object.
def copy_object(source_object)
  dest_object = nil
  puts("\nDo you want to copy #{source_object.key} to a subfolder in your bucket
(y/n)? ")
  answer = gets.chomp.downcase
  if answer == "y"
    dest_object = source_object.bucket.object("demo-folder/#{source_object.key}")
    dest_object.copy_from(source_object)
    puts("Copied #{source_object.key} to #{dest_object.key}.")
  end
rescue Aws::Errors::ServiceError => e
  puts("Couldn't copy #{source_object.key}.")
  puts("\t#{e.code}: #{e.message}")
  raise
else
  dest_object
end

# Lists the objects in an Amazon S3 bucket.
#
# @param bucket [Aws::S3::Bucket] The bucket to query.
def list_objects(bucket)
  puts("\nYour bucket contains the following objects:")
  bucket.objects.each do |obj|
    puts("\t#{obj.key}")
  end
rescue Aws::Errors::ServiceError => e
  puts("Couldn't list the objects in bucket #{bucket.name}.")
  puts("\t#{e.code}: #{e.message}")
  raise
end
```

```
# Deletes the objects in an Amazon S3 bucket and deletes the bucket.
#
# @param bucket [Aws::S3::Bucket] The bucket to empty and delete.
def delete_bucket(bucket)
  puts("\nDo you want to delete all of the objects as well as the bucket (y/n)? ")
  answer = gets.chomp.downcase
  if answer == "y"
    bucket.objects.batch_delete!
    bucket.delete
    puts("Emptied and deleted bucket #{bucket.name}.\n")
  end
rescue Aws::Errors::ServiceError => e
  puts("Couldn't empty and delete bucket #{bucket.name}.")
  puts("\t#{e.code}: #{e.message}")
  raise
end
end

# Runs the Amazon S3 getting started scenario.
def run_scenario(scenario)
  puts("-" * 88)
  puts("Welcome to the Amazon S3 getting started demo!")
  puts("-" * 88)

  bucket = scenario.create_bucket
  s3_object = scenario.upload_file(bucket)
  scenario.download_file(s3_object)
  scenario.copy_object(s3_object)
  scenario.list_objects(bucket)
  scenario.delete_bucket(bucket)

  puts("Thanks for watching!")
  puts("-" * 88)
rescue Aws::Errors::ServiceError
  puts("Something went wrong with the demo!")
end

run_scenario(ScenarioGettingStarted.new(Aws::S3::Resource.new)) if $PROGRAM_NAME ==
__FILE__
```

- 如需 API 詳細資訊，請參閱《AWS SDK for Ruby API 參考》中的下列主題。

- [CopyObject](#)
- [CreateBucket](#)
- [DeleteBucket](#)
- [DeleteObjects](#)
- [GetObject](#)
- [ListObjectsV2](#)
- [PutObject](#)

Amazon SES 示例使用 SDK for Ruby

下列程式碼範例說明如何透過 AWS SDK for Ruby 與 Amazon SES 搭配使用來執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境和跨服務範例中查看內容中的動作。

案例是向您展示如何呼叫相同服務中的多個函數來完成特定任務的程式碼範例。

每個範例都包含一個連結 GitHub，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- [動作](#)

動作

取得身分的狀態

下列程式碼範例說明如何取得 Amazon SES 身分的狀態。

適用於 Ruby 的開發套件

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫](#)。

```
require "aws-sdk-ses" # v2: require 'aws-sdk'
```

```
# Create client in us-west-2 region
# Replace us-west-2 with the AWS Region you're using for Amazon SES.
client = Aws::SES::Client.new(region: "us-west-2")

# Get up to 1000 identities
ids = client.list_identities({
  identity_type: "EmailAddress"
})

ids.identities.each do |email|
  attrs = client.get_identity_verification_attributes({
    identities: [email]
  })

  status = attrs.verification_attributes[email].verification_status

  # Display email addresses that have been verified
  if status == "Success"
    puts email
  end
end
```

- 如需 API 詳細資訊，請參閱 AWS SDK for Ruby API 參考[GetIdentityVerificationAttributes](#)中的。

列出身分

下列程式碼範例顯示如何列出 Amazon SES 身分識別。

適用於 Ruby 的開發套件

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫](#)。

```
require "aws-sdk-ses" # v2: require 'aws-sdk'

# Create client in us-west-2 region
# Replace us-west-2 with the AWS Region you're using for Amazon SES.
```

```
client = Aws::SES::Client.new(region: "us-west-2")

# Get up to 1000 identities
ids = client.list_identities({
  identity_type: "EmailAddress"
})

ids.identities.each do |email|
  attrs = client.get_identity_verification_attributes({
    identities: [email]
  })

  status = attrs.verification_attributes[email].verification_status

  # Display email addresses that have been verified
  if status == "Success"
    puts email
  end
end
```

- 如需 API 詳細資訊，請參閱 AWS SDK for Ruby API 參考 [ListIdentities](#) 中的。

傳送電子郵件

下列程式碼範例示範如何使用 Amazon SES 傳送電子郵件。

適用於 Ruby 的開發套件

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在 [AWS 設定和執行程式碼範例儲存庫](#)。

```
require "aws-sdk-ses" # v2: require 'aws-sdk'

# Replace sender@example.com with your "From" address.
# This address must be verified with Amazon SES.
sender = "sender@example.com"

# Replace recipient@example.com with a "To" address. If your account
```



```
# is still in the sandbox, this address must be verified.
recipient = "recipient@example.com"

# Specify a configuration set. To use a configuration
# set, uncomment the next line and line 74.
# configsetname = "ConfigSet"

# The subject line for the email.
subject = "Amazon SES test (AWS SDK for Ruby)"

# The HTML body of the email.
htmlbody =
  "<h1>Amazon SES test (AWS SDK for Ruby)</h1>\"
  '<p>This email was sent with <a href="https://aws.amazon.com/ses/">'\
  'Amazon SES</a> using the <a href="https://aws.amazon.com/sdk-for-ruby/">'\
  "AWS SDK for Ruby</a>."

# The email body for recipients with non-HTML email clients.
textbody = "This email was sent with Amazon SES using the AWS SDK for Ruby."

# Specify the text encoding scheme.
encoding = "UTF-8"

# Create a new SES client in the us-west-2 region.
# Replace us-west-2 with the AWS Region you're using for Amazon SES.
ses = Aws::SES::Client.new(region: "us-west-2")

# Try to send the email.
begin
  # Provide the contents of the email.
  ses.send_email(
    destination: {
      to_addresses: [
        recipient
      ]
    },
    message: {
      body: {
        html: {
          charset: encoding,
          data: htmlbody
        },
        text: {
          charset: encoding,
```

```
        data: textbody
      }
    },
    subject: {
      charset: encoding,
      data: subject
    }
  },
  source: sender,
  # Uncomment the following line to use a configuration set.
  # configuration_set_name: configsetname,
)

puts "Email sent to " + recipient

# If something goes wrong, display an error message.
rescue Aws::SES::Errors::ServiceError => error
  puts "Email not sent. Error message: #{error}"
end
```

- 如需 API 詳細資訊，請參閱 AWS SDK for Ruby API 參考[SendEmail](#)中的。

驗證電子郵件身分

下列程式碼範例說明如何使用 Amazon SES 驗證電子郵件身分。

適用於 Ruby 的開發套件

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫](#)。

```
require "aws-sdk-ses" # v2: require 'aws-sdk'

# Replace recipient@example.com with a "To" address.
recipient = "recipient@example.com"

# Create a new SES resource in the us-west-2 region.
# Replace us-west-2 with the AWS Region you're using for Amazon SES.
```

```
ses = Aws::SES::Client.new(region: "us-west-2")

# Try to verify email address.
begin
  ses.verify_email_identity({
    email_address: recipient
  })

  puts "Email sent to " + recipient

# If something goes wrong, display an error message.
rescue Aws::SES::Errors::ServiceError => error
  puts "Email not sent. Error message: #{error}"
end
```

- 如需 API 詳細資訊，請參閱 AWS SDK for Ruby API 參考[VerifyEmailIdentity](#)中的。

Amazon SES API v2 示例使用 SDK for Ruby

下列程式碼範例說明如何使用 Amazon SES API v2 來執行動作和實作常見案例。AWS SDK for Ruby

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境和跨服務範例中查看內容中的動作。

案例是向您展示如何呼叫相同服務中的多個函數來完成特定任務的程式碼範例。

每個範例都包含一個連結 GitHub，您可以在其中找到如何在內容中設定和執程式碼的指示。

主題


- [動作](#)

動作

傳送電子郵件

以下程式碼範例說明如何傳送 Amazon SES API v2 電子郵件。

適用於 Ruby 的開發套件

 Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫](#)。

```
require "aws-sdk-sesv2"
require_relative "config" # Recipient and sender email addresses.

# Set up the SESv2 client.
client = Aws::SESV2::Client.new(region: AWS_REGION)

def send_email(client, sender_email, recipient_email)
  response = client.send_email(
    {
      from_email_address: sender_email,
      destination: {
        to_addresses: [recipient_email]
      },
      content: {
        simple: {
          subject: {
            data: "Test email subject"
          },
          body: {
            text: {
              data: "Test email body"
            }
          }
        }
      }
    }
  )
  puts "Email sent from #{SENDER_EMAIL} to #{RECIPIENT_EMAIL} with message ID:
#{response.message_id}"
end

send_email(client, SENDER_EMAIL, RECIPIENT_EMAIL)
```

- 如需 API 詳細資訊，請參閱 AWS SDK for Ruby API 參考[SendEmail](#)中的。

使用 SDK for Ruby Amazon SNS 範例

下列程式碼範例說明如何透過 AWS SDK for Ruby 搭配 Amazon SNS 使用來執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境和跨服務範例中查看內容中的動作。

案例是向您展示如何呼叫相同服務中的多個函數來完成特定任務的程式碼範例。

每個範例都包含一個連結 GitHub，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- [動作](#)
- [無伺服器範例](#)

動作

建立主題

下列程式碼範例顯示如何建立 Amazon SNS 主題。

適用於 Ruby 的開發套件

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在 [AWS 程式碼範例儲存庫](#) 中設定和執行。

```
# This class demonstrates how to create an Amazon Simple Notification Service (SNS)
# topic.
class SNSTopicCreator
  # Initializes an SNS client.
  #
  # Utilizes the default AWS configuration for region and credentials.
  def initialize
    @sns_client = Aws::SNS::Client.new
  end
end
```

```
# Attempts to create an SNS topic with the specified name.
#
# @param topic_name [String] The name of the SNS topic to create.
# @return [Boolean] true if the topic was successfully created, false otherwise.
def create_topic(topic_name)
  @sns_client.create_topic(name: topic_name)
  puts "The topic '#{topic_name}' was successfully created."
  true
rescue Aws::SNS::Errors::ServiceError => e
  # Handles SNS service errors gracefully.
  puts "Error while creating the topic named '#{topic_name}': #{e.message}"
  false
end
end

# Example usage:
if $PROGRAM_NAME == __FILE__
  topic_name = "YourTopicName" # Replace with your topic name
  sns_topic_creator = SNSTopicCreator.new

  puts "Creating the topic '#{topic_name}'..."
  unless sns_topic_creator.create_topic(topic_name)
    puts "The topic was not created. Stopping program."
    exit 1
  end
end
end
```

- 如需詳細資訊，請參閱 [《AWS SDK for Ruby 開發人員指南》](#)。
- 如需 API 詳細資訊，請參閱 AWS SDK for Ruby API 參考 [CreateTopic](#) 中的。

列出主題的訂閱者

下列程式碼範例顯示如何擷取 Amazon SNS 主題的訂閱者清單。

適用於 Ruby 的開發套件

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在 [AWS 程式碼範例儲存庫](#) 中設定和執行。

```
# This class demonstrates how to list subscriptions to an Amazon Simple Notification
Service (SNS) topic
class SnsSubscriptionLister
  def initialize(sns_client)
    @sns_client = sns_client
    @logger = Logger.new($stdout)
  end

  # Lists subscriptions for a given SNS topic
  # @param topic_arn [String] The ARN of the SNS topic
  # @return [Types::ListSubscriptionsResponse] subscriptions: The response object
  def list_subscriptions(topic_arn)
    @logger.info("Listing subscriptions for topic: #{topic_arn}")
    subscriptions = @sns_client.list_subscriptions_by_topic(topic_arn: topic_arn)
    subscriptions.subscriptions.each do |subscription|
      @logger.info("Subscription endpoint: #{subscription.endpoint}")
    end
    subscriptions
  rescue Aws::SNS::Errors::ServiceError => e
    @logger.error("Error listing subscriptions: #{e.message}")
    raise
  end
end

# Example usage:
if $PROGRAM_NAME == __FILE__
  sns_client = Aws::SNS::Client.new
  topic_arn = "SNS_TOPIC_ARN" # Replace with your SNS topic ARN
  lister = SnsSubscriptionLister.new(sns_client)

  begin
    lister.list_subscriptions(topic_arn)
  rescue StandardError => e
    puts "Failed to list subscriptions: #{e.message}"
    exit 1
  end
end
```

- 如需詳細資訊，請參閱 [《AWS SDK for Ruby 開發人員指南》](#)。
- 如需 API 詳細資訊，請參閱 AWS SDK for Ruby API 參考 [ListSubscriptions](#) 中的。

列出主題

下列程式碼範例顯示如何列出 Amazon SNS 主題。

適用於 Ruby 的開發套件

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在 [AWS 程式碼範例儲存庫](#) 中設定和執行。

```
require "aws-sdk-sns" # v2: require 'aws-sdk'

def list_topics?(sns_client)
  sns_client.topics.each do |topic|
    puts topic.arn
  rescue StandardError => e
    puts "Error while listing the topics: #{e.message}"
  end
end

def run_me

  region = "REGION"
  sns_client = Aws::SNS::Resource.new(region: region)

  puts "Listing the topics."

  if list_topics?(sns_client)
  else
    puts "The bucket was not created. Stopping program."
    exit 1
  end
end

# Example usage:
run_me if $PROGRAM_NAME == __FILE__
```


- 如需詳細資訊，請參閱 [《AWS SDK for Ruby 開發人員指南》](#)。
- 如需 API 詳細資訊，請參閱 AWS SDK for Ruby API 參考 [ListTopics](#) 中的。

發布到主題

下列程式碼範例顯示如何將訊息發佈到 Amazon SNS 主題。

適用於 Ruby 的開發套件

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在 [AWS 程式碼範例儲存庫](#) 中設定和執行。

```
# Service class for sending messages using Amazon Simple Notification Service (SNS)
class SnsMessageSender
  # Initializes the SnsMessageSender with an SNS client
  #
  # @param sns_client [Aws::SNS::Client] The SNS client
  def initialize(sns_client)
    @sns_client = sns_client
    @logger = Logger.new($stdout)
  end

  # Sends a message to a specified SNS topic
  #
  # @param topic_arn [String] The ARN of the SNS topic
  # @param message [String] The message to send
  # @return [Boolean] true if message was successfully sent, false otherwise
  def send_message(topic_arn, message)
    @sns_client.publish(topic_arn: topic_arn, message: message)
    @logger.info("Message sent successfully to #{topic_arn}.")
    true
  rescue Aws::SNS::Errors::ServiceError => e
    @logger.error("Error while sending the message: #{e.message}")
    false
  end
end

# Example usage:
```

```
if $PROGRAM_NAME == __FILE__
  topic_arn = "SNS_TOPIC_ARN" # Should be replaced with a real topic ARN
  message = "MESSAGE"         # Should be replaced with the actual message content

  sns_client = Aws::SNS::Client.new
  message_sender = SnsMessageSender.new(sns_client)

  @logger.info("Sending message.")
  unless message_sender.send_message(topic_arn, message)
    @logger.error("Message sending failed. Stopping program.")
    exit 1
  end
end
```

- 如需詳細資訊，請參閱 [《AWS SDK for Ruby 開發人員指南》](#)。
- 如需 API 詳細資訊，請參閱 AWS SDK for Ruby API 參考中的 [發佈](#)。

設定主題屬性

下列程式碼範例顯示如何設定 Amazon SNS 主題屬性。

適用於 Ruby 的開發套件

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在 [AWS 程式碼範例儲存庫](#) 中設定和執行。

```
# Service class to enable an SNS resource with a specified policy
class SnsResourceEnabler
  # Initializes the SnsResourceEnabler with an SNS resource client
  #
  # @param sns_resource [Aws::SNS::Resource] The SNS resource client
  def initialize(sns_resource)
    @sns_resource = sns_resource
    @logger = Logger.new($stdout)
  end

  # Sets a policy on a specified SNS topic
```

```
#
# @param topic_arn [String] The ARN of the SNS topic
# @param resource_arn [String] The ARN of the resource to include in the policy
# @param policy_name [String] The name of the policy attribute to set
def enable_resource(topic_arn, resource_arn, policy_name)
  policy = generate_policy(topic_arn, resource_arn)
  topic = @sns_resource.topic(topic_arn)

  topic.set_attributes({
    attribute_name: policy_name,
    attribute_value: policy
  })

  @logger.info("Policy #{policy_name} set successfully for topic #{topic_arn}.")
rescue Aws::SNS::Errors::ServiceError => e
  @logger.error("Failed to set policy: #{e.message}")
end

private

# Generates a policy string with dynamic resource ARNs
#
# @param topic_arn [String] The ARN of the SNS topic
# @param resource_arn [String] The ARN of the resource
# @return [String] The policy as a JSON string
def generate_policy(topic_arn, resource_arn)
  {
    Version: "2008-10-17",
    Id: "__default_policy_ID",
    Statement: [{
      Sid: "__default_statement_ID",
      Effect: "Allow",
      Principal: { "AWS": "*" },
      Action: ["SNS:Publish"],
      Resource: topic_arn,
      Condition: {
        ArnEquals: {
          "AWS:SourceArn": resource_arn
        }
      }
    }]
  }.to_json
end
end
```

```
# Example usage:
if $PROGRAM_NAME == __FILE__
  topic_arn = "MY_TOPIC_ARN"      # Should be replaced with a real topic ARN
  resource_arn = "MY_RESOURCE_ARN" # Should be replaced with a real resource ARN
  policy_name = "POLICY_NAME"     # Typically, this is "Policy"

  sns_resource = Aws::SNS::Resource.new
  enabler = SnsResourceEnabler.new(sns_resource)

  enabler.enable_resource(topic_arn, resource_arn, policy_name)
end
```

- 如需詳細資訊，請參閱 [《AWS SDK for Ruby 開發人員指南》](#)。
- 如需 API 詳細資訊，請參閱 AWS SDK for Ruby API 參考 [SetTopicAttributes](#) 中的。

讓電子郵件地址訂閱主題

下列程式碼範例顯示如何訂閱 Amazon SNS 主題的電子郵件地址。

適用於 Ruby 的開發套件

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在 [AWS 程式碼範例儲存庫](#) 中設定和執行。

```
require "aws-sdk-sns"
require "logger"

# Represents a service for creating subscriptions in Amazon Simple Notification
# Service (SNS)
class SubscriptionService
  # Initializes the SubscriptionService with an SNS client
  #
  # @param sns_client [Aws::SNS::Client] The SNS client
  def initialize(sns_client)
    @sns_client = sns_client
    @logger = Logger.new($stdout)
  end
end
```

```
# Attempts to create a subscription to a topic
#
# @param topic_arn [String] The ARN of the SNS topic
# @param protocol [String] The subscription protocol (e.g., email)
# @param endpoint [String] The endpoint that receives the notifications (email
address)
# @return [Boolean] true if subscription was successfully created, false otherwise
def create_subscription(topic_arn, protocol, endpoint)
  @sns_client.subscribe(topic_arn: topic_arn, protocol: protocol, endpoint:
endpoint)
  @logger.info("Subscription created successfully.")
  true
rescue Aws::SNS::Errors::ServiceError => e
  @logger.error("Error while creating the subscription: #{e.message}")
  false
end
end

# Main execution if the script is run directly
if $PROGRAM_NAME == __FILE__
  protocol = "email"
  endpoint = "EMAIL_ADDRESS" # Should be replaced with a real email address
  topic_arn = "TOPIC_ARN"    # Should be replaced with a real topic ARN

  sns_client = Aws::SNS::Client.new
  subscription_service = SubscriptionService.new(sns_client)

  @logger.info("Creating the subscription.")
  unless subscription_service.create_subscription(topic_arn, protocol, endpoint)
    @logger.error("Subscription creation failed. Stopping program.")
    exit 1
  end
end
end
```

- 如需詳細資訊，請參閱 [《AWS SDK for Ruby 開發人員指南》](#)。
- 如需 API 詳細資訊，請參閱 [《AWS SDK for Ruby API 參考》](#) 中的 [訂閱](#)。

無伺服器範例

使用 Amazon SNS 觸發條件調用 Lambda 函數

下列程式碼範例示範如何實作 Lambda 函數，該函數會接收來自 SNS 主題的訊息而觸發的事件。函數會從事件參數擷取訊息，並記錄每一則訊息的內容。

適用於 Ruby 的開發套件

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[無伺服器範例](#)儲存庫中設定和執行。

使用紅寶石使用與 Lambda 的 SNS 事件。

```
def lambda_handler(event:, context:)
  event['Records'].map { |record| process_message(record) }
end

def process_message(record)
  message = record['Sns']['Message']
  puts("Processing message: #{message}")
rescue StandardError => e
  puts("Error processing message: #{e}")
  raise
end
```

Amazon SQS 示例使用 SDK for Ruby

下列程式碼範例說明如何透過 AWS SDK for Ruby 搭配 Amazon SQS 使用來執行動作和實作常見案例。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境和跨服務範例中查看內容中的動作。

案例是向您展示如何呼叫相同服務中的多個函數來完成特定任務的程式碼範例。

每個範例都包含一個連結 GitHub，您可以在其中找到如何在內容中設定和執行程式碼的指示。

主題

- [動作](#)
- [無伺服器範例](#)

動作

變更訊息逾時可見性

下列程式碼範例顯示如何變更 Amazon SQS 訊息逾時可見性。

適用於 Ruby 的開發套件

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫](#)。

```
require "aws-sdk-sqs" # v2: require 'aws-sdk'
# Replace us-west-2 with the AWS Region you're using for Amazon SQS.
sqs = Aws::SQS::Client.new(region: "us-west-2")

begin
  queue_name = "my-queue"
  queue_url = sqs.get_queue_url(queue_name: queue_name).queue_url

  receive_message_result_before = sqs.receive_message({
    queue_url: queue_url,
    max_number_of_messages: 10 # Receive up to 10 messages, if there are that many.
  })

  puts "Before attempting to change message visibility timeout: received
#{receive_message_result_before.messages.count} message(s)."

  receive_message_result_before.messages.each do |message|
    sqs.change_message_visibility({
      queue_url: queue_url,
      receipt_handle: message.receipt_handle,
      visibility_timeout: 30 # This message will not be visible for 30 seconds after
first receipt.
    })
  end
end
```

```
    })
  end

  # Try to retrieve the original messages after setting their visibility timeout.
  receive_message_result_after = sqs.receive_message({
    queue_url: queue_url,
    max_number_of_messages: 10
  })

  puts "\nAfter attempting to change message visibility timeout: received
#{receive_message_result_after.messages.count} message(s)."
```

```
rescue Aws::SQS::Errors::NonExistentQueue
  puts "Cannot receive messages for a queue named '#{receive_queue_name}', as it
does not exist."
end
```

- 如需 API 詳細資訊，請參閱 AWS SDK for Ruby API 參考[ChangeMessageVisibility](#)中的。

建立佇列

下列程式碼範例示範如何建立 Amazon SQS 佇列。

適用於 Ruby 的開發套件

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫](#)。

```
# This code example demonstrates how to create a queue in Amazon Simple Queue
Service (Amazon SQS).

require "aws-sdk-sqs"

# @param sqs_client [Aws::SQS::Client] An initialized Amazon SQS client.
# @param queue_name [String] The name of the queue.
# @return [Boolean] true if the queue was created; otherwise, false.
# @example
#   exit 1 unless queue_created?(
#     Aws::SQS::Client.new(region: 'us-west-2'),
```



```
# 'my-queue'
# )
def queue_created?(sqs_client, queue_name)
  sqs_client.create_queue(queue_name: queue_name)
  true
rescue StandardError => e
  puts "Error creating queue: #{e.message}"
  false
end

# Full example call:
# Replace us-west-2 with the AWS Region you're using for Amazon SQS.
def run_me
  region = "us-west-2"
  queue_name = "my-queue"
  sqs_client = Aws::SQS::Client.new(region: region)

  puts "Creating the queue named '#{queue_name}'..."

  if queue_created?(sqs_client, queue_name)
    puts "Queue created."
  else
    puts "Queue not created."
  end
end

# Example usage:
run_me if $PROGRAM_NAME == __FILE__
```

- 如需 API 詳細資訊，請參閱 AWS SDK for Ruby API 參考[CreateQueue](#)中的。

刪除佇列

下列程式碼範例顯示如何刪除 Amazon SQS 佇列。

適用於 Ruby 的開發套件

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫](#)。

```
require "aws-sdk-sqs" # v2: require 'aws-sdk'
# Replace us-west-2 with the AWS Region you're using for Amazon SQS.
sqs = Aws::SQS::Client.new(region: "us-west-2")

sqs.delete_queue(queue_url: URL)
```

- 如需 API 詳細資訊，請參閱 AWS SDK for Ruby API 參考 [DeleteQueue](#) 中的。

列出佇列

下列程式碼範例顯示如何列出 Amazon SQS 佇列。

適用於 Ruby 的開發套件

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在 [AWS 設定和執行程式碼範例儲存庫](#)。

```
require "aws-sdk-sqs"
require "aws-sdk-sts"

# @param sqs_client [Aws::SQS::Client] An initialized Amazon SQS client.
# @example
# list_queue_urls(Aws::SQS::Client.new(region: 'us-west-2'))
def list_queue_urls(sqs_client)
  queues = sqs_client.list_queues

  queues.queue_urls.each do |url|
    puts url
  end
rescue StandardError => e
  puts "Error listing queue URLs: #{e.message}"
end

# Lists the attributes of a queue in Amazon Simple Queue Service (Amazon SQS).
#
# @param sqs_client [Aws::SQS::Client] An initialized Amazon SQS client.
# @param queue_url [String] The URL of the queue.
```

```
# @example
# list_queue_attributes(
#   Aws::SQS::Client.new(region: 'us-west-2'),
#   'https://sqs.us-west-2.amazonaws.com/111111111111/my-queue'
# )
def list_queue_attributes(sqs_client, queue_url)
  attributes = sqs_client.get_queue_attributes(
    queue_url: queue_url,
    attribute_names: ["All"]
  )

  attributes.attributes.each do |key, value|
    puts "#{key}: #{value}"
  end

rescue StandardError => e
  puts "Error getting queue attributes: #{e.message}"
end

# Full example call:
# Replace us-west-2 with the AWS Region you're using for Amazon SQS.
def run_me
  region = "us-west-2"
  queue_name = "my-queue"

  sqs_client = Aws::SQS::Client.new(region: region)

  puts "Listing available queue URLs..."
  list_queue_urls(sqs_client)

  sts_client = Aws::STS::Client.new(region: region)

  # For example:
  # 'https://sqs.us-west-2.amazonaws.com/111111111111/my-queue'
  queue_url = "https://sqs." + region + ".amazonaws.com/" +
    sts_client.get_caller_identity.account + "/" + queue_name

  puts "\nGetting information about queue '#{queue_name}'..."
  list_queue_attributes(sqs_client, queue_url)
end
```

- 如需 API 詳細資訊，請參閱 AWS SDK for Ruby API 參考 [ListQueues](#) 中的。

從佇列接收訊息

下列程式碼範例顯示如何從 Amazon SQS 佇列接收訊息。

適用於 Ruby 的開發套件

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫](#)。

```
require "aws-sdk-sqs"
require "aws-sdk-sts"

# Receives messages in a queue in Amazon Simple Queue Service (Amazon SQS).
#
# @param sqs_client [Aws::SQS::Client] An initialized Amazon SQS client.
# @param queue_url [String] The URL of the queue.
# @param max_number_of_messages [Integer] The maximum number of messages
#   to receive. This number must be 10 or less. The default is 10.
# @example
#   receive_messages(
#     Aws::SQS::Client.new(region: 'us-west-2'),
#     'https://sqs.us-west-2.amazonaws.com/111111111111/my-queue',
#     10
#   )
def receive_messages(sqs_client, queue_url, max_number_of_messages = 10)

  if max_number_of_messages > 10
    puts "Maximum number of messages to receive must be 10 or less. " \
      "Stopping program."
    return
  end

  response = sqs_client.receive_message(
    queue_url: queue_url,
    max_number_of_messages: max_number_of_messages
  )

  if response.messages.count.zero?
    puts "No messages to receive, or all messages have already " \
      "been previously received."
```

```
    return
  end

  response.messages.each do |message|
    puts "-" * 20
    puts "Message body: #{message.body}"
    puts "Message ID:  #{message.message_id}"
  end

rescue StandardError => e
  puts "Error receiving messages: #{e.message}"
end

# Full example call:
# Replace us-west-2 with the AWS Region you're using for Amazon SQS.
def run_me
  region = "us-west-2"
  queue_name = "my-queue"
  max_number_of_messages = 10

  sts_client = Aws::STS::Client.new(region: region)

  # For example:
  # 'https://sqs.us-west-2.amazonaws.com/111111111111/my-queue'
  queue_url = "https://sqs." + region + ".amazonaws.com/" +
    sts_client.get_caller_identity.account + "/" + queue_name

  sqs_client = Aws::SQS::Client.new(region: region)

  puts "Receiving messages from queue '#{queue_name}'..."

  receive_messages(sqs_client, queue_url, max_number_of_messages)
end

# Example usage:
run_me if $PROGRAM_NAME == __FILE__
```

- 如需 API 詳細資訊，請參閱 AWS SDK for Ruby API 參考[ReceiveMessage](#)中的。

將一批消息發送到隊列

下列程式碼範例顯示如何將一批訊息傳送至 Amazon SQS 佇列。

適用於 Ruby 的開發套件

 Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫](#)。

```
require "aws-sdk-sqs"
require "aws-sdk-sts"

#
# @param sqs_client [Aws::SQS::Client] An initialized Amazon SQS client.
# @param queue_url [String] The URL of the queue.
# @param entries [Hash] The contents of the messages to be sent,
#   in the correct format.
# @return [Boolean] true if the messages were sent; otherwise, false.
# @example
#   exit 1 unless messages_sent?(
#     Aws::SQS::Client.new(region: 'us-west-2'),
#     'https://sqs.us-west-2.amazonaws.com/111111111111/my-queue',
#     [
#       {
#         id: 'Message1',
#         message_body: 'This is the first message.'
#       },
#       {
#         id: 'Message2',
#         message_body: 'This is the second message.'
#       }
#     ]
#   )
def messages_sent?(sqs_client, queue_url, entries)
  sqs_client.send_message_batch(
    queue_url: queue_url,
    entries: entries
  )
  true
rescue StandardError => e
  puts "Error sending messages: #{e.message}"
  false
end
```

```
# Full example call:
# Replace us-west-2 with the AWS Region you're using for Amazon SQS.
def run_me
  region = "us-west-2"
  queue_name = "my-queue"
  entries = [
    {
      id: "Message1",
      message_body: "This is the first message."
    },
    {
      id: "Message2",
      message_body: "This is the second message."
    }
  ]

  sts_client = Aws::STS::Client.new(region: region)

  # For example:
  # 'https://sqs.us-west-2.amazonaws.com/111111111111/my-queue'
  queue_url = "https://sqs." + region + ".amazonaws.com/" +
    sts_client.get_caller_identity.account + "/" + queue_name

  sqs_client = Aws::SQS::Client.new(region: region)

  puts "Sending messages to the queue named '#{queue_name}'..."

  if messages_sent?(sqs_client, queue_url, entries)
    puts "Messages sent."
  else
    puts "Messages not sent."
  end
end
```

- 如需 API 詳細資訊，請參閱 AWS SDK for Ruby API 參考 [SendMessageBatch](#) 中的。

將訊息傳送至佇列

下列程式碼範例顯示如何將訊息傳送至 Amazon SQS 佇列。

適用於 Ruby 的開發套件

 Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫](#)。

```
require "aws-sdk-sqs"
require "aws-sdk-sts"

# @param sqs_client [Aws::SQS::Client] An initialized Amazon SQS client.
# @param queue_url [String] The URL of the queue.
# @param message_body [String] The contents of the message to be sent.
# @return [Boolean] true if the message was sent; otherwise, false.
# @example
#   exit 1 unless message_sent?(
#     Aws::SQS::Client.new(region: 'us-west-2'),
#     'https://sqs.us-west-2.amazonaws.com/111111111111/my-queue',
#     'This is my message.'
#   )
def message_sent?(sqs_client, queue_url, message_body)
  sqs_client.send_message(
    queue_url: queue_url,
    message_body: message_body
  )
  true
rescue StandardError => e
  puts "Error sending message: #{e.message}"
  false
end

# Full example call:
# Replace us-west-2 with the AWS Region you're using for Amazon SQS.
def run_me
  region = "us-west-2"
  queue_name = "my-queue"
  message_body = "This is my message."

  sts_client = Aws::STS::Client.new(region: region)

  # For example:
  # 'https://sqs.us-west-2.amazonaws.com/111111111111/my-queue'
```



```
queue_url = "https://sqs." + region + ".amazonaws.com/" +
  sts_client.get_caller_identity.account + "/" + queue_name

sqs_client = Aws::SQS::Client.new(region: region)

puts "Sending a message to the queue named '#{queue_name}'..."

if message_sent?(sqs_client, queue_url, message_body)
  puts "Message sent."
else
  puts "Message not sent."
end
end

# Example usage:
run_me if $PROGRAM_NAME == __FILE__
```

- 如需 API 詳細資訊，請參閱 AWS SDK for Ruby API 參考[SendMessage](#)中的。

無伺服器範例

使用 Amazon SQS 觸發條件調用 Lambda 函數

下列程式碼範例示範如何實作 Lambda 函數，此函數會接收由 SQS 佇列接收訊息而觸發的事件。函數會從事件參數擷取訊息，並記錄每一則訊息的內容。

適用於 Ruby 的開發套件

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[無伺服器範例](#)儲存庫中設定和執行。

使用 Ruby 搭配 Lambda 來使用 SQS 事件。

```
def lambda_handler(event:, context:)
  event['Records'].each do |message|
    process_message(message)
  end
  puts "done"
```

```
end

def process_message(message)
  begin
    puts "Processed message #{message['body']}"
    # TODO: Do interesting work based on the new message
  rescue StandardError => err
    puts "An error occurred"
    raise err
  end
end
```

使用 Amazon SQS 觸發條件報告 Lambda 函數的批次項目失敗

下列程式碼範例示範如何針對接收來自 SQS 佇列之事件的 Lambda 函數實作部分批次回應。此函數會在回應中報告批次項目失敗，指示 Lambda 稍後重試這些訊息。

適用於 Ruby 的開發套件

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[無伺服器範例](#)儲存庫中設定和執行。

使用 Ruby 搭配 Lambda 報告 SQS 批次項目失敗。

```
require 'json'

def lambda_handler(event:, context:)
  if event
    batch_item_failures = []
    sqs_batch_response = {}

    event["Records"].each do |record|
      begin
        # process message
      rescue StandardError => e
        batch_item_failures << {"itemIdentifier" => record['messageId']}
      end
    end
  end
end
```

```
sqs_batch_response["batchItemFailures"] = batch_item_failures
return sqs_batch_response
end
end
```

AWS STS 使用 SDK for Ruby 示例

下列程式碼範例說明如何使用 AWS SDK for Ruby 與來執行動作及實作常見案例 AWS STS。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境和跨服務範例中查看內容中的動作。

案例是向您展示如何呼叫相同服務中的多個函數來完成特定任務的程式碼範例。

每個範例都包含一個連結 GitHub，您可以在其中找到如何在內容中設定和執程式碼的指示。

主題

- [動作](#)

動作

擔任角色

下列程式碼範例會示範如何假設具有的角色 AWS STS。

適用於 Ruby 的開發套件

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執程式碼範例儲存庫](#)。

```
# Creates an AWS Security Token Service (AWS STS) client with specified
credentials.
# This is separated into a factory function so that it can be mocked for unit
testing.
#
# @param key_id [String] The ID of the access key used by the STS client.
```

```
# @param key_secret [String] The secret part of the access key used by the STS
client.
def create_sts_client(key_id, key_secret)
  Aws::STS::Client.new(access_key_id: key_id, secret_access_key: key_secret)
end

# Gets temporary credentials that can be used to assume a role.
#
# @param role_arn [String] The ARN of the role that is assumed when these
credentials
#
#           are used.
# @param sts_client [AWS::STS::Client] An AWS STS client.
# @return [Aws::AssumeRoleCredentials] The credentials that can be used to assume
the role.
def assume_role(role_arn, sts_client)
  credentials = Aws::AssumeRoleCredentials.new(
    client: sts_client,
    role_arn: role_arn,
    role_session_name: "create-use-assume-role-scenario"
  )
  @logger.info("Assumed role '#{role_arn}', got temporary credentials.")
  credentials
end
```

- 如需 API 詳細資訊，請參閱 AWS SDK for Ruby API 參考[AssumeRole](#)中的。

Amazon WorkDocs 示例使用 SDK for Ruby

下列程式碼範例說明如何透過 AWS SDK for Ruby 搭配 Amazon 使用來執行動作和實作常見案例 WorkDocs。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境和跨服務範例中查看內容中的動作。

案例是向您展示如何呼叫相同服務中的多個函數來完成特定任務的程式碼範例。

每個範例都包含一個連結 GitHub，您可以在其中找到如何在內容中設定和執程式碼的指示。

主題

- [動作](#)

動作

描述根資料夾內容

下列程式碼範例說明如何描述 Amazon WorkDocs 根資料夾內容。

適用於 Ruby 的開發套件

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫](#)。

```
# Retrieves the root folder for a user by email
# @param users [Array<Types::User>] A list of users selected from API response
# @param user_email [String] The email of the user.
def get_user_folder(users, user_email)
  user = users.find { |user| user.email_address == user_email }
  if user
    user.root_folder_id
  else
    @logger.error "Could not get root folder for user with email address
#{user_email}"
    exit(1)
  end
end

# Describes the contents of a folder
# @param [String] folder_id - The Id of the folder to describe.
def describe_folder_contents(folder_id)
  resp = @client.describe_folder_contents({
    folder_id: folder_id, # required
    sort: "NAME", # accepts DATE, NAME
    order: "ASCENDING", # accepts
    ASCENDING, DESCENDING
  })

  resp.documents.each do |doc|
    md = doc.latest_version_metadata
    @logger.info "Name:          #{md.name}"
    @logger.info "Size (bytes):  #{md.size}"
    @logger.info "Last modified: #{doc.modified_timestamp}"
    @logger.info "Doc ID:       #{doc.id}"
    @logger.info "Version ID:   #{md.id}"
  end
end
```

```
@logger.info ""
end
rescue Aws::WorkDocs::Errors::ServiceError => e
  @logger.error "Error listing folder contents: #{e.message}"
  exit(1)
end
```

- 如需 API 詳細資訊，請參閱 AWS SDK for Ruby API 參考 [DescribeRootFolders](#) 中的。

描述使用者

下列程式碼範例說明如何描述 Amazon WorkDocs 使用者。

適用於 Ruby 的開發套件

Note

還有更多關於 GitHub。尋找完整範例，並了解如何在 [AWS 設定和執行程式碼範例儲存庫](#)。

```
# Describes users within an organization
# @param [String] org_id: The ID of the org.
def describe_users(org_id)
  resp = @client.describe_users({
    organization_id: org_id,
    include: "ALL", # accepts ALL, ACTIVE_PENDING
    order: "ASCENDING", # accepts ASCENDING,
DESCENDING
    sort: "USER_NAME", # accepts USER_NAME,
FULL_NAME, STORAGE_LIMIT, USER_STATUS, STORAGE_USED
  })

  resp.users.each do |user|
    @logger.info "First name: #{user.given_name}"
    @logger.info "Last name:  #{user.surname}"
    @logger.info "Email:      #{user.email_address}"
    @logger.info "Root folder: #{user.root_folder_id}"
    @logger.info ""
  end
  resp.users
rescue Aws::WorkDocs::Errors::ServiceError => e
  @logger.error "AWS WorkDocs Service Error: #{e.message}"
end
```

```
    exit(1)
  end
```

- 如需 API 詳細資訊，請參閱 AWS SDK for Ruby API 參考 [DescribeUsers](#) 中的。

使用 SDK for Ruby 跨服務範例

下列範例應用程式使 AWS SDK for Ruby 用跨多個工作 AWS 服務。

跨服務範例鎖定進階層級的經驗，可協助您開始建置應用程式。

範例

- [建立可分析客戶意見回饋並合成音訊的應用程式](#)

建立可分析客戶意見回饋並合成音訊的應用程式

適用於 Ruby 的開發套件

此範例應用程式會分析和存儲客戶的意見回饋卡。具體來說，它滿足了紐約市一家虛構飯店的需求。飯店以實體評論卡的形式收到賓客以各種語言撰寫的意見回饋。這些意見回饋透過 Web 用戶端上傳至應用程式。評論卡的影像上傳後，系統會執行下列步驟：

- 文字內容是使用 Amazon Textract 從影像中擷取。
- Amazon Comprehend 會決定擷取文字及其用語的情感。
- 擷取的文字內容會使用 Amazon Translate 翻譯成英文。
- Amazon Polly 會使用擷取的文字內容合成音訊檔案。

完整的應用程式可透過 AWS CDK 部署。如需原始程式碼和部署指示，請參閱中的專案 [GitHub](#)。

此範例中使用的服務

- Amazon Comprehend
- Lambda
- Amazon Polly
- Amazon Textract
- Amazon Translate

適用於 AWS SDK for Ruby 安全性

雲端安全是 Amazon Web Services (AWS) 最重視的一環。身為 AWS 客戶的您，將能從資料中心和網路架構的建置中獲益，以滿足組織最為敏感的安全要求。安全是 AWS 與您之間共同承擔的責任。[共同責任模型](#) 將此描述為雲端本身的安全和雲端內部的安全。

雲的安全性 — AWS 負責保護運行 AWS 雲中提供的所有服務的基礎設施，並為您提供可以安全使用的服務。我們的安全責任是我們的首要任務 AWS，並且我們的安全性有效性是由第三方審計師定期測試和驗證，作為[AWS 合規計劃](#)的一部分。

雲端安全性 — 您的責任取決於 AWS 服務 您使用的人，以及其他因素，包括資料的敏感度、組織的需求以及適用的法律和法規。

主題

- [AWS SDK for Ruby 中的資料保護](#)
- [適用於 Ruby 的 AWS SDK Identity and Access Management](#)
- [適用於 Ruby 的 AWS SDK 合規性驗證](#)
- [適用於 AWS SDK for Ruby 彈性](#)
- [適用於 Ruby 的 AWS SDK 基礎架構安全性](#)
- [在 AWS 開發套件中強制執行適用於 Ruby 的最低 TLS 版本](#)
- [Amazon S3 加密客戶端遷移](#)

AWS SDK for Ruby 中的資料保護

AWS [共用責任模型](#)適用於中的資料保護。如此模型中所述，AWS 負責保護執行所有 AWS 雲端。您負責維護在此基礎設施上託管內容的控制權。您還必須負責您所使用 AWS 服務 的安全組態和管理任務。如需資料隱私權的詳細資訊，請參閱[資料隱私權常見問答集](#)。如需歐洲資料保護的相關資訊，請參閱 AWS 安全性部落格上的 [AWS 共同責任模型](#)和 [GDPR](#) 部落格文章。

基於資料保護目的，我們建議您使用 AWS IAM Identity Center 或 AWS Identity and Access Management (IAM) 保護 AWS 帳戶 登入資料並設定個別使用者。如此一來，每個使用者都只會獲得授與完成其任務所必須的許可。我們也建議您採用下列方式保護資料：

- 每個帳戶均要使用多重要素驗證 (MFA)。
- 使用 SSL/TLS 與 AWS 資源進行通訊。我們需要 TLS 1.2 並建議使用 TLS 1.3。

- 使用設定 API 和使用者活動記錄 AWS CloudTrail。
- 使用 AWS 加密解決方案以及其中的所有默認安全控制 AWS 服務。
- 使用進階的受管安全服務 (例如 Amazon Macie) , 協助探索和保護儲存在 Amazon S3 的敏感資料。
- 如果您在透過命令列介面或 API 存取時需要經 AWS 過 FIPS 140-2 驗證的加密模組, 請使用 FIPS 端點。如需 FIPS 和 FIPS 端點的相關資訊, 請參閱[聯邦資訊處理標準 \(FIPS\) 140-2 概觀](#)。

我們強烈建議您絕對不要將客戶的電子郵件地址等機密或敏感資訊, 放在標籤或自由格式的文字欄位中, 例如名稱 欄位。這包括當您使用主控台、API 或 AWS SDK 時 AWS 服務 使用或其他使用時。AWS CLI您在標籤或自由格式文字欄位中輸入的任何資料都可能用於計費或診斷日誌。如果您提供外部伺服器的 URL, 我們強烈建議請勿在驗證您對該伺服器請求的 URL 中包含憑證資訊。

適用於 Ruby 的 AWS SDK Identity and Access Management

AWS Identity and Access Management (IAM) 是一種 Amazon Web Services (AWS) 服務, 可協助管理員安全地控制 AWS 資源的存取。IAM 管理員可以控制身分身分驗證 (已登入) 和授權 (具有許可) 以使用 AWS 服務資源。IAM 是您可以使用的 AWS 服務, 無需額外付費。

要使用 AWS SDK for Ruby 訪問 AWS, 您需要一個 AWS 帳戶和 AWS 憑據。為提高 AWS 帳戶的安全性, 建議您使用 IAM 使用者 提供存取登入資料, 而不要使用 AWS 帳戶登入資料。

如需使用 IAM 的詳細資訊, 請參閱 [IAM](#)。

如需概略了解 IAM 使用者以及它們對帳戶安全性的重要性, 請參閱 [Amazon Web Services 一般參考](#) 中的 [AWS 安全登入資料](#)。

AWS SDK for Ruby 會透過其支援的特定亞馬遜網路服務 (AWS) 服務, 遵循[共同的責任模型](#)。如需 AWS 服務 安全性資訊, 請參閱[AWS 服務 安全性說明文件頁面AWS 服務, 以及符合性計劃 AWS 遵循合規工作範圍內的](#)資訊。

適用於 Ruby 的 AWS SDK 合規性驗證

AWS SDK for Ruby 會透過其支援的特定亞馬遜網路服務 (AWS) 服務, 遵循[共同的責任模型](#)。如需 AWS 服務 安全性資訊, 請參閱[AWS 服務 安全性說明文件頁面AWS 服務, 以及符合性計劃 AWS 遵循合規工作範圍內的](#)資訊。

Amazon Web Services (AWS) 服務的安全性和合規性是由第三方稽核人員評估為多個 AWS 合規計劃的一部分。這些措施包括 SOC、PCI、FedRAMP、HIPAA 等。AWS 在合規方案的[AWS 服務範圍 AWS 服務](#) 中, 提供經常更新的特定合規方案範圍清單。

第三方稽核報告可供您使用下載 AWS Artifact。如需詳細資訊，請參閱[在 AWS Artifact 中下載報表](#)。

如需 AWS 規範遵循方案的詳細資訊，請參閱[AWS 合規性方案](#)。

使用 AWS SDK for Ruby 存取時，您的合規責任取決 AWS 服務 於資料的敏感度、組織的合規目標以及適用的法律和法規。如果您使用的 AWS 服務 產品符合 HIPAA、PCI 或 FedRAMP 等標準，則會 AWS 提供資源以協助：

- [安全性與合規性快速入門指南](#) — 部署指南，討論架構考量，並提供步驟，以便在上部署以安全性為重點和遵循法規遵循的基準環境。AWS
- [建構 HIPAA 安全性與合規性白皮書](#) — 說明公司如何使用建立 HIPAA 相容應用 AWS 程式的白皮書。
- [AWS 合規性資源](#) — 可能適用於您的產業和位置的工作簿和指南集合。
- [AWS Config](#) — 評估您的資源組態是否符合內部實務、產業準則和法規的服務。
- [AWS Security Hub](#) — 全面檢視您的安全狀態，可協助 AWS 您檢查您是否符合安全性產業標準和最佳做法。

適用於 AWS SDK for Ruby 彈性

Amazon Web Services (AWS) 全球基礎設施是圍繞 AWS 區域 和可用區域建立的。

AWS 區域 提供多個實體分離和隔離的可用區域，這些區域透過低延遲、高輸送量和高度備援的網路連線。

透過可用區域，您所設計與操作的應用程式和資料庫，就能夠在可用區域之間自動容錯移轉，而不會發生中斷。可用區域的可用性、容錯能力和擴充能力，均較單一或多個資料中心的傳統基礎設施還高。

如需 AWS 區域 和可用區域的詳細資訊，請參閱[AWS 全域基礎結構](#)。

AWS SDK for Ruby 會透過其支援的特定亞馬遜網路服務 (AWS) 服務，遵循[共同的責任模型](#)。如需 AWS 服務 安全性資訊，請參閱[AWS 服務 安全性說明文件頁面](#)[AWS 服務](#)，以及符合性計劃 [AWS 遵循合規工作範圍內的](#)資訊。

適用於 Ruby 的 AWS SDK 基礎架構安全性

AWS SDK for Ruby 會透過其支援的特定亞馬遜網路服務 (AWS) 服務，遵循[共同的責任模型](#)。如需 AWS 服務 安全性資訊，請參閱[AWS 服務 安全性說明文件頁面](#)[AWS 服務](#)，以及符合性計劃 [AWS 遵循合規工作範圍內的](#)資訊。

如需 AWS 安全性程序的相關資訊，請參閱 [AWS：安全性程序概觀](#) 白皮書。

在 AWS 開發套件中強制執行適用於 Ruby 的最低 TLS 版本

適用於 Ruby 的 AWS SDK 之間的通訊，並 AWS 使用安全通訊端層 (SSL) 或傳輸層安全性 (TLS) 進行保護。所有版本的 SSL 和 1.2 之前的 TLS 版本都有可能危及您通訊安全性的弱點 AWS。基於這個原因，您應該確定您使用的是 Ruby 版本支援 TLS 1.2 或更新版本的 Ruby 版本的 AWS SDK。

紅寶石會使用 OpenSSL 程式庫來保護 HTTP 連線的安全。[透過系統套件管理員 \(yum、apt 及其他\)、官方安裝程式或 Ruby 管理程式 \(rbenv、RVM 及其他\) 安裝的支援版本 \(1.9.3 及更新版本\) 通常會納入支援 TLS 1.2 的 OpenSSL 1.0.1 或更新版本。](#)

當與支援的 Ruby 版本搭配 OpenSSL 1.0.1 或更新版本搭配使用時，AWS SDK for Ruby 偏好 TLS 1.2，並使用用戶端和伺服器都支援的最新版本的 SSL 或 TLS。這始終至少是 TLS 1.2 的 AWS 服務。(SDK 會使用 `Ruby Net::HTTP` 類別搭配使用 `use_ssl=true`。)

正在檢 OpenSSL 版本

為了確保您的 Ruby 安裝使用 OpenSSL 1.0.1 或更新版本，請輸入以下命令。

```
ruby -r openssl -e 'puts OpenSSL::OPENSSL_VERSION'
```

另一種取得 OpenSSL 版本的方法是直接查詢 `openssl` 可執行檔。首先，使用以下命令找到適當的可執行文件。

```
ruby -r rbconfig -e 'puts RbConfig::CONFIG["configure_args"]'
```

輸出應該具有 `--with-openssl-dir=/path/to/openssl` 指示 OpenSSL 安裝的位置。記下此路徑。若要檢查 OpenSSL 的版本，請輸入下列指令。

```
cd /path/to/openssl  
bin/openssl version
```

這後一種方法可能不適用於 Ruby 的所有安裝。

升級 TLS 支援

[如果您的 Ruby 安裝所使用的 OpenSSL 版本早於 1.0.1，請使用您的系統套件管理員、Ruby 安裝程式或 Ruby 管理員來升級您的 Ruby 或 OpenSSL 安裝程式，如 Ruby 的安裝指南所述。如果您是從](#)

源代碼安裝 Ruby，請先安裝[最新的 OpenSSL](#)，然後在運行`--with-openssl-dir=/path/to/upgraded/openssl./configure`時通過。

Amazon S3 加密客戶端遷移

本主題說明如何將應用程式從 Amazon 簡單儲存服務 (Amazon S3) 加密用戶端的版本 1 (V1) 遷移到第 2 版 (V2)，並確保應用程式在整個遷移過程中可用性。

移轉概觀

此遷移分兩個階段進行：

1. 更新現有用戶端以讀取新格式。首先，將適用 AWS SDK for Ruby 的更新版本部署到您的應用程式中。這將允許現有的 V1 加密用戶端解密由新 V2 用戶端寫入的物件。如果您的應用程式使用多個 AWS SDK，您必須分別升級每個 SDK。
2. 將加密和解密用戶端移轉至 V2。一旦所有 V1 加密用戶端都可以讀取新格式，您就可以將現有的加密和解密用戶端遷移至各自的 V2 版本。

更新現有用戶端以讀取新格式

V2 加密用戶端使用舊版用戶端不支援的加密演算法。遷移的第一個步驟是將 V1 解密用戶端更新為最新的 SDK 版本。完成此步驟之後，您的應用程式的 V1 用戶端將能夠解密由 V2 加密用戶端加密的物件。請參閱以下有關 Ruby 適用於每個主要版本的 AWS SDK 的詳細信息。

為紅寶石版本 3 更新 AWS SDK

版本 3 是適用於紅寶石的 AWS SDK 的最新版本。要完成此遷移，您需要使用 `aws-sdk-s3` gem 的 1.76.0 版或更高版本。

從命令列安裝

對於安裝 `aws-sdk-s3` gem 的項目，請使用版本選項來驗證是否安裝了最低版本的 1.76.0。

```
gem install aws-sdk-s3 -v '>= 1.76.0'
```

使用寶石文件

對於使用 Gemfile 來管理依賴項的項目，請將 `aws-sdk-s3` gem 的最低版本設置為 1.76.0。例如：

```
gem 'aws-sdk-s3', '>= 1.76.0'
```

1. 修改你的寶石文件。
2. 執行 `bundle update aws-sdk-s3`。若要驗證您的版本，請執行 `bundle info aws-sdk-s3`。

適用於紅寶石版 AWS 本 2 的最新 SDK

Ruby 版本第二版的 AWS SDK 將於 2021 年 11 月 21 日進入[維護模式](#)。要完成此遷移，您需要使用 aws-sdk 寶石的 2.11.562 版或更高版本。

從命令列安裝

對於安裝 `aws-sdk gem` 的項目，請從命令行使用版本選項來驗證是否已安裝 2.11.562 的最低版本。

```
gem install aws-sdk -v '>= 2.11.562'
```

使用寶石文件

對於使用 Gemfile 來管理依賴項的項目，請將 `aws-sdk gem` 的最低版本設置為 2.11.562。例如：

```
gem 'aws-sdk', '>= 2.11.562'
```

1. 修改你的寶石文件。如果您有 GEM 檔案鎖定檔案，請將其刪除或更新。
2. 執行 `bundle update aws-sdk`。若要驗證您的版本，請執行 `bundle info aws-sdk`。

將加密和解密用戶端移轉至 V2

更新用戶端以讀取新的加密格式之後，您可以將應用程式更新為 V2 加密和解密用戶端。下列步驟說明如何成功將程式碼從 V1 遷移至 V2。

在更新程式碼以使用 V2 加密用戶端之前，請確定您已遵循上述步驟，並使用 `aws-sdk-s3 gem` 版本 2.11.562 或更新版本。

Note

使用 AES-GCM 進行解密時，請在開始使用解密的資料之前將整個物件讀到最後。這是為了驗證該對象自加密以來沒有被修改。

設定 V2 加密用戶端

加密 V2:: 客戶端需要額外的配置。如需詳細的組態資訊，請參閱[加密 V2:: Client 文件](#)或本主題稍後提供的範例。

1. 必須在用戶端建構中指定金鑰換行方法和內容加密演算法。建立新的時候`EncryptionV2::Client`，您需要為`key_wrap_schema`和提供值`content_encryption_schema`。

`key_wrap_schema`-如果您正在使用 AWS KMS，則必須將其設置為`:kms_context`。如果您使用對稱 (AES) 金鑰，則必須將其設定為`:aes_gcm`。如果您使用的是非對稱 (RSA) 金鑰，則必須將其設定為`:rsa_oaep_sha1`。

`content_encryption_schema`-必須將此值設定為 `:AES_gcm_no_padding`。

2. 必須在用戶端建構中指定安全性設定檔。建立新的時候`EncryptionV2::Client`，您需要提供的值`security_profile`。安全性設定檔參數會決定讀取使用舊版 V1 所寫入物件的支援。`Encryption::Client`有兩個值：`:V2` 和 `:v2` 和舊版。若要支援移轉，請將設定為`security_profile :v2_`和舊版。使用 `:v2` 僅用於新的應用程序開發。

3. AWS KMS key 依預設，會強制執行 ID。在 V1 中`Encryption::Client`，未提供`kms_key_id`用來建立用戶端的 AWS KMS Decrypt call。AWS KMS 可以從元數據中獲取此信息，並將其添加到對稱密文 Blob 中。在 V2 中，`kms_key_id` 會傳遞至「解密」呼叫，如果呼叫不符合用來加 AWS KMS 密物件的金鑰，則呼叫會失敗。如果您的代碼以前依賴於不設置特定的`kms_key_id`，則可以`kms_key_id: :kms_allow_decrypt_with_any_cmk`在客戶端創建時設置或在`get_object`調用時進`kms_allow_decrypt_with_any_cmk: true`行設置。

範例：使用對稱 (AES) 金鑰

遷移前

```
client = Aws::S3::Encryption::Client.new(encryption_key: aes_key)
client.put_object(bucket: bucket, key: key, body: secret_data)
resp = client.get_object(bucket: bucket, key: key)
```

移轉後

```
client = Aws::S3::EncryptionV2::Client.new(
  encryption_key: rsa_key,
  key_wrap_schema: :rsa_oaep_sha1, # the key_wrap_schema must be rsa_oaep_sha1 for
  asymmetric keys
```

```
content_encryption_schema: :aes_gcm_no_padding,  
security_profile: :v2_and_legacy # to allow reading/decrypting objects encrypted by  
the V1 encryption client  
)  
client.put_object(bucket: bucket, key: key, body: secret_data) # No changes  
resp = client.get_object(bucket: bucket, key: key) # No changes
```

範例：AWS KMS 搭配使用

遷移前

```
client = Aws::S3::Encryption::Client.new(kms_key_id: kms_key_id)  
client.put_object(bucket: bucket, key: key, body: secret_data)  
resp = client.get_object(bucket: bucket, key: key)
```

移轉後

```
client = Aws::S3::EncryptionV2::Client.new(  
  kms_key_id: kms_key_id,  
  key_wrap_schema: :kms_context, # the key_wrap_schema must be kms_context for KMS keys  
  content_encryption_schema: :aes_gcm_no_padding,  
  security_profile: :v2_and_legacy # to allow reading/decrypting objects encrypted by  
  the V1 encryption client  
)  
client.put_object(bucket: bucket, key: key, body: secret_data) # No changes  
resp = client.get_object(bucket: bucket, key: key) # No change
```

範例：在 AWS KMS 沒有 KMS_id 的情況下使用

遷移前

```
client = Aws::S3::Encryption::Client.new(kms_key_id: kms_key_id)  
client.put_object(bucket: bucket, key: key, body: secret_data)  
resp = client.get_object(bucket: bucket, key: key)
```

移轉後

```
client = Aws::S3::EncryptionV2::Client.new(  
  kms_key_id: kms_key_id,  
  key_wrap_schema: :kms_context, # the key_wrap_schema must be kms_context for KMS keys  
  content_encryption_schema: :aes_gcm_no_padding,
```

```
    security_profile: :v2_and_legacy # to allow reading/decrypting objects encrypted by
    the V1 encryption client
  )
  client.put_object(bucket: bucket, key: key, body: secret_data) # No changes
  resp = client.get_object(bucket: bucket, key: key, kms_allow_decrypt_with_any_cmk:
    true) # To allow decrypting with any cmk
```

移轉後替代方案

如果您只使用 S2 加密用戶端讀取和解密 (絕不寫入和加密) 物件，請使用此程式碼。

```
client = Aws::S3::EncryptionV2::Client.new(
  kms_key_id: :kms_allow_decrypt_with_any_cmk, # set kms_key_id to allow all get_object
  requests to use any cmk
  key_wrap_schema: :kms_context, # the key_wrap_schema must be kms_context for KMS keys
  content_encryption_schema: :aes_gcm_no_padding,
  security_profile: :v2_and_legacy # to allow reading/decrypting objects encrypted by
  the V1 encryption client
)
resp = client.get_object(bucket: bucket, key: key) # No change
```


文件歷史記錄

下表說明本的重要變更。如需獲得此文件更新的通知，您可以訂閱 [RSS 摘要](#)。

變更	描述	日期
表格的內容	更新了目錄，讓程式碼範例更容易存取。	2023 年 6 月 1 日
IAM 務務務務務務	更新了指南以符合 IAM 最佳實務。如需詳細資訊，請參閱 IAM 中的安全最佳實務 。開始使用的更新。	2023 年 5 月 8 日
一般更新	更新相關外部資源的歡迎頁面。還更新了 v2.3 所需的最低 Ruby 版本。更新 AWS Key Management Service 章節以反映術語更新。更新了 REPL 實用程序的使用信息以清晰起見。	2022 年 8 月 9 日
修正斷開的連結	修正了破碎的例子鏈接。移除多餘的提示和技巧頁面；重新導向至 Amazon EC2 範例內容。包含在程式碼範例儲存庫中可用 GitHub 的程式碼範例清單。	2022 年 8 月 3 日
取得所有亞馬遜 RDS 安全群組的相關資訊	新增有關淘汰 EC2-Classic 的注意事項。	2022 年 7 月 26 日
SDK 指標	移除有關為企業 Support 啟用 SDK 指標的資訊，該資訊已被取代。	2022 年 1 月 28 日

本文為英文版的機器翻譯版本，如內容有任何歧義或不一致之處，概以英文版為準。