

Panduan Developerr

# AWS SDK for Ruby



# AWS SDK for Ruby: Panduan Developer

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

Merek dagang dan tampilan dagang Amazon tidak boleh digunakan sehubungan dengan produk atau layanan apa pun yang bukan milik Amazon, dengan cara apa pun yang dapat menyebabkan kebingungan antara para pelanggan, atau dengan cara apa pun yang menghina atau mendiskreditkan Amazon. Semua merek dagang lain yang tidak dimiliki oleh Amazon merupakan kekayaan masing-masing pemiliknya, yang mungkin berafiliasi, terkait dengan, atau disponsori oleh Amazon, atau mungkin tidak.

---

# Table of Contents

Apa AWS SDK for Ruby? .....	1
Dokumentasi dan sumber daya tambahan .....	1
Menerapkan ke Cloud AWS .....	2
Pemeliharaan dan dukungan untuk versi utama SDK .....	2
Memulai .....	3
Otentikasi SDK dengan AWS .....	3
Memulai sesi portal AWS akses .....	4
Informasi autentikasi lebih lanjut .....	5
Instal SDK .....	5
Prasyarat .....	6
Instalasi SDK .....	6
Halo tutorial .....	7
Tulis kode .....	7
Menjalankan program .....	8
Catatan untuk pengguna Windows .....	9
Langkah selanjutnya .....	9
Gunakan AWS Cloud9 dengan SDK .....	9
Langkah 1: Atur Akun AWS untuk menggunakan AWS Cloud9 .....	10
Langkah 2: Atur lingkungan AWS Cloud9 pengembangan Anda .....	10
Langkah 3: Atur AWS SDK for Ruby .....	11
Langkah 4: Unduh kode contoh .....	12
Langkah 5: Jalankan kode .....	13
Konfigurasi SDK .....	15
Rantai penyedia kredensi .....	15
Membuat token AWS STS akses .....	17
Mengatur Wilayah .....	17
Mengatur Wilayah menggunakan config file bersama .....	18
Mengatur Wilayah menggunakan variabel lingkungan .....	18
Mengatur Wilayah dengan <code>Aws.config</code> .....	18
Mengatur Wilayah di klien atau objek sumber daya .....	19
Menetapkan titik akhir yang tidak standar .....	19
Menggunakan SDK .....	20
Gunakan utilitas REPL .....	20
Prasyarat .....	20

Pengaturan bundler .....	21
Menjalankan REPL .....	21
Menggunakan SDK dengan Ruby on Rails .....	22
Tip debugging: Dapatkan informasi pelacakan kawat dari klien .....	22
Respons dan kesalahan klien rintisan .....	23
Tanggapan klien yang mematahkan .....	23
Kesalahan klien yang mematahkan .....	25
Paginasi .....	25
Respons paged dapat dihitung .....	25
Menangani tanggapan paged secara manual .....	26
Kelas data paged .....	26
Pelayan .....	26
Memanggil pelayan .....	27
Kegagalan tunggu kegagalan .....	27
Mengkonfigurasi pelayan .....	28
Memperluas pelayan .....	28
Tentukan perilaku coba lagi klien .....	29
Bermigrasi dari versi 1 atau 2 ke AWS SDK versi 3 untuk Ruby .....	30
ide-by-side Penggunaan S .....	30
Perbedaan umum .....	30
Perbedaan klien .....	31
Perbedaan sumber daya .....	32
Bekerja dengan AWS layanan .....	34
Contoh kode dengan panduan .....	34
Contoh AWS CloudTrail .....	35
CloudWatch Contoh Amazon .....	41
Contoh AWS CodeBuild .....	74
Contoh Amazon EC2 .....	77
Contoh AWS Elastic Beanstalk .....	131
AWS Identity and Access Management(IAM) Contoh .....	135
Contoh AWS KMS .....	170
Contoh AWS Lambda .....	174
Contoh Amazon Polly .....	179
Contoh Amazon RDS .....	184
Contoh Amazon SES .....	191
Contoh Amazon SNS .....	196

Amazon SQS Contoh .....	201
WorkDocs Contoh Amazon .....	229
Contoh kode .....	233
Tindakan dan skenario .....	233
CloudTrail .....	234
CloudWatch .....	239
DynamoDB .....	251
Amazon EC2 .....	277
Elastic Beanstalk .....	311
EventBridge .....	316
AWS Glue .....	338
IAM .....	366
Kinesis .....	423
AWS KMS .....	426
Lambda .....	430
Amazon Polly .....	450
Amazon RDS .....	454
Amazon S3 .....	459
Amazon SES .....	488
Amazon SES API v2 .....	494
Amazon SNS .....	495
Amazon SQS .....	505
AWS STS .....	518
Amazon WorkDocs .....	520
Contoh lintas layanan .....	522
Buat aplikasi untuk menganalisis umpan balik pelanggan .....	523
Keamanan .....	524
Perlindungan Data .....	524
Manajemen Identitas dan Akses .....	525
Validasi Kepatuhan .....	526
Ketangguhan .....	527
Keamanan Infrastruktur .....	528
Menegakkan versi TLS minimum .....	528
Memeriksa versi OpenSSL .....	528
Meningkatkan dukungan TLS .....	529
Migrasi Klien Enkripsi S3 .....	529

Ikhtisar Migrasi .....	529
Perbarui Klien yang Ada untuk Membaca Format Baru .....	529
Migrasi Klien Enkripsi dan Dekripsi ke V2 .....	531
Riwayat Dokumen .....	535
.....	dxxxvii

# Apa AWS SDK for Ruby?

Selamat datang di AWS SDK for Ruby Developer Guide. AWSSDK for Ruby, Amazon Elastic Compute Cloud (Amazon S3), Amazon Elastic Compute Cloud (Amazon EC2)Layanan AWS, Amazon Elastic Compute Cloud (Amazon EC2), Amazon Elastic Compute Cloud (Amazon EC2), Amazon Elastic Compute Cloud (Amazon EC2), Amazon DynamoDB.

AWSSDK for Ruby Developer Guide menyediakan informasi tentang cara menginstal, mengatur, dan menggunakan AWS SDK for Ruby untuk membuat aplikasi Ruby yang menggunakan. Layanan AWS

[Memulai AWS SDK for Ruby](#)

## Dokumentasi dan sumber daya tambahan

Untuk AWS sumber daya selengkapnya, lihat berikut:

- [AWSPanduan Referensi SDK dan Alat](#) - Berisi pengaturan, fitur, dan konsep dasar lainnya yang umum di antara SDK AWS
- [AWS SDK for RubyReferensi API - Versi 3](#)
- [AWSContoh Kode Repositori](#) pada GitHub
- [RubyGems.org](#) - Versi terbaru SDK dimodulasi menjadi permata khusus layanan yang tersedia di sini
  - [Layanan yang Didukung](#) - Mencantumkan semua permata yang didukung AWS SDK for Ruby
- AWSSDK untuk sumber Ruby pada: GitHub
  - [Sumber](#) dan [README](#)
  - [Ubah log di bawah setiap permata](#)
  - [Pindah dari v2 ke v3](#)
  - [Masalah](#)
  - [Catatan peningkatan inti](#)
- [Blog pengembang](#)
- [Saluran Gitter](#)
- [@awsforruby](#) di Twitter

## Menerapkan ke Cloud AWS

Anda dapat menggunakan Layanan AWS seperti AWS Elastic Beanstalk, AWS OpsWorks, dan AWS CodeDeploy untuk menyebarkan aplikasi Anda ke AWS Cloud. Untuk menerapkan aplikasi Ruby dengan Elastic Beanstalk, lihat [Deploying Elastic Beanstalk Applications di Ruby Using EB CLI dan Git di dalam Developer Guide](#). AWS Elastic Beanstalk Untuk men-deploy aplikasi Ruby on Rails AWS OpsWorks, lihat [Deploying Ruby on Rails Applications ke AWS OpsWorks](#) Untuk ikhtisar layanan AWS penyebaran, lihat [Ikhtisar Opsi Penyebaran aktif](#). AWS

## Pemeliharaan dan dukungan untuk versi utama SDK

Untuk informasi tentang pemeliharaan dan dukungan untuk versi utama SDK dan dependensi yang mendasarinya, lihat berikut di [Panduan Referensi SDK dan Alat AWS](#):

- [AWS Kebijakan Pemeliharaan SDK dan Alat](#)
- [AWS Matriks Support Versi SDK dan Alat](#)



# Memulai AWS SDK for Ruby

Pelajari cara menginstal, mengatur, dan menggunakan SDK untuk membuat aplikasi Ruby untuk mengakses AWS sumber daya secara terprogram.

Topik

- [Otentikasi SDK dengan AWS](#)
- [Instal AWS SDK for Ruby](#)
- [Halo tutorial untuk AWS SDK for Ruby](#)
- [Gunakan AWS Cloud9 dengan AWS SDK for Ruby](#)

## Otentikasi SDK dengan AWS

Anda harus menetapkan bagaimana kode Anda mengautentikasi dengan AWS ketika mengembangkan dengan Layanan AWS. Anda dapat mengonfigurasi akses terprogram ke AWS sumber daya dengan cara yang berbeda tergantung pada lingkungan dan AWS akses yang tersedia untuk Anda.

Untuk memilih metode autentikasi dan mengkonfigurasinya untuk SDK, lihat [Autentikasi dan akses](#) di Panduan Referensi AWS SDK dan Alat.

Kami menyarankan agar pengguna baru yang berkembang secara lokal dan tidak diberi metode autentikasi oleh majikan mereka harus menyiapkan AWS IAM Identity Center. Metode ini termasuk menginstal AWS CLI untuk kemudahan konfigurasi dan untuk masuk secara teratur ke portal AWS akses. Jika Anda memilih metode ini, lingkungan Anda harus berisi elemen-elemen berikut setelah Anda menyelesaikan prosedur untuk [otentikasi IAM Identity Center](#) di AWSSDK dan Tools Reference Guide:

- Itu AWS CLI, yang Anda gunakan untuk memulai sesi AWS akses portal sebelum Anda menjalankan aplikasi Anda.
- [AWSconfigFile bersama](#) yang memiliki [default] profil dengan serangkaian nilai konfigurasi yang dapat direferensikan dari SDK. Untuk menemukan lokasi file ini, lihat [Lokasi file bersama](#) di Panduan Referensi AWS SDK dan Alat.
- `configFile` bersama menetapkan [region](#) pengaturan. Ini menetapkan default Wilayah AWS yang digunakan SDK untuk AWS permintaan. Wilayah ini digunakan untuk permintaan layanan SDK yang tidak ditentukan dengan Wilayah untuk digunakan.

- SDK menggunakan [konfigurasi penyedia token SSO](#) profil untuk memperoleh kredensi sebelum mengirim permintaan ke AWS. `sso_role_name` Nilai, yang merupakan peran IAM yang terhubung ke set izin IAM Identity Center, memungkinkan akses ke yang Layanan AWS digunakan dalam aplikasi Anda.

`configFile` contoh berikut menunjukkan profil default yang diatur dengan konfigurasi penyedia token SSO. `sso_session` Pengaturan profil mengacu pada [sso-sessionbagian](#) bernama. `sso-session` Bagian ini berisi pengaturan untuk memulai sesi portal AWS akses.

```
[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
```

AWSSDK for Ruby tidak memerlukan paket tambahan (seperti `SSO` dan `SSO0IDC`) untuk ditambahkan ke aplikasi Anda untuk menggunakan otentikasi IAM Identity Center.

## Memulai sesi portal AWS akses

Sebelum menjalankan aplikasi yang mengakses Layanan AWS, Anda memerlukan sesi portal AWS akses aktif agar SDK menggunakan autentikasi IAM Identity Center untuk menyelesaikan kredensi. Bergantung pada panjang sesi yang dikonfigurasi, akses Anda pada akhirnya akan kedaluwarsa dan SDK akan mengalami kesalahan autentikasi. Untuk masuk ke portal AWS akses, jalankan perintah berikut di AWS CLI.

```
aws sso login
```

Jika Anda mengikuti panduan dan memiliki pengaturan profil default, Anda tidak perlu memanggil perintah dengan `--profile` opsi. Jika konfigurasi penyedia token SSO Anda menggunakan profil bernama, perintahnya adalah `aws sso login --profile named-profile`.

Untuk menguji secara opsional jika Anda sudah memiliki sesi aktif, jalankan AWS CLI perintah berikut.

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

Jika sesi Anda aktif, respons terhadap perintah ini melaporkan akun IAM Identity Center dan set izin yang dikonfigurasi dalam `config` file bersama.

### Note

Jika Anda sudah memiliki sesi portal AWS akses aktif dan menjalankannya `aws sso login`, Anda tidak akan diminta untuk memberikan kredensi.

Proses login mungkin meminta Anda untuk mengizinkan AWS CLI akses ke data Anda. Karena AWS CLI dibangun di atas SDK untuk Python, pesan izin mungkin berisi variasi `botocore` nama.

## Informasi autentikasi lebih lanjut

Pengguna manusia, juga dikenal sebagai identitas manusia, adalah orang, administrator, pengembang, operator, dan konsumen aplikasi Anda. Mereka harus memiliki identitas untuk mengakses AWS lingkungan dan aplikasi Anda. Pengguna manusia yang merupakan anggota organisasi Anda - itu berarti Anda, pengembang - dikenal sebagai identitas tenaga kerja.

Gunakan kredensi sementara saat mengakses AWS. Anda dapat menggunakan penyedia identitas bagi pengguna manusia untuk menyediakan akses federasi ke AWS akun dengan mengasumsikan peran, yang memberikan kredensi sementara. Untuk manajemen akses terpusat, kami menyarankan Anda menggunakan AWS IAM Identity Center (IAM Identity Center) untuk mengelola akses ke akun dan izin Anda dalam akun tersebut. Untuk alternatif lebih lanjut, lihat hal berikut:

- Untuk mempelajari lebih lanjut tentang praktik terbaik, lihat [Praktik terbaik keamanan di IAM](#) dalam Panduan Pengguna IAM.
- Untuk membuat AWS kredensi jangka pendek, lihat [Kredensi Keamanan Sementara](#) di Panduan Pengguna IAM.
- Untuk mempelajari tentang penyedia kredensi AWS SDK for Ruby lainnya, lihat penyedia [kredensi standar](#) di Panduan Referensi AWS SDK dan Alat.

## Instal AWS SDK for Ruby

Bagian ini mencakup prasyarat dan petunjuk instalasi untuk SDK for Ruby. AWS

## Prasyarat

Sebelum Anda menggunakan AWS SDK for Ruby, Anda harus mengotentikasi dengan AWS. Untuk informasi tentang menyiapkan autentikasi, lihat [Otentikasi SDK dengan AWS](#).

## Instalasi SDK

Anda dapat menginstal AWS SDK for Ruby seperti yang Anda lakukan setiap Ruby gem. Permata yang tersedia di [RubyGems](#). AWSSDK for Ruby dirancang untuk menjadi modular dan dipisahkan oleh Layanan AWS. Instalasi seluruh `aws-sdk` permata besar dan mungkin memakan waktu lebih dari satu jam.

Kami sarankan hanya menginstal permata untuk yang Layanan AWS Anda gunakan. Ini dinamai like `aws-sdk-service_abbreviation` dan daftar lengkapnya ditemukan di tabel [Layanan yang Didukung](#) dari file AWS SDK for Ruby README. Misalnya, permata untuk berinteraksi dengan layanan Amazon S3 tersedia langsung di [aws-sdk-s3](#)

## Manajer versi Ruby

Alih-alih menggunakan sistem Ruby, sebaiknya gunakan manajer versi Ruby seperti berikut ini:

- [RVM](#)
- [chruby](#)
- [rbenv](#)

Misalnya, jika Anda menggunakan sistem operasi Amazon Linux 2, perintah berikut dapat digunakan untuk memperbarui RVM, mencantumkan versi Ruby yang tersedia, lalu pilih versi yang ingin Anda gunakan untuk pengembangan dengan AWS SDK for Ruby. Versi Ruby minimum yang dibutuhkan adalah 2.3.

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

## Bundler

Jika Anda menggunakan [Bundler](#), perintah berikut menginstal AWS SDK for Ruby gem untuk Amazon S3:

## 1. Instal Bundler dan buat: Gemfile

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

2. Buka yang dibuat Gemfile dan tambahkan gem baris untuk setiap permata AWS layanan yang akan digunakan kode Anda. Untuk mengikuti contoh Amazon S3, tambahkan baris berikut ke bagian bawah file:

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

3. Simpan Gemfile.

4. Instal dependensi yang ditentukan dalam: Gemfile

```
$ bundle install
```

## Halo tutorial untuk AWS SDK for Ruby

Sapa Amazon S3 menggunakan AWS SDK for Ruby. Contoh berikut menampilkan daftar bucket Amazon S3 Anda.

### Tulis kode

Salin dan tempel kode berikut ke file sumber baru. Beri nama file `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 dirancang untuk modular dan dipisahkan oleh. Layanan AWS Setelah permata diinstal, `require` pernyataan di bagian atas file sumber Ruby Anda mengimpor kelas dan metode AWS SDK untuk layanan Amazon S3. Untuk daftar lengkap AWS service gems yang tersedia, lihat tabel [Layanan yang Didukung](#) dari file AWS SDK for Ruby README.

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

## Menjalankan program

Buka prompt perintah untuk menjalankan program Ruby Anda. Sintaks perintah khas untuk menjalankan program Ruby adalah:

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

Kode contoh ini tidak menggunakan argumen. Untuk menjalankan kode ini, masukkan yang berikut ini ke command prompt:

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

## Catatan untuk pengguna Windows

Saat Anda menggunakan sertifikat SSL di Windows dan menjalankan kode Ruby Anda, Anda mungkin melihat kesalahan yang mirip dengan yang berikut ini.

```
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'
...

```

Untuk memperbaiki masalah ini, tambahkan baris berikut ke file sumber Ruby Anda, di suatu tempat sebelum AWS panggilan pertama Anda.

```
Aws.use_bundled_cert!
```

Jika Anda hanya menggunakan `aws-sdk-s3` permata dalam program Ruby Anda dan Anda ingin menggunakan sertifikat yang dibundel, Anda juga perlu menambahkan permata `aws-sdk-core`

## Langkah selanjutnya

Untuk menguji banyak operasi Amazon S3 lainnya, lihat [Repositori Contoh AWS Kode](#) di GitHub

## Gunakan AWS Cloud9 dengan AWS SDK for Ruby

AWS Cloud9 adalah lingkungan pengembangan terintegrasi berbasis web (IDE) yang berisi kumpulan alat yang Anda gunakan untuk kode, membangun, menjalankan, menguji, debug, dan melepaskan perangkat lunak di cloud. Anda dapat menggunakan AWS Cloud9 AWS SDK for Ruby untuk menulis dan menjalankan kode Ruby Anda dengan menggunakan browser. AWS Cloud9 termasuk alat seperti editor kode dan terminal. Karena AWS Cloud9 IDE berbasis cloud, Anda dapat mengerjakan proyek Anda dari kantor, rumah, atau di mana saja dengan menggunakan mesin yang terhubung ke internet. Untuk informasi umum tentang AWS Cloud9, lihat [Panduan AWS Cloud9 Pengguna](#).

Ikuti petunjuk berikut untuk mengatur AWS Cloud9 AWS SDK for Ruby:

- [Langkah 1: Atur Anda Akun AWS untuk menggunakan AWS Cloud9](#)
- [Langkah 2: Atur lingkungan AWS Cloud9 pengembangan Anda](#)
- [Langkah 3: Atur AWS SDK for Ruby](#)
- [Langkah 4: Unduh kode contoh](#)
- [Langkah 5: Jalankan kode](#)

## Langkah 1: Atur Anda Akun AWS untuk menggunakan AWS Cloud9

Untuk menggunakan AWS Cloud9, masuk ke AWS Cloud9 konsol dari AWS Management Console.

### Note

Jika Anda menggunakan AWS IAM Identity Center untuk mengautentikasi, Anda mungkin perlu menambahkan izin yang diperlukan `iam:ListInstanceProfilesForRole` ke kebijakan yang dilampirkan pengguna di konsol IAM.

Untuk menyiapkan entitas IAM di AWS akun Anda agar dapat mengakses AWS Cloud9 dan masuk ke AWS Cloud9 konsol, lihat [Pengaturan Tim AWS Cloud9](#) di Panduan AWS Cloud9 Pengguna.

## Langkah 2: Atur lingkungan AWS Cloud9 pengembangan Anda

Setelah Anda masuk ke AWS Cloud9 konsol, gunakan konsol untuk membuat lingkungan AWS Cloud9 pengembangan. Setelah Anda membuat lingkungan, AWS Cloud9 membuka IDE untuk lingkungan itu.

Untuk detailnya, lihat [Membuat Lingkungan AWS Cloud9 di](#) dalam Panduan AWS Cloud9 Pengguna.

### Note

Saat Anda membuat lingkungan Anda di konsol untuk pertama kalinya, kami sarankan Anda memilih opsi untuk membuat instance baru untuk lingkungan (EC2). Opsi ini memberi tahu AWS Cloud9 untuk membuat lingkungan, meluncurkan instans Amazon EC2, dan kemudian menghubungkan instans baru dengan lingkungan baru. Ini adalah cara tercepat untuk mulai menggunakan AWS Cloud9.



Jika terminal belum terbuka di IDE, buka. Pada bilah menu di IDE, pilih Window, New Terminal. Anda dapat menggunakan jendela terminal untuk menginstal alat dan membangun aplikasi Anda.

## Langkah 3: Atur AWS SDK for Ruby

Setelah AWS Cloud9 membuka IDE untuk lingkungan pengembangan Anda, gunakan jendela terminal untuk mengatur AWS SDK for Ruby di lingkungan Anda.

Anda dapat menginstal AWS SDK for Ruby seperti yang Anda lakukan setiap Ruby gem. Permata yang tersedia di [RubyGems](#). AWSSDK for Ruby dirancang untuk menjadi modular dan dipisahkan oleh. Layanan AWS Instalasi seluruh `aws-sdk` permata besar dan mungkin memakan waktu lebih dari satu jam.

Kami sarankan hanya menginstal permata untuk yang Layanan AWS Anda gunakan. Ini dinamai like `aws-sdk-service_abbreviation` dan daftar lengkapnya ditemukan di tabel [Layanan yang Didukung](#) dari file AWS SDK for Ruby README. Misalnya, permata untuk berinteraksi dengan layanan Amazon S3 tersedia langsung di. [aws-sdk-s3](#)

## Manajer versi Ruby

Alih-alih menggunakan sistem Ruby, sebaiknya gunakan manajer versi Ruby seperti berikut ini:

- [RVM](#)
- [chruby](#)
- [rbenv](#)

Misalnya, jika Anda menggunakan sistem operasi Amazon Linux 2, perintah berikut dapat digunakan untuk memperbarui RVM, mencantumkan versi Ruby yang tersedia, lalu pilih versi yang ingin Anda gunakan untuk pengembangan dengan AWS SDK for Ruby. Versi Ruby minimum yang dibutuhkan adalah 2.3.

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

## Bundler

Jika Anda menggunakan [Bundler](#), perintah berikut menginstal AWS SDK for Ruby gem untuk Amazon S3:

1. Instal Bundler dan buat: Gemfile

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

2. Buka yang dibuat Gemfile dan tambahkan gem baris untuk setiap permata AWS layanan yang akan digunakan kode Anda. Untuk mengikuti contoh Amazon S3, tambahkan baris berikut di bagian bawah file:

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

3. Simpan Gemfile.
4. Instal dependensi yang ditentukan dalam: Gemfile

```
$ bundle install
```

## Langkah 4: Unduh kode contoh

Gunakan jendela terminal untuk mengunduh kode contoh AWS SDK for Ruby ke lingkungan AWS Cloud9 pengembangan.

Untuk mengunduh salinan semua contoh kode yang digunakan dalam dokumentasi AWS SDK resmi ke direktori root lingkungan Anda, jalankan perintah berikut:

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

Contoh kode untuk AWS SDK for Ruby terletak di `ENVIRONMENT_NAME/aws-doc-sdk-examples/ruby` direktori, di mana `ENVIRONMENT_NAME` adalah nama lingkungan pengembangan Anda.

Untuk mengikuti menggunakan contoh Amazon S3, sebaiknya mulai dengan contoh `ENVIRONMENT_NAME/aws-doc-sdk-examples/ruby/example_code/s3/bucket_list.rb` kode. Gunakan jendela terminal untuk menavigasi ke `s3` direktori dan daftar file.

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

Untuk membuka file diAWS Cloud9, Anda dapat mengklik `bucket_list.rb` langsung di jendela terminal.

Untuk dukungan lebih lanjut dalam memahami contoh kode, lihat [AWSSDK for Ruby Code Examples](#).

## Langkah 5: Jalankan kode

Untuk menjalankan kode di lingkungan AWS Cloud9 pengembangan Anda, pilih tombol Jalankan di bilah menu atas. AWS Cloud9 akan secara otomatis mendeteksi ekstensi `.rb` file dan menggunakan runner Ruby untuk menjalankan kode. Untuk informasi selengkapnya tentang menjalankan kode AWS Cloud9, lihat [Menjalankan Kode Anda](#) di Panduan AWS Cloud9 Pengguna.

Di tangkapan layar berikut, perhatikan area dasar ini:

- 1: Jalankan. Tombol Run terletak di bilah menu atas. Ini akan membuka tab baru untuk hasil Anda.

### Note

Anda juga dapat secara manual membuat konfigurasi Jalankan baru. Pada bilah menu, pilih Jalankan, Jalankan Konfigurasi, Jalankan Konfigurasi Baru.

- 2: Perintah. AWS Cloud9 mengisi kotak teks Command dengan path dan nama file ke file yang Anda jalankan. Jika kode Anda mengharapkan parameter baris perintah untuk diteruskan, ini dapat ditambahkan ke baris perintah dengan cara yang sama seperti yang Anda lakukan ketika menjalankan kode melalui jendela terminal.
- 3: Pelari. AWS Cloud9 mendeteksi bahwa ekstensi file Anda `.rb` dan memilih Ruby Runner untuk menjalankan kode Anda.

```
Go Run Tools Window Support Preview Run Share
bucket_list.rb
9 require "aws-sdk-s3"
10
11 # Wraps Amazon S3 resource actions.
12 class BucketListWrapper
13   attr_reader :s3_resource
14
15   # @param s3_resource [Aws::S3::Resource] An Amazon S3 resource.
16   def initialize(s3_resource)
17     @s3_resource = s3_resource
18   end
19
20   # Lists buckets for the current account.
21   #
```

1

2

3

bash - "ip-172-31-35-38.ec" x aws-doc-sdk-examples/ru x

Run Command: aws-doc-sdk-examples/ruby/example\_code/s3/bucket\_list.rb Runner: Ruby CWD ENV

Found these buckets:

Output apa pun yang dihasilkan dari kode yang sedang berjalan ditampilkan di tab.

# Konfigurasi AWS SDK for Ruby

Pelajari cara mengonfigurasi AWS SDK for Ruby. Anda harus menetapkan bagaimana kode Anda mengautentikasi dengan AWS ketika Anda mengembangkan dengan Layanan AWS. Anda juga harus mengatur yang ingin Wilayah AWS Anda gunakan.

## Rantai penyedia kredensi

Semua SDK memiliki serangkaian tempat (atau sumber) yang mereka periksa untuk mendapatkan kredensial yang valid untuk digunakan untuk membuat permintaan ke file. Layanan AWS Setelah kredensial yang valid ditemukan, pencarian dihentikan. Pencarian sistematis ini disebut rantai penyedia kredensi default.

Untuk setiap langkah dalam rantai, ada berbagai cara untuk mengatur nilai. Menetapkan nilai secara langsung dalam kode selalu diutamakan, diikuti dengan pengaturan sebagai variabel lingkungan, dan kemudian di file bersama `AWSconfig`. Untuk informasi selengkapnya, [lihat Prioritas pengaturan di AWS SDK dan Panduan Referensi](#) Alat.

AWSSDK dan Tools Reference Guide memiliki informasi tentang pengaturan konfigurasi SDK yang digunakan oleh semua AWS SDK dan. AWS CLI Untuk mempelajari lebih lanjut tentang cara mengonfigurasi SDK melalui `AWS config` file bersama, lihat File [konfigurasi dan kredensial bersama](#). Untuk mempelajari lebih lanjut tentang cara mengonfigurasi SDK melalui pengaturan variabel lingkungan, lihat [Dukungan variabel lingkungan](#).

Untuk mengautentikasi AWS, AWS SDK for Ruby memeriksa penyedia kredensi dalam urutan yang tercantum dalam tabel berikut.

Penyedia kredensi berdasarkan prioritas	AWS Panduan Referensi SDK dan Alat	AWS SDK for Ruby Referensi API
Kredensial statis	<a href="#">AWS kunci akses</a>	<a href="#">Aws::Credentials</a> <a href="#">Aws::SharedCredentials</a>
Token identitas web dari AWS Security Token Service (AWS STS)	<a href="#">Asumsikan penyedia kredensi peran</a>	<a href="#">Aws::AssumeRoleWebIdentityCredentials</a>

Penyedia kredensi berdasarkan prioritas	AWS Panduan Referensi SDK dan Alat	AWS SDK for Ruby Referensi API
	Menggunakan <code>role_arn</code> , <code>role_session_name</code> , dan <code>web_identity_token_file</code>	
AWS IAM Identity Center. Dalam panduan ini, lihat <a href="#">Otentikasi SDK dengan AWS</a> .	<a href="#">Penyedia kredensi Pusat Identitas IAM</a>	<a href="#">Aws::SSOCredentials</a>
Penyedia entitas tepercaya (seperti <code>AWS_ROLE_ARN</code> ). Dalam panduan ini, lihat <a href="#">Membuat token AWS STS akses</a> .	<a href="#">Asumsikan penyedia kredensi peran</a> Menggunakan <code>role_arn</code> dan <code>role_session_name</code>	<a href="#">Aws::AssumeRoleCredentials</a>
Penyedia kredensi proses	<a href="#">Penyedia kredensi proses</a>	<a href="#">Aws::ProcessCredentials</a>
Kredensi Amazon Elastic Container Service (Amazon ECS)	<a href="#">Penyedia kredensi kontainer</a>	<a href="#">Aws::ECSCredentials</a>
Kredensi profil instans Amazon Elastic Compute Cloud (Amazon EC2) (penyedia kredensi IMDS)	<a href="#">Penyedia kredensi IMDS</a>	<a href="#">Aws::InstanceProfileCredentials</a>

Jika variabel `AWS_SDK_CONFIG_OPT_OUT` lingkungan AWS SDK for Ruby disetel, file bersama, `~/.aws/config` biasanya AWS config di, tidak akan diuraikan untuk kredensialnya.

Jika Anda mengikuti pendekatan yang disarankan bagi pengguna baru untuk memulai, Anda menyiapkan AWS IAM Identity Center [Otentikasi SDK dengan AWS](#) autentikasi selama topik Memulai. Metode otentikasi lainnya berguna untuk situasi yang berbeda. Untuk menghindari risiko

keamanan, kami sarankan untuk selalu menggunakan kredensi jangka pendek. Untuk prosedur metode otentikasi lainnya, lihat [Otentikasi dan akses di Panduan Referensi AWSSDK dan Alat](#).

## Membuat token AWS STS akses

Dengan asumsi peran melibatkan penggunaan seperangkat kredensial keamanan sementara yang dapat Anda gunakan untuk mengakses AWS sumber daya yang biasanya tidak dapat Anda akses. Kredensial sementara ini terdiri dari access key ID, secret access key, dan token keamanan. Anda dapat menggunakan [Aws::AssumeRoleCredentials](#) metode ini untuk membuat token akses AWS Security Token Service (AWS STS).

Contoh berikut menggunakan token akses untuk membuat objek klien Amazon S3, di mana `linked::account::arn` adalah Amazon Resource Name (ARN) dari peran yang akan diasumsikan dan `session-name` merupakan pengidentifikasi untuk sesi peran yang diasumsikan.

```
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)
```

Untuk informasi selengkapnya tentang pengaturan `role_arn` atau `role_session_name`, atau tentang menyetelnya menggunakan AWS config file bersama, lihat [Mengasumsikan penyedia kredensi peran](#) di Panduan Referensi AWS SDK dan Alat.

## Mengatur Wilayah

Anda perlu mengatur Wilayah saat menggunakan sebagian besar Layanan AWS. AWSSDK for Ruby menelusuri Wilayah dengan urutan sebagai berikut:

1. [Mengatur Wilayah di klien atau objek sumber daya](#)
2. [Mengatur Wilayah dengan menggunakan `Aws.config`](#)
3. [Mengatur Wilayah dengan menggunakan variabel lingkungan](#)
4. [Mengatur Wilayah dengan menggunakan config file bersama](#)

Untuk informasi selengkapnya tentang `region` pengaturan, lihat [Wilayah AWS](#) di Panduan Referensi AWS SDK dan Alat. Sisa bagian ini menjelaskan cara mengatur Wilayah, dimulai dengan pendekatan yang paling umum.

## Mengatur Wilayah menggunakan **config** file bersama

Atur wilayah dengan mengatur `region` variabel dalam AWS config file bersama. Untuk informasi selengkapnya tentang config file bersama, lihat File [konfigurasi dan kredensial bersama](#) di Panduan Referensi AWSSDK dan Alat.

Contoh pengaturan nilai ini dalam config file:

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

configFile bersama tidak diperiksa jika variabel lingkungan `AWS_SDK_CONFIG_OPT_OUT` disetel.

## Mengatur Wilayah menggunakan variabel lingkungan

Mengatur Region dengan mengatur variabel `AWS_REGION` lingkungan.

Gunakan `export` perintah untuk mengatur variabel ini pada sistem berbasis Unix, seperti Linux atau macOS. Contoh berikut menetapkan Region ke `us-west-2`.

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

Untuk mengatur variabel ini pada Windows, gunakan `set` perintah. Contoh berikut menetapkan Region ke `us-west-2`.

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

## Mengatur Wilayah dengan **Aws.config**

Atur Region dengan menambahkan `region` nilai ke `Aws.config` hash. Contoh berikut memperbarui `Aws.config` hash untuk menggunakan `us-west-1` Wilayah.

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

Setiap klien atau sumber daya yang Anda buat nanti terikat ke Wilayah ini.



## Mengatur Wilayah di klien atau objek sumber daya

Atur Wilayah saat Anda membuat AWS klien atau sumber daya. Contoh berikut membuat objek sumber daya Amazon S3 di Wilayah. `us-west-1` Pilih Wilayah yang tepat untuk AWS sumber daya Anda. Objek klien layanan tidak dapat diubah, jadi Anda harus membuat klien baru untuk setiap layanan yang Anda ajukan permintaan dan untuk membuat permintaan ke layanan yang sama menggunakan konfigurasi yang berbeda.

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

## Menetapkan titik akhir yang tidak standar

Wilayah ini digunakan untuk membangun titik akhir SSL untuk digunakan untuk permintaan. AWS Jika Anda perlu menggunakan titik akhir yang tidak standar di Wilayah yang Anda pilih, tambahkan `endpoint` entri ke `Aws.config` Atau, atur `endpoint`: saat membuat klien layanan atau objek sumber daya. Contoh berikut membuat objek sumber daya Amazon S3 di titik akhir. `other_endpoint`

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

Untuk menggunakan titik akhir yang Anda pilih untuk permintaan API dan agar pilihan tersebut tetap ada, lihat opsi konfigurasi [titik akhir khusus layanan](#) di AWSSDK dan Panduan Referensi Alat.

# Menggunakan AWS SDK for Ruby

Bagian ini menyediakan informasi tentang pengembangan perangkat lunak dengan AWS SDK for Ruby, termasuk cara menggunakan beberapa fitur canggih SDK.

[Panduan Referensi AWS SDK dan Alat](#) juga berisi pengaturan, fitur, dan konsep dasar lainnya yang umum di antara banyak SDK. AWS

## Topik

- [Gunakan utilitas AWS SDK for Ruby REPL](#)
- [Menggunakan SDK dengan Ruby on Rails](#)
- [Tip debugging: Dapatkan informasi pelacakan kawat dari klien](#)
- [Respons dan kesalahan klien rintisan](#)
- [Paginasi](#)
- [Pelayan](#)
- [Tentukan perilaku coba lagi klien](#)
- [Bermigrasi dari versi 1 atau 2 ke AWS SDK versi 3 untuk Ruby](#)

## Gunakan utilitas AWS SDK for Ruby REPL

`aws-sdk` Permata mencakup antarmuka baris perintah interaktif Read-Eval-Print-Loop (REPL) di mana Anda dapat menguji SDK for Ruby dan segera melihat hasilnya. [SDK for Ruby gems tersedia RubyGems di .org.](#)

## Prasyarat

- [Instal AWS SDK for Ruby.](#)
- `aws-v3.rb` itu terletak di `aws-sdk-resources` permata. `aws-sdk-resources` Permata itu juga disertakan oleh `aws-sdk` permata utama.
- Anda akan membutuhkan perpustakaan xml. seperti `rexml` permata.
- Meskipun program ini bekerja dengan Interactive Ruby Shell (`irb`), kami menyarankan Anda menginstal `pry` permata, yang menyediakan lingkungan REPL yang lebih kuat.

## Pengaturan bundler

Jika Anda menggunakan [Bundler](#), pembaruan berikut untuk Anda Gemfile akan membahas permata prasyarat:

1. Buka Gemfile yang Anda buat saat menginstal AWS SDK for Ruby. Tambahkan baris berikut ke file:

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

2. Simpan Gemfile.
3. Instal dependensi yang ditentukan dalam: Gemfile

```
$ bundle install
```

## Menjalankan REPL

Anda dapat mengakses REPL dengan menjalankan `aws-v3.rb` dari baris perintah.

```
aws-v3.rb
```

Atau, Anda dapat mengaktifkan pencatatan kawat HTTP dengan menyetel flag verbose. HTTP wire logging memberikan informasi tentang komunikasi antara AWS SDK for AWS Ruby dan. Catatan, flag verbose juga menambahkan overhead yang dapat membuat kode Anda berjalan lebih lambat.

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

SDK for Ruby mencakup kelas klien yang menyediakan antarmuka ke file. Layanan AWS Setiap kelas klien mendukung yang tertentu Layanan AWS. Dalam REPL, setiap kelas layanan memiliki helper yang mengembalikan objek klien baru untuk berinteraksi dengan layanan itu. Nama pembantu akan menjadi nama layanan yang dikonversi ke huruf kecil. Misalnya, nama-nama objek `s3` pembantu Amazon S3 dan Amazon EC2 adalah dan, masing-masing. `ec2` Untuk membuat daftar bucket Amazon S3 di akun Anda, Anda dapat masuk `s3.list_buckets` ke prompt.

Anda dapat mengetik `quit` ke prompt REPL untuk keluar.

## Menggunakan SDK dengan Ruby on Rails

[Ruby on Rails](#) menyediakan kerangka pengembangan web yang membuatnya mudah untuk membuat website dengan Ruby.

AWS menyediakan `aws-sdk-rails` permata untuk memungkinkan integrasi yang mudah dengan Rails. Anda dapat menggunakan AWS Elastic Beanstalk, AWS OpsWorks, AWS CodeDeploy, atau [Penyedia AWS Rails untuk](#) menerapkan dan menjalankan aplikasi Rails Anda di AWS Cloud.

[Untuk informasi tentang menginstal dan menggunakan `aws-sdk-rails` permata, lihat GitHub repositori <https://github.com/aws/.aws-sdk-rails>](#)

### Tip debugging: Dapatkan informasi pelacakan kawat dari klien

Anda bisa mendapatkan informasi wire trace dari AWS klien dengan menetapkan `http_wire_trace` Boolean. Informasi pelacakan kawat membantu membedakan perubahan klien, masalah layanan, dan kesalahan pengguna. `Kapan true`, pengaturan menunjukkan apa yang dikirim pada kawat. Contoh berikut membuat klien Amazon S3 dengan pelacakan kawat diaktifkan pada saat pembuatan klien.

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

Mengingat kode berikut dan argumen `bucket_name`, output menampilkan pesan yang mengatakan apakah bucket dengan nama yang ada.

```
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
```

Jika ember ada, output serupa dengan yang berikut ini. (Pengembalian ditambahkan ke HEAD baris untuk keterbacaan.)

```
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: 5MD4APQOS815QVBR\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

```

Anda juga dapat mengaktifkan pelacakan kawat setelah pembuatan klien.

```

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

```

Untuk informasi selengkapnya tentang bidang dalam informasi pelacakan kawat yang dilaporkan, lihat [Header permintaan wajib Transfer Family](#).

## Respons dan kesalahan klien rintisan

Pelajari cara menghentikan respons klien dan kesalahan klien dalam aplikasi AWS SDK for Ruby.

### Tanggapan klien yang mematahkan

Ketika Anda menghentikan respons, AWS SDK for Ruby menonaktifkan lalu lintas jaringan dan klien mengembalikan data stubbed (atau palsu). Jika Anda tidak menyediakan data stubbed, klien akan mengembalikan:

- Daftar sebagai array kosong

- Maps sebagai hash kosong
- Nilai numerik sebagai nol
- Tanggal sebagai now

Contoh berikut menampilkan nama stubbed untuk daftar bucket Amazon S3.

```
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
```

Menjalankan kode ini menampilkan berikut ini.

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

#### Note

Setelah Anda menyediakan data stubbed, nilai default tidak lagi berlaku untuk atribut instans yang tersisa. Ini berarti bahwa dalam contoh sebelumnya, atribut instance yang tersisa `creation_date`, tidak `now` tapinil.

AWSSDK for Ruby memvalidasi data stubbed Anda. Jika Anda meneruskan data dari jenis yang salah, itu menimbulkan `ArgumentError` pengecualian. Misalnya, jika alih-alih tugas sebelumnya `bucket_data`, Anda menggunakan yang berikut:

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

AWSSDK for Ruby memunculkan dua pengecualian. `ArgumentError`

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

## Kesalahan klien yang mematahkan

Anda juga dapat melakukan rintisan error yang diajukan AWS SDK for Ruby untuk metode tertentu. Contoh berikut menampilkan `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
```

## Paginasi

Beberapa AWS panggilan memberikan respons paginated untuk membatasi jumlah data yang dikembalikan dengan setiap respons. Halaman data mewakili hingga 1.000 item.

## Respons paginated dapat dihitung

Cara paling sederhana untuk menangani data respons paginated adalah dengan menggunakan pencacah built-in dalam objek respons, seperti yang ditunjukkan pada contoh berikut.

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

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

Ini menghasilkan satu objek respons per panggilan API yang dibuat, dan menghitung objek dalam bucket bernama. SDK mengambil halaman data tambahan untuk menyelesaikan permintaan.

## Menangani tanggapan paged secara manual

Untuk menangani paging sendiri, gunakan `next_page?` metode respons untuk memverifikasi ada lebih banyak halaman untuk diambil, atau gunakan `last_page?` metode untuk memverifikasi bahwa tidak ada lagi halaman yang akan diambil.

Jika ada lebih banyak halaman, gunakan metode `next_page` (pemberitahuan tidak ada?) untuk mengambil halaman hasil berikutnya, seperti yang ditunjukkan pada contoh berikut.

```
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

Jika Anda memanggil `next_page` metode dan tidak ada lagi halaman yang akan diambil, SDK akan memunculkan pengecualian [Aws::PageableResponse::LastPageError](#)

## Kelas data paged

Data paged dalam AWS SDK for Ruby ditangani oleh `PageableResponse` kelas [Aws::](#), yang [disertakan dengan Seahorse](#): [:Client](#): [:Response](#) untuk menyediakan akses ke data paged.

## Pelayan

Pelayan adalah metode utilitas yang polling untuk negara tertentu terjadi pada klien. Pelayan dapat gagal setelah sejumlah upaya pada interval pemungutan suara yang ditentukan untuk klien layanan. Untuk contoh bagaimana pelayan digunakan, lihat metode [create\\_table](#) dari Klien Enkripsi Amazon DynamoDB di Repositori Contoh Kode. AWS



## Memanggil pelayan

Untuk memanggil pelayan, hubungi klien `wait_until` layanan. Pada contoh berikut, pelayan menunggu sampai instance `i-12345678` berjalan sebelum melanjutkan.

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

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

Parameter pertama adalah nama pelayan, yang khusus untuk klien layanan dan menunjukkan operasi mana yang sedang menunggu. Parameter kedua adalah hash parameter yang diteruskan ke metode klien yang disebut oleh pelayan, yang bervariasi sesuai dengan nama pelayan.

Untuk daftar operasi yang dapat ditunggu dan metode klien dipanggil untuk setiap operasi, lihat dokumentasi `waiter_names` dan `wait_until` bidang untuk klien yang Anda gunakan.

## Kegagalan tunggu kegagalan

Pelayan dapat gagal dengan salah satu pengecualian berikut.

### [Aws::Writers::Errors::FailureStateError](#)

Keadaan kegagalan ditemui saat menunggu.

### [Aws::Writers::Errors::NoSuchWaiterError](#)

Nama pelayan yang ditentukan tidak didefinisikan untuk klien yang digunakan.

### [Aws::Writers::Errors::TooManyAttemptsError](#)

Jumlah upaya melebihi nilai pelayan. `max_attempts`

### [Aws::Writers::Errors::UnexpectedError](#)

Terjadi galat tak terduga saat menunggu.

### [Aws::Writers::Errors::WaiterFailed](#)

Salah satu negara menunggu terlampaui atau kegagalan lain terjadi saat menunggu.

Semua kesalahan ini — kecuali - didasarkan `NoSuchWaiterError` pada `WaiterFailed` Untuk catch kesalahan dalam pelayan, gunakan `WaiterFailed`, seperti yang ditunjukkan dalam contoh berikut.

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

## Mengkonfigurasi pelayan

Setiap pelayan memiliki interval polling default dan jumlah upaya maksimum yang akan dilakukan sebelum mengembalikan kontrol ke program Anda. Untuk mengatur nilai-nilai ini, gunakan `max_attempts` dan `delay:` parameter dalam `wait_until` panggilan Anda. Contoh berikut menunggu hingga 25 detik, polling setiap lima detik.

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

Untuk menonaktifkan kegagalan tunggu, tetapkan nilai salah satu parameter ini `nil`.

## Memperluas pelayan

Untuk mengubah perilaku pelayan, Anda dapat mendaftarkan callback yang dipicu sebelum setiap upaya polling dan sebelum menunggu.

Contoh berikut mengimplementasikan cadangan eksponensial dalam pelayan dengan menggandakan jumlah waktu untuk menunggu setiap upaya.

```
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
```

Contoh berikut menonaktifkan jumlah maksimum upaya, dan sebaliknya menunggu selama satu jam (3600 detik) sebelum gagal.

```
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
```

## Tentukan perilaku coba lagi klien

Secara default, AWS SDK for Ruby melakukan hingga tiga percobaan ulang, dengan 15 detik antara percobaan ulang, dengan total hingga empat upaya. Oleh karena itu, operasi bisa memakan waktu hingga 60 detik untuk time out.

Contoh berikut membuat klien Amazon S3 di wilayah tersebutus-west-2, dan menetapkan untuk menunggu lima detik antara dua percobaan ulang pada setiap operasi klien. Oleh karena itu, operasi klien Amazon S3 dapat memakan waktu hingga 15 detik untuk waktu habis.

```
s3 = Aws::S3::Client.new(
  region: region,
  retry_limit: 2,
  retry_backoff: lambda { |c| sleep(5) }
)
```

Contoh ini menunjukkan cara mengubah parameter coba lagi secara langsung di dalam kode. Namun, Anda juga dapat menggunakan variabel lingkungan atau AWS config file bersama untuk mengatur ini untuk aplikasi Anda. Untuk informasi selengkapnya tentang setelan ini, lihat [Coba lagi perilaku](#) di Panduan Referensi AWS SDK dan Alat. Pengaturan eksplisit apa pun yang disetel dalam kode atau pada klien layanan itu sendiri lebih diutamakan daripada yang ditetapkan dalam variabel lingkungan atau file bersama. config

# Bermigrasi dari versi 1 atau 2 ke AWS SDK versi 3 untuk Ruby

Tujuan dari topik ini adalah untuk membantu Anda bermigrasi dari AWS SDK versi 1 atau 2 untuk Ruby ke versi 3.

## Side-by-side Penggunaan

Tidak perlu mengganti AWS SDK untuk Ruby versi 1 atau 2 dengan versi 3. Anda dapat menggunakannya bersama-sama dalam aplikasi yang sama. Lihat [posting blog ini](#) untuk informasi lebih lanjut.

Contoh cepat berikut.

```
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
```

Anda tidak perlu menulis ulang kode versi 1 atau 2 yang sudah ada untuk mulai menggunakan SDK versi 3. Strategi migrasi yang valid adalah hanya menulis kode baru terhadap SDK versi 3.

## Perbedaan umum

Versi 3 berbeda dari versi 2 dengan satu cara penting.

- Setiap layanan tersedia sebagai permata terpisah.

Versi 2 berbeda dari versi 1 dalam beberapa cara penting.

- Namespace root yang berbeda - `Aws` versus `AWS`. Hal ini memungkinkan side-by-side penggunaan.
- `Aws.config`- Sekarang vanilla Ruby hash, bukan metode.
- Opsi konstruktor yang ketat - Saat membangun objek klien atau sumber daya dalam SDK versi 1, opsi konstruktor yang tidak dikenal diabaikan. Dalam versi 2, pilihan konstruktor yang tidak diketahui memicu `ArgumentError`. Misalnya:

```
# 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
```

## Perbedaan klien

Tidak ada perbedaan antara kelas klien di versi 2 dan versi 3.

Antara versi 1 dan versi 2, kelas klien memiliki perbedaan eksternal paling sedikit. Banyak klien layanan akan memiliki antarmuka yang kompatibel setelah konstruksi klien. Beberapa perbedaan penting:

- `Aws::S3::Client`- Versi 1 kelas klien Amazon S3 adalah tangan-kode. Versi 2 dihasilkan dari model layanan. Nama metode dan input sangat berbeda dalam versi 2.
- `Aws::EC2::Client`- Versi 2 menggunakan nama jamak untuk daftar output, versi 1 menggunakan `akhiran_set`. Misalnya:

```
# 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`- Versi 2 menggunakan respons terstruktur, di mana versi 1 menggunakan vanilla Ruby hash.
- Layanan kelas mengubah nama - Versi 2 menggunakan nama yang berbeda untuk beberapa layanan:
  - `AWS::SimpleWorkflow` telah menjadi `Aws::SWF`
  - `AWS::ELB` telah menjadi `Aws::ElasticLoadBalancing`
  - `AWS::SimpleEmailService` telah menjadi `Aws::SES`
- Opsi konfigurasi klien - Beberapa opsi konfigurasi versi 1 diganti namanya dalam versi 2. Lainnya dihapus atau diganti. Berikut adalah perubahan yang mungkin:

- `:use_ssl` telah dihapus. Versi 2 menggunakan SSL di mana-mana. Untuk menonaktifkan SSL Anda harus mengkonfigurasi `:endpoint` yang menggunakan `http://`.
- `:ssl_ca_file` sekarang `:ssl_ca_bundle`
- `:ssl_ca_path` sekarang `:ssl_ca_directory`
- Ditambahkan `:ssl_ca_store`.
- `:endpoint` sekarang harus menjadi HTTP atau HTTPS URI yang memenuhi syarat, bukan nama host.
- Dihapus `:_port` pilihan untuk setiap layanan, sekarang digantikan oleh `:endpoint`.
- `:user_agent_prefix` sekarang `:user_agent_suffix`

## Perbedaan sumber daya

Tidak ada perbedaan antara antarmuka sumber daya di versi 2 dan versi 3.

Ada perbedaan yang signifikan antara antarmuka sumber daya dalam versi 1 dan versi 2. Versi 1 sepenuhnya tangan-kode, di mana sebagai versi 2 antarmuka sumber daya yang dihasilkan dari model. Antarmuka sumber daya versi 2 secara signifikan lebih konsisten. Beberapa perbedaan sistemik meliputi:

- Kelas sumber daya terpisah - Dalam versi 2, nama layanan adalah modul, bukan kelas. Dalam modul ini, itu adalah antarmuka sumber daya:

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

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

- Sumber daya referensi - SDK versi 2 memisahkan koleksi dan pengambil sumber daya individual menjadi dua metode berbeda:

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

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

- Operasi Batch - Dalam versi 1, semua operasi batch adalah utilitas dengan kode tangan. Dalam versi 2, banyak operasi batch yang autogenerated operasi batching atas API. Antarmuka batching versi 2 sangat berbeda dari versi 1.

# Bekerja dengan Layanan AWS

Bagian berikut berisi diskusi dan contoh yang menunjukkan cara menggunakan AWS SDK for Ruby untuk bekerja dengannya. Layanan AWS

Jika Anda baru mengenal AWS SDK for Ruby, Anda mungkin ingin [Memulai](#) membaca topiknya terlebih dahulu.

- [Contoh kode dengan panduan](#)— Memberikan contoh terpandu untuk beberapa Layanan AWS.
- [Contoh kode](#)— Menyediakan daftar lengkap contoh layanan yang tersedia (tetapi tanpa panduan tambahan di luar kode).

Kode sumber untuk semua contoh ini tersedia untuk diunduh di [Repositori Contoh AWS Kode](#) di GitHub. Untuk mengusulkan contoh kode baru agar tim AWS dokumentasi mempertimbangkan untuk memproduksi, buat permintaan baru. Tim ingin menghasilkan contoh kode yang mencakup skenario dan kasus penggunaan yang lebih luas, dibandingkan cuplikan kode sederhana yang hanya mencakup panggilan API individual. Untuk petunjuk, lihat bagian Mengusulkan contoh kode baru di [Readme](#) on GitHub.

## Contoh kode dengan panduan untuk AWS SDK for Ruby

Bagian ini memberikan contoh yang dapat Anda gunakan untuk mengakses Layanan AWS dengan menggunakan AWS SDK for Ruby.

Temukan kode sumber untuk contoh-contoh ini dan lainnya di [Repositori Contoh AWS Kode](#) di GitHub.

### Topik

- [CloudTrail Contoh Menggunakan AWS SDK for Ruby](#)
- [CloudWatch Contoh Amazon Menggunakan AWS SDK for Ruby](#)
- [CodeBuild Contoh Menggunakan AWS SDK for Ruby](#)
- [Contoh Amazon EC2 Menggunakan AWS SDK for Ruby](#)
- [AWS Elastic Beanstalk Contoh Menggunakan AWS SDK for Ruby](#)
- [AWS Identity and Access Management\(IAM\) Contoh Menggunakan AWS SDK for Ruby](#)
- [AWS Key Management Service Contoh Menggunakan AWS SDK for Ruby](#)



- [AWS Lambda Contoh Menggunakan AWS SDK for Ruby](#)
- [Contoh Amazon Polly Menggunakan AWS SDK for Ruby](#)
- [Contoh Amazon RDS Menggunakan AWS SDK for Ruby](#)
- [Contoh Amazon SES Menggunakan AWS SDK for Ruby](#)
- [Contoh Amazon SNS Menggunakan AWS SDK for Ruby](#)
- [Contoh Amazon SQS Menggunakan AWS SDK for Ruby](#)
- [WorkDocs Contoh Amazon](#)

## CloudTrail Contoh Menggunakan AWS SDK for Ruby

CloudTrail adalah Layanan AWS yang dapat Anda gunakan untuk memantau AWS penerapan Anda di cloud dengan mendapatkan riwayat panggilan AWS API untuk akun Anda. Anda dapat menggunakan contoh kode AWS SDK for Ruby berikut untuk mengakses AWS CloudTrail Untuk informasi selengkapnya CloudTrail, lihat [dokumentasi AWS CloudTrail](#).

### Topik

- [Daftar CloudTrail Jejak](#)
- [Membuat CloudTrail Jejak](#)
- [Daftar Acara CloudTrail Trail](#)
- [Menghapus Jejak CloudTrail](#)

### Daftar CloudTrail Jejak

Contoh ini menggunakan metode [describe\\_trails](#) untuk mencantumkan nama CloudTrail jejak dan bucket tempat CloudTrail menyimpan informasi di wilayah tersebut. `us-west-2`

Pilih Copy untuk menyimpan kode secara lokal.

Buat file `describe_trails.rb` dengan kode berikut.

```
# 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
```

Lihat [contoh lengkapnya](#) di GitHub.

## Membuat CloudTrail Jejak

Contoh ini menggunakan metode [create\\_trail](#) untuk membuat jejak di wilayah CloudTrail tersebut. `us-west-2` ini membutuhkan dua input, nama jejak dan nama ember tempat CloudTrail menyimpan informasi. Jika bucket tidak memiliki kebijakan yang tepat, sertakan flag `-p` untuk melampirkan kebijakan yang benar ke bucket.

Pilih Copy untuk menyimpan kode secara lokal.

```
# 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',
            },
          },
        },
      ],
    },
  end
end
]
```

```
}.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
  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
  })

  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
```

Lihat [contoh lengkapnya](#) di GitHub.

## Daftar Acara CloudTrail Trail

Contoh ini menggunakan metode [lookup\\_events](#) untuk membuat daftar peristiwa CloudTrail jejak di wilayah tersebut. us-west-2

Pilih Copy untuk menyimpan kode secara lokal.

```
# 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

# 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
```

Lihat [contoh lengkapnya](#) di GitHub.

## Menghapus Jejak CloudTrail

Contoh ini menggunakan metode [delete\\_trail](#) untuk menghapus jejak di wilayah CloudTrail tersebut. `us-west-2` ini membutuhkan satu masukan, nama jejak.

Pilih Copy untuk menyimpan kode secara lokal.

```
# 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
```

Lihat [contoh lengkapnya](#) di GitHub.

## CloudWatch Contoh Amazon Menggunakan AWS SDK for Ruby

Amazon CloudWatch (CloudWatch) adalah layanan pemantauan untuk sumber daya AWS cloud dan aplikasi yang Anda jalankan AWS. Anda dapat menggunakan contoh berikut untuk mengakses CloudWatch dengan menggunakan AWS SDK for Ruby. Untuk informasi selengkapnya CloudWatch, lihat [CloudWatch dokumentasi Amazon](#).

### Topik

- [Mendapatkan Informasi tentang Amazon CloudWatch Alarm](#)
- [Membuat CloudWatch Alarm Amazon](#)

- [Mengaktifkan dan Menonaktifkan Tindakan Alarm Amazon CloudWatch](#)
- [Mendapatkan Informasi tentang Metrik Kustom untuk Amazon CloudWatch](#)
- [Mengirim Acara ke CloudWatch Acara Amazon](#)

## Mendapatkan Informasi tentang Amazon CloudWatch Alarm

Contoh kode berikut menampilkan informasi tentang alarm metrik yang tersedia di Amazon 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
    end
  end
end
```



```
    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

# 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__

```

## Membuat CloudWatch Alarm Amazon

Contoh kode berikut membuat CloudWatch alarm baru (atau memperbarui alarm yang ada, jika alarm dengan nama yang ditentukan sudah ada).

```

# 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}"
  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

run_me if $PROGRAM_NAME == __FILE__
```

## Mengaktifkan dan Menonaktifkan Tindakan Alarm Amazon CloudWatch

Contoh kode berikut:

1. Membuat dan mengaktifkan CloudWatch alarm baru (atau memperbarui alarm yang ada, jika alarm dengan nama yang ditentukan sudah ada).
2. Menonaktifkan alarm baru atau yang sudah ada. Untuk mengaktifkan alarm lagi, `hubbungienable_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 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

# 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

run_me if $PROGRAM_NAME == __FILE__
```

## Mendapatkan Informasi tentang Metrik Kustom untuk Amazon CloudWatch

Contoh kode berikut:

1. Menambahkan titik data ke metrik khusus di. CloudWatch
2. Menampilkan daftar metrik yang tersedia untuk namespace metrik di. 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
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
      end
    end
  end
end

```

```
        end
      else
        puts 'No dimensions found.'
      end
    end
  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__
```

## Mengirim Acara ke CloudWatch Acara Amazon

Contoh kode berikut menunjukkan cara membuat dan memicu aturan di Amazon CloudWatch Events. Aturan ini mengirimkan pemberitahuan ke topik yang ditentukan di Amazon Simple Notification Service (Amazon SNS) Simple Notification Service (Amazon SNS) setiap kali instance yang tersedia di Amazon Elastic Compute Cloud (Amazon EC2) berubah menjadi status berjalan. Selain itu, informasi peristiwa terkait dicatat ke grup log di CloudWatch Acara.

```
# 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
  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

# 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
end

```



```
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
    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'
  )
end
```

```

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
      )
    end
  end
end

```

```

    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}'."
end

```

```
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)
  end
  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 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 Contoh Menggunakan AWS SDK for Ruby

CodeBuild adalah layanan build yang dikelola sepenuhnya yang mengkompilasi kode sumber, menjalankan pengujian, dan menghasilkan paket perangkat lunak yang siap digunakan. Anda dapat



menggunakan contoh kode AWS SDK for Ruby berikut untuk mengakses. AWS CodeBuild Untuk informasi selengkapnya CodeBuild, lihat [AWS CodeBuild dokumentasi](#).

## Topik

- [Mendapatkan Informasi Tentang Semua AWS CodeBuild Proyek](#)
- [Membangun AWS CodeBuild Proyek](#)
- [Daftar AWS CodeBuild Proyek Builds](#)

## Mendapatkan Informasi Tentang Semua AWS CodeBuild Proyek

Contoh berikut mencantumkan nama hingga 100 AWS CodeBuild proyek Anda.

```
# 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
```

Pilih Copy untuk menyimpan kode secara lokal. Lihat [contoh lengkapnya](#) di GitHub.

## Membangun AWS CodeBuild Proyek

Contoh berikut membangun AWS CodeBuild proyek yang ditentukan pada baris perintah. Jika tidak ada argumen baris perintah yang diberikan, itu memancarkan kesalahan dan berhenti.

```
# 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
```

Pilih Copy untuk menyimpan kode secara lokal. Lihat [contoh lengkapnya](#) di GitHub.

## Daftar AWS CodeBuild Proyek Builds

Contoh berikut menampilkan informasi tentang build AWS CodeBuild proyek Anda. Informasi ini mencakup nama proyek, kapan build dimulai, dan berapa lama waktu yang dibutuhkan setiap fase build, dalam hitungan detik.

```
# 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
```

Pilih Copy untuk menyimpan kode secara lokal. Lihat [contoh lengkapnya](#) di GitHub.

## Contoh Amazon EC2 Menggunakan AWS SDK for Ruby

Amazon Elastic Compute Cloud (Amazon EC2) adalah layanan web yang menyediakan kapasitas komputasi yang dapat diubah ukurannya — secara harfiah server di pusat data Amazon — yang Anda gunakan untuk membangun dan meng-host sistem perangkat lunak Anda. Anda dapat menggunakan contoh berikut untuk mengakses Amazon EC2 menggunakan AWS SDK for Ruby. Untuk informasi selengkapnya tentang Amazon EC2, lihat Dokumentasi [Amazon EC2](#).

## Topik

- [Membuat VPC Amazon EC2](#)
- [Membuat Internet Gateway dan Melampirkannya ke VPC di Amazon EC2](#)
- [Membuat Subnet Publik untuk Amazon EC2](#)
- [Membuat Tabel Rute Amazon EC2 dan Mengaitkannya dengan Subnet](#)
- [Menggunakan Alamat IP Elastis di Amazon EC2](#)
- [Membuat Grup Keamanan Amazon EC2](#)
- [Bekerja dengan Grup Keamanan Amazon EC2](#)
- [Bekerja dengan Pasangan Kunci di Amazon EC2](#)
- [Mendapatkan Informasi tentang Semua Instans Amazon EC2](#)
- [Mendapatkan Informasi tentang Semua Instans Amazon EC2 dengan Nilai Tag Tertentu](#)
- [Mendapatkan Informasi tentang Instans Amazon EC2 Tertentu](#)
- [Membuat Instans Amazon EC2](#)
- [Menghentikan Instans Amazon EC2](#)
- [Memulai Instans Amazon EC2](#)
- [Mem-boot ulang Instans Amazon EC2](#)
- [Mengelola Instans Amazon EC2](#)
- [Mengakhiri Instans Amazon EC2](#)
- [Mendapatkan Informasi tentang Wilayah dan Availability Zone untuk Amazon EC2](#)

## Membuat VPC Amazon EC2

Contoh kode berikut membuat virtual private cloud (VPC) di Amazon Virtual Private Cloud (Amazon VPC) dan kemudian menandai VPC.

```
# 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__
```

## Membuat Internet Gateway dan Melampirkannya ke VPC di Amazon EC2

Contoh kode berikut membuat gateway internet dan kemudian melampirkannya ke virtual private cloud (VPC) di Amazon Virtual Private Cloud (Amazon 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
```

```
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

  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__
```

## Membuat Subnet Publik untuk Amazon EC2

Contoh kode berikut membuat subnet dalam virtual private cloud (VPC) di Amazon Virtual Private Cloud (Amazon VPC) dan kemudian menandai subnet.

```
# 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__
```

## Membuat Tabel Rute Amazon EC2 dan Mengaitkannya dengan Subnet

Contoh kode berikut membuat tabel rute di Amazon Virtual Private Cloud (Amazon VPC) dan kemudian mengaitkan tabel rute dengan subnet di Amazon VPC.

```
# 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\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

run_me if $PROGRAM_NAME == __FILE__
```

## Menggunakan Alamat IP Elastis di Amazon EC2

Contoh kode berikut:

1. Menampilkan informasi tentang alamat apa pun yang terkait dengan instans Amazon Elastic Compute Cloud (Amazon EC2).

2. Membuat alamat IP Elastis di Amazon Virtual Private Cloud (Amazon VPC).
3. Mengaitkan alamat dengan instance.
4. Menampilkan informasi lagi tentang alamat yang terkait dengan instance. Kali ini, asosiasi alamat baru harus ditampilkan.
5. Melepaskan alamatnya.
6. Menampilkan informasi lagi tentang alamat yang terkait dengan instance. Kali ini, alamat yang dirilis seharusnya tidak ditampilkan.

```
# 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]
      }
    ]
  )
  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
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__

```

## Membuat Grup Keamanan Amazon EC2

Contoh kode berikut membuat grup keamanan Amazon EC2 dan kemudian menambahkan aturan keluar ke grup keamanan tersebut.

```

# 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
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__
```

## Bekerja dengan Grup Keamanan Amazon EC2

Contoh berikut:

1. Membuat grup keamanan Amazon EC2.
2. Menambahkan aturan masuk ke grup keamanan.
3. Menampilkan informasi tentang grup keamanan yang tersedia.
4. Menghapus grup keamanan.

```
# 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
end

```

```

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

  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
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?
      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
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

# 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
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

run_me if $PROGRAM_NAME == __FILE__
```

## Bekerja dengan Pasangan Kunci di Amazon EC2

Contoh kode berikut:

1. Membuat key pair di Amazon EC2.
2. Menampilkan informasi tentang pasangan kunci yang tersedia.
3. Menghapus 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
end
```

```
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__
```

## Mendapatkan Informasi tentang Semua Instans Amazon EC2

Contoh kode berikut mencantumkan ID dan status saat ini dari instans Amazon EC2 yang tersedia.

```
# 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__
```

## Mendapatkan Informasi tentang Semua Instans Amazon EC2 dengan Nilai Tag Tertentu

Contoh kode berikut mencantumkan ID dan status saat ini dari instans Amazon EC2 yang tersedia yang cocok dengan kunci dan nilai tag yang ditentukan.

```
# 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
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__
```

## Mendapatkan Informasi tentang Instans Amazon EC2 Tertentu

Contoh berikut mencantumkan status instans Amazon EC2 yang ditentukan.

```
# 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__

```

## Membuat Instans Amazon EC2

Contoh berikut membuat dan menandai instans Amazon EC2.

```

# 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
      ]
    )
  end
  # 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__
```

## Menghentikan Instans Amazon EC2

Contoh berikut mencoba menghentikan instans Amazon EC2 yang ditentukan.

```
# 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__
```

## Memulai Instans Amazon EC2

Contoh berikut mencoba untuk memulai instans Amazon EC2 yang ditentukan.

```
# 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__
```

## Mem-boot ulang Instans Amazon EC2

Contoh berikut mencoba untuk me-reboot instans Amazon EC2 yang ditentukan.

```
# 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
end
```

```
end

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

run_me if $PROGRAM_NAME == __FILE__
```

## Mengelola Instans Amazon EC2

Contoh kode berikut:

1. Menghentikan instans Amazon EC2.
2. Memulai ulang instance.
3. Reboot instance.
4. Memungkinkan pemantauan terperinci untuk contoh ini.
5. Menampilkan informasi tentang instance yang tersedia.

```
# 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

# 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}"
        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}"
      end
    end
  end
end

```

```
    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

  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__
```

## Mengakhiri Instans Amazon EC2

Contoh berikut mencoba untuk menghentikan instans Amazon EC2 yang ditentukan.

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

## Mendapatkan Informasi tentang Wilayah dan Availability Zone untuk Amazon EC2

Contoh berikut:

1. Menampilkan daftar Wilayah AWS untuk Amazon EC2 yang tersedia untuk Anda.
2. Menampilkan daftar Zona Ketersediaan Amazon EC2 yang tersedia untuk Anda tergantung pada klien Wilayah AWS Amazon EC2.

```
# 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
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 Contoh Menggunakan AWS SDK for Ruby

AWS Elastic Beanstalk memungkinkan Anda untuk dengan cepat menyebarkan dan mengelola aplikasi di AWS Cloud tanpa khawatir tentang infrastruktur yang menjalankan aplikasi tersebut. Anda dapat menggunakan contoh berikut untuk mengakses Elastic Beanstalk menggunakan AWS SDK

for Ruby. [Untuk informasi lebih lanjut tentang Elastic Beanstalk, lihat dokumentasinya. AWS Elastic Beanstalk](#)

## Topik

- [Mendapatkan Informasi tentang Semua Aplikasi di AWS Elastic Beanstalk](#)
- [Mendapatkan Informasi tentang Aplikasi Tertentu di AWS Elastic Beanstalk](#)
- [Memperbarui Aplikasi Ruby on Rails untuk AWS Elastic Beanstalk](#)

## Mendapatkan Informasi tentang Semua Aplikasi di AWS Elastic Beanstalk

Contoh berikut mencantumkan nama, deskripsi, dan URL semua aplikasi Elastic Beanstalk Anda di wilayah tersebut. 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
```

## Mendapatkan Informasi tentang Aplikasi Tertentu di AWS Elastic Beanstalk

Contoh berikut mencantumkan nama, deskripsi, dan URL MyRailsApp aplikasi di us-west-2 wilayah tersebut.

```
# 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
```

## Memperbarui Aplikasi Ruby on Rails untuk AWS Elastic Beanstalk

Contoh berikut memperbarui aplikasi Ruby on Rails MyRailsApp di wilayah tersebut. us-west-2

### Note

Anda harus berada di root aplikasi Rails Anda untuk berhasil menjalankan skrip.

```
# 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) Contoh Menggunakan AWS SDK for Ruby

AWS Identity and Access Management(IAM) adalah layanan web untuk mengontrol akses dengan aman. Layanan AWS Anda dapat menggunakan contoh berikut untuk mengakses IAM menggunakan AWS SDK for Ruby. Untuk informasi selengkapnya tentang IAM, lihat dokumentasi [IAM](#).

### Topik

- [Mendapatkan Informasi tentang Pengguna IAM](#)
- [Daftar Pengguna IAM yang merupakan Administrator](#)
- [Menambahkan Pengguna IAM Baru](#)
- [Membuat Kunci Akses Pengguna untuk Pengguna IAM](#)
- [Menambahkan Kebijakan Terkelola ke Pengguna IAM](#)
- [Membuat Peran IAM](#)
- [Mengelola Pengguna IAM](#)
- [Bekerja dengan Kebijakan IAM](#)
- [Mengelola Kunci Akses IAM](#)
- [Bekerja dengan Sertifikat Server IAM](#)
- [Mengelola Alias Akun IAM](#)

## Mendapatkan Informasi tentang Pengguna IAM

Contoh berikut mencantumkan grup, kebijakan, dan ID kunci akses pengguna IAM di us-west-2 wilayah tersebut. Jika ada lebih dari 100 pengguna, `iam.list_users.IsTruncated` benar dan `iam.list_users.Marker` berisi nilai yang dapat Anda gunakan untuk mendapatkan informasi tentang pengguna tambahan. Lihat topik [Aws: :IAM: :client.list\\_users](#) untuk informasi lebih lanjut.

```
# 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
```

```
        end
      else
        puts ' None'
      end
    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__
```

## Daftar Pengguna IAM yang merupakan Administrator

Contoh berikut menggunakan [get\\_account\\_authorization\\_details](#), metode untuk mendapatkan daftar pengguna untuk akun saat ini.

Pilih Copy untuk menyimpan kode secara lokal.

Buat file `get_admins.rb`.

Tambahkan permata IAM yang diperlukan dan permata os, dan gunakan yang terakhir untuk menggunakan sertifikat yang dibundel jika Anda menjalankan Microsoft Windows.

### Note

Versi 2 AWS SDK for Ruby tidak memiliki permata khusus layanan.

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

if OS.windows?
```



```
Aws.use_bundled_cert!  
end
```

Buat metode untuk menentukan apakah pengguna memiliki kebijakan dengan hak administrator.

```
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
```

Buat metode untuk menentukan apakah pengguna memiliki kebijakan terlampir dengan hak administrator.

```
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
```

Buat metode untuk menentukan apakah grup yang dimiliki pengguna memiliki kebijakan dengan hak administrator.

Buat metode untuk menentukan apakah grup yang dimiliki pengguna memiliki kebijakan terlampir dengan hak administrator.

```
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

  false
end
```

Buat metode untuk menentukan apakah grup yang dimiliki pengguna memiliki hak administrator.

```
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
```

Buat metode untuk menentukan apakah pengguna memiliki hak administrator.

```
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

false
end
```

Buat metode untuk mengulang daftar pengguna dan mengembalikan berapa banyak dari pengguna tersebut yang memiliki hak administrator.

```
<code>
```

Rutinitas utama dimulai di sini. Buat klien IAM dan variabel untuk menyimpan jumlah pengguna, jumlah pengguna yang memiliki hak istimewa administrator, dan string yang mengidentifikasi kebijakan yang menyediakan hak istimewa administrator.

```
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
```

Hubungi `get_account_authorization_details` untuk mendapatkan detail akun dan dapatkan pengguna untuk akun tersebut `user_detail_list`. Melacak berapa banyak pengguna yang kami dapatkan, hubungi `get_admin_count` untuk mendapatkan jumlah pengguna yang memiliki hak administrator, dan melacak jumlah tersebut.

```
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
```

Jika panggilan pertama `get_account_authorization_details` tidak mendapatkan semua detail, hubungi lagi dan ulangi proses menentukan berapa banyak yang memiliki hak administrator.

```
<code>
```

Terakhir, tampilkan berapa banyak pengguna yang memiliki hak administrator.

```
more_users = details.is_terpotong
```

sementara `more_users`

```
  rincian = client.get_account_authorization_details (
    filter: ['Pengguna'], penanda: details.marker
  )
```

```
  pengguna = details.user_detail_list
```

```
  num_users += users.count
  more_admins = get_admin_count (klien, pengguna, access_admin)
  num_admin += more_admin
```

```
  more_users = details.is_terpotong
```

```
end
```

Lihat [contoh lengkapnya](#) di GitHub.

## Menambahkan Pengguna IAM Baru

Contoh berikut membuat pengguna IAM `my_groovy_user` di `us-west-2` wilayah dengan kata sandi `REPLACE_ME`, dan menampilkan ID akun pengguna. Jika pengguna dengan nama itu sudah ada, itu akan menampilkan pesan dan tidak membuat pengguna baru.

```
# 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__

```

## Membuat Kunci Akses Pengguna untuk Pengguna IAM

Contoh berikut membuat kunci akses dan kunci rahasia untuk pengguna IAM `my_groovy_user` di `us-west-2` wilayah tersebut.

```

# 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__
```

## Menambahkan Kebijakan Terkelola ke Pengguna IAM

Contoh berikut menambahkan kebijakan terkelola AmazonS3FullAccess ke pengguna IAM my\_groovy\_user di us-west-2 wilayah tersebut.

```
# 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
end

run_me if $PROGRAM_NAME == __FILE__

```

## Membuat Peran IAM

Contoh berikut membuat peran `my_groovy_role` sehingga Amazon EC2 dapat mengakses Amazon S3 dan Amazon DynamoDB di wilayah tersebut. `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__
```

## Mengelola Pengguna IAM

Pengguna IAM mewakili orang atau layanan yang berinteraksi dengannya. AWS Untuk informasi selengkapnya tentang pengguna IAM, lihat Pengguna [IAM](#).

Dalam contoh ini, Anda menggunakan AWS SDK for Ruby dengan IAM untuk:

1. Dapatkan informasi tentang pengguna AWS IAM yang tersedia dengan menggunakan [Aws: :IAM: :Client #list\\_users](#).
2. Buat pengguna dengan menggunakan [Aws: :IAM: :Client #create\\_user](#).
3. Perbarui nama pengguna dengan menggunakan [Aws: :IAM: :Client #update\\_user](#).
4. Hapus pengguna dengan menggunakan [Aws: :IAM: :Client #delete\\_user](#).

### Prasyarat

Sebelum menjalankan kode contoh, Anda perlu menginstal dan mengkonfigurasi AWS SDK for Ruby, seperti yang dijelaskan dalam:

- [Memasang AWS SDK for Ruby](#)
- [Mengkonfigurasi AWS SDK for Ruby](#)

### Contoh

```
# 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
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}'..."
end
```

```
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__
```

## Bekerja dengan Kebijakan IAM

Kebijakan IAM adalah dokumen yang menentukan satu atau beberapa izin. Untuk informasi selengkapnya tentang kebijakan IAM, lihat [Ikhtisar Kebijakan IAM](#).

Dalam contoh ini, Anda menggunakan AWS SDK for Ruby dengan IAM untuk:

1. Buat kebijakan, menggunakan [Aws: :IAM: :Client #create\\_policy](#).
2. Dapatkan informasi tentang kebijakan, menggunakan [Aws: :IAM: :Client #get\\_policy](#).
3. Lampirkan kebijakan ke peran, menggunakan [Aws: :IAM: :Client #attach\\_role\\_policy](#).
4. Buat daftar kebijakan yang dilampirkan pada peran, menggunakan [Aws: :IAM: :Client #list\\_attached\\_role\\_policies](#).
5. Lepaskan kebijakan dari peran, menggunakan [Aws: :IAM: :Client #detach\\_role\\_policy](#).

### Prasyarat

Sebelum menjalankan kode contoh, Anda perlu menginstal dan mengkonfigurasi AWS SDK for Ruby, seperti yang dijelaskan dalam:

- [Memasang AWS SDK for Ruby](#)
- [Mengkonfigurasi AWS SDK for Ruby](#)

Anda juga perlu membuat peran (peran saya) yang ditentukan dalam skrip. Anda dapat melakukan ini di konsol IAM.

### Contoh

```
# 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
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)
  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__
```

## Mengelola Kunci Akses IAM

Pengguna memerlukan kunci akses mereka sendiri untuk melakukan panggilan terprogram AWS dari AWS SDK for Ruby. Untuk memenuhi kebutuhan ini, Anda dapat membuat, memodifikasi, melihat, atau memutar kunci akses (ID kunci akses dan kunci akses rahasia) untuk pengguna IAM. Secara default, saat Anda membuat kunci akses, statusnya adalah Aktif. Ini berarti pengguna dapat menggunakan kunci akses untuk panggilan API. Untuk informasi selengkapnya tentang kunci akses, lihat [Mengelola Kunci Akses untuk Pengguna IAM](#).

Dalam contoh ini, Anda menggunakan AWS SDK for Ruby dengan IAM untuk:

1. Buat daftar kunci akses pengguna AWS IAM, menggunakan [Aws: :IAM: :Client #list\\_access\\_keys](#).
2. Buat kunci akses, menggunakan [Aws: :IAM: :Client #create\\_access\\_key](#).
3. Tentukan kapan kunci akses terakhir digunakan, menggunakan [Aws: :IAM: :Client #get\\_access\\_key\\_last\\_used](#).
4. Nonaktifkan kunci akses, menggunakan [Aws: :IAM: :Client #update\\_access\\_key](#).
5. Hapus kunci akses, menggunakan [Aws: :IAM: :Client #delete\\_access\\_key](#).

### Prasyarat

Sebelum menjalankan kode contoh, Anda perlu menginstal dan mengkonfigurasi AWS SDK for Ruby, seperti yang dijelaskan dalam:

- [Memasang AWS SDK for Ruby](#)
- [Mengkonfigurasi AWS SDK for Ruby](#)

Anda juga perlu membuat pengguna (my-user) yang ditentukan dalam skrip. Anda dapat membuat pengguna IAM baru di konsol IAM atau secara terprogram, seperti yang ditunjukkan pada [Menambahkan Pengguna IAM Baru](#).

### Contoh

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

```

```
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)
end
```



```
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
end

run_me if $PROGRAM_NAME == __FILE__
```

## Bekerja dengan Sertifikat Server IAM

Untuk mengaktifkan koneksi HTTPS ke situs web atau aplikasi Anda AWS, Anda memerlukan sertifikat server SSL/TLS. Untuk menggunakan sertifikat yang Anda peroleh dari penyedia eksternal dengan situs web atau aplikasi Anda AWS, Anda harus mengunggah sertifikat ke IAM atau mengimpornya ke AWS Certificate Manager. Untuk informasi selengkapnya tentang sertifikat server, lihat [Bekerja dengan Sertifikat Server](#).

Dalam contoh ini, Anda menggunakan AWS SDK for Ruby dengan IAM untuk:

1. Perbarui sertifikat server, menggunakan [Aws: :IAM: :Client #update\\_server\\_certificate](#).
2. Hapus sertifikat server, menggunakan [Aws: :IAM: :Client #delete\\_server\\_certificate](#).
3. Daftar informasi tentang sertifikat server yang tersisa, menggunakan [Aws: :IAM: :Client #list\\_server\\_certificates](#).

## Prasyarat

Sebelum menjalankan kode contoh, Anda perlu menginstal dan mengkonfigurasi AWS SDK for Ruby, seperti yang dijelaskan dalam:

- [Memasang AWS SDK for Ruby](#)
- [Mengkonfigurasi AWS SDK for Ruby](#)

### Note

Sertifikat server harus sudah ada, atau skrip akan menampilkan kesalahan  
Aws: :IAM: :Errors:: NoSuchEntity

## Contoh

```
# 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
  end
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

    puts "Server certificate names now are:\n\n"
    list_server_certificate_names(iam_client)
  end
end
end
end

run_me if $PROGRAM_NAME == __FILE__
```

## Mengelola Alias Akun IAM

Jika Anda ingin URL untuk halaman masuk berisi nama perusahaan atau pengenal ramah lainnya, bukan ID AWS akun Anda, Anda dapat membuat alias akun IAM untuk ID akun Anda. AWS Jika Anda membuat alias akun IAM, URL halaman masuk Anda berubah untuk memasukkan alias. Untuk informasi selengkapnya tentang alias akun IAM, lihat [ID AWS Akun Anda dan Aliasnya](#).

Dalam contoh ini, Anda menggunakan AWS SDK for Ruby dengan IAM untuk:

1. Buat daftar alias AWS akun, menggunakan [Aws::IAM::Client#list\\_account\\_aliases](#).
2. Buat alias akun, menggunakan [Aws::IAM::Client#create\\_account\\_alias](#).
3. Hapus alias akun, menggunakan [Aws::IAM::Client#delete\\_account\\_alias](#).

### Prasyarat

Sebelum menjalankan kode contoh, Anda perlu menginstal dan mengkonfigurasi AWS SDK for Ruby, seperti yang dijelaskan dalam:

- [Memasang AWS SDK for Ruby](#)
- [Mengkonfigurasi AWS SDK for Ruby](#)

Dalam kode contoh, ubah my-account-aliasstring menjadi sesuatu yang unik di semua produk Amazon Web Services.

### Contoh

```
# 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
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
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 Contoh Menggunakan AWS SDK for Ruby

AWS Key Management Service(AWS KMS) adalah enkripsi dan layanan manajemen kunci yang diskalakan untuk cloud. Anda dapat menggunakan contoh berikut untuk mengakses AWS KMS menggunakan AWS SDK for Ruby. Untuk informasi selengkapnya AWS KMS, lihat [AWS KMS dokumentasi](#). Untuk informasi referensi tentang AWS KMS klien, lihat [Aws: :KMS: :Client](#).



## Topik

- [Membuat AWS KMS key](#)
- [Mengkripsi Data di AWS KMS](#)
- [Mendekripsi Gumpalan Data di AWS KMS](#)
- [Mengkripsi ulang Gumpalan Data di AWS KMS](#)

## Membuat AWS KMS key

Contoh berikut menggunakan metode AWS SDK [for](#) Ruby `create_key`, yang [CreateKey](#) mengimplementasikan operasi untuk membuat AWS KMS keys. Karena contoh hanya mengenkripsi sejumlah kecil data, kunci KMS baik-baik saja untuk tujuan kita. Untuk jumlah data yang lebih besar, gunakan kunci KMS untuk mengenkripsi kunci enkripsi data (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
```

Lihat [contoh lengkapnya](#) di GitHub.

## Mengenkripsi Data di AWS KMS

Contoh berikut menggunakan metode enkripsi AWS SDK [for Ruby](#), yang [mengimplementasikan operasi Enkripsi, untuk mengenkripsi string](#) "1234567890". Contoh ini menampilkan versi yang dapat dibaca dari gumpalan terenkripsi yang dihasilkan.

```
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*")
```

Lihat [contoh lengkapnya](#) di GitHub.

## Mendekripsi Gumpalan Data di AWS KMS

Contoh berikut menggunakan metode dekripsi AWS SDK [for Ruby](#), yang [mengimplementasikan operasi Dekripsi](#), untuk mendekripsi string yang disediakan dan memancarkan hasilnya.

```
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
```

Lihat [contoh lengkapnya](#) di GitHub.

## Mengenkripsi ulang Gumpalan Data di AWS KMS

Contoh berikut menggunakan metode AWS SDK [for Ruby](#) `re_encrypt`, yang [ReEncrypt](#) mengimplementasikan operasi, untuk mendekripsi data terenkripsi dan kemudian segera mengenkripsi ulang data di bawah yang baru. AWS KMS key Operasi dilakukan sepenuhnya di sisi server dalam AWS KMS, jadi mereka tidak pernah mengekspos plaintext Anda di luar AWS KMS. Contoh ini menampilkan versi yang dapat dibaca dari gumpalan terenkripsi ulang yang dihasilkan.

```
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*")
```

Lihat [contoh lengkapnya](#) di GitHub.

## AWS Lambda Contoh Menggunakan AWS SDK for Ruby

AWS Lambda(Lambda) adalah platform komputasi nol administrasi untuk pengembang web backend yang menjalankan kode Anda untuk Anda di AWS Cloud, dan memberi Anda struktur harga berbutir halus. Anda dapat menggunakan contoh berikut untuk mengakses Lambda menggunakan AWS SDK for Ruby. [Untuk informasi selengkapnya tentang Lambda, lihat dokumentasi. AWS Lambda](#)

### Topik

- [Menampilkan Informasi tentang Semua Fungsi Lambda](#)
- [Membuat fungsi Lambda](#)
- [Menjalankan Fungsi Lambda](#)
- [Mengkonfigurasi Fungsi Lambda untuk Menerima Pemberitahuan](#)

### Menampilkan Informasi tentang Semua Fungsi Lambda

Contoh berikut menampilkan nama, ARN, dan peran semua fungsi Lambda Anda di wilayah tersebut. 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-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
```

## Membuat fungsi Lambda

Contoh berikut membuat fungsi Lambda bernama `my-notification-function` di `us-west-2` wilayah menggunakan nilai-nilai ini:

- Peran ARN: `my-resource-arn` Dalam kebanyakan kasus, Anda hanya perlu melampirkan kebijakan `AWSLambdaExecute` terkelola ke kebijakan untuk peran ini.
- Titik masuk fungsi: `my-package.my-class`
- Runtime: `java8`
- Berkas zip: `my-zip-file.zip`
- Ember: `my-notification-bucket`
- Kunci: `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)
```

## Menjalankan Fungsi Lambda

Contoh berikut menjalankan fungsi Lambda bernama `MyGetItemsFunction` di wilayah tersebut. `us-west-2` Fungsi ini mengembalikan daftar item dari database. Masukan JSON terlihat seperti berikut ini.

```
{
  "SortBy": "name|time",
  "SortOrder": "ascending|descending",
  "Number": 50
}
```

di mana:

- `SortBy` adalah kriteria untuk menyortir hasil. Contoh kami menggunakan `time`, yang berarti item yang dikembalikan diurutkan dalam urutan di mana mereka ditambahkan ke database.
- `SortOrder` adalah urutan penyortiran. Contoh kami menggunakan `descending`, yang berarti item terbaru adalah yang terakhir dalam daftar.
- `Number` adalah jumlah maksimum item untuk diambil (defaultnya adalah 50). Contoh kami menggunakan `10`, yang berarti mendapatkan 10 item terbaru.

Output JSON terlihat seperti berikut, di mana:

- `STATUS-CODE` adalah kode status HTTP, `200` berarti panggilan berhasil.
- `RESULT` adalah hasil dari panggilan, salah satu `success` atau `failure`.
- `ERROR` adalah pesan kesalahan jika `result` adalah `failure`, jika tidak, string kosong
- `DATA` adalah array hasil yang dikembalikan jika `result` adalah `success`, jika tidak nihil.

```
{
  "statusCode": "STATUS-CODE",
  "body": {
    "result": "RESULT",
    "error": "ERROR",
    "data": "DATA"
  }
}
```

Langkah pertama adalah memuat modul yang kami gunakan:

- `aws-sdk` memuat modul AWS SDK for Ruby yang kita gunakan untuk menjalankan fungsi Lambda.
- `json` memuat modul JSON yang kami gunakan untuk marshall dan membatalkan muatan permintaan dan respons.
- `os` memuat modul OS yang kami gunakan untuk memastikan kami dapat menjalankan aplikasi Ruby kami di Microsoft Windows. Jika Anda menggunakan sistem operasi yang berbeda, Anda dapat menghapus garis-garis itu.
- Kami kemudian membuat klien Lambda yang kami gunakan untuk menjalankan fungsi Lambda.
- Selanjutnya kita membuat hash untuk argumen permintaan dan panggilan `MyGetItemsFunction`.
- Akhirnya kami mengurai respons, dan jika berhasil, kami mencetak item.

```
# 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
```

Lihat [contoh lengkapnya](#) di GitHub.

## Mengkonfigurasi Fungsi Lambda untuk Menerima Pemberitahuan

Contoh berikut mengonfigurasi fungsi Lambda yang `my-notification-function` dinamai di wilayah `us-west-2` tersebut untuk menerima pemberitahuan dari sumber daya dengan ARN `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)
```

## Contoh Amazon Polly Menggunakan AWS SDK for Ruby

Amazon Polly adalah layanan cloud yang mengubah teks menjadi ucapan yang hidup. Contoh AWS SDK for Ruby dapat mengintegrasikan Amazon Polly ke dalam aplikasi Anda. Pelajari selengkapnya tentang Amazon Polly di dokumentasi [Amazon Polly](#). Contoh mengasumsikan Anda telah menyiapkan dan mengonfigurasi SDK (yaitu, Anda telah mengimpor semua paket yang diperlukan dan menetapkan kredensial dan wilayah Anda). Untuk informasi selengkapnya, lihat [Menginstal AWS SDK for Ruby dan AWS Mengonfigurasi SDK for Ruby](#).

### Topik

- [Mendapatkan Daftar Suara](#)
- [Mendapatkan Daftar Leksikon](#)
- [Mensintesis Pidato](#)


### Mendapatkan Daftar Suara

Contoh ini menggunakan metode [describe\\_voices](#) untuk mendapatkan daftar suara bahasa Inggris AS di wilayah tersebut. `us-west-2`

Pilih Copy untuk menyimpan kode secara lokal.

Buat file `polly_describe_voices.rb`.

Tambahkan permata yang dibutuhkan.

 Note

Versi 2 AWS SDK for Ruby tidak memiliki permata khusus layanan.

```
# 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
```

Lihat [contoh lengkapnya](#) di GitHub.

## Mendapatkan Daftar Leksikon

Contoh ini menggunakan metode [list\\_lexicons](#) untuk mendapatkan daftar leksikon di wilayah tersebut. us-west-2

Pilih Copy untuk menyimpan kode secara lokal.

Buat file `polly_list_lexicons.rb`.

Tambahkan permata yang dibutuhkan.

### Note

Versi 2 AWS SDK for Ruby tidak memiliki permata khusus layanan.

```
# 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
  end
end
```

```
    puts ' Language:' + l.attributes.language
  puts
end
rescue StandardError => ex
  puts 'Could not get lexicons'
  puts 'Error message:'
  puts ex.message
end
```

Lihat [contoh lengkapnya](#) di GitHub.

## Mensintesis Pidato

Contoh ini menggunakan metode [synthesize\\_speech](#) untuk mendapatkan teks dari file dan menghasilkan file MP3 yang berisi ucapan yang disintesis.

Pilih Copy untuk menyimpan kode secara lokal.

Buat file `polly_synthesize_speech.rb`.

Tambahkan permata yang dibutuhkan.

### Note

Versi 2 AWS SDK for Ruby tidak memiliki permata khusus layanan.

```
# 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 -> 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
```

**Note**

File MP3 yang dihasilkan dalam format MPEG-2.

Lihat [contoh lengkapnya](#) di GitHub.

## Contoh Amazon RDS Menggunakan AWS SDK for Ruby

Amazon Relational Database Service (Amazon RDS) adalah layanan web yang memudahkan pengaturan, pengoperasian, dan skala database relasional di cloud. Anda dapat menggunakan contoh berikut untuk mengakses Amazon RDS menggunakan AWS SDK for Ruby. Untuk informasi selengkapnya tentang Amazon RDS, lihat dokumentasi [Amazon Relational Database Service](#).

**Note**

Beberapa contoh berikut menggunakan metode yang diperkenalkan dalam 2.2.18 versi `Aws::RDS::Resource` kelas. Untuk menjalankan contoh-contoh tersebut, Anda harus menggunakan versi itu atau versi `aws-sdk` permata yang lebih baru.

### Topik

- [Mendapatkan Informasi tentang Semua Instans Amazon RDS](#)
- [Mendapatkan Informasi tentang Semua Snapshot Amazon RDS](#)
- [Mendapatkan Informasi tentang Semua Cluster Amazon RDS dan Snapshots Mereka](#)
- [Mendapatkan Informasi tentang Semua Grup Keamanan Amazon RDS](#)
- [Mendapatkan Informasi tentang Semua Grup Subnet Amazon RDS](#)
- [Mendapatkan Informasi tentang Semua Grup Parameter Amazon RDS](#)
- [Membuat Snapshot dari Instans Amazon RDS](#)
- [Membuat Snapshot dari Amazon RDS Cluster](#)

## Mendapatkan Informasi tentang Semua Instans Amazon RDS

Contoh berikut mencantumkan nama (ID) dan status semua instans Amazon RDS Anda di wilayah tersebutus-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')

rds.db_instances.each do |i|
  puts "Name (ID): #{i.id}"
  puts "Status : #{i.db_instance_status}"
  puts
end
```

## Mendapatkan Informasi tentang Semua Snapshot Amazon RDS

Contoh berikut mencantumkan nama (ID) dan status semua snapshot Amazon RDS (instance) Anda di wilayah tersebutus-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')

rds.db_snapshots.each do |s|
  puts "Name (ID): #{s.snapshot_id}"
  puts "Status:    #{s.status}"
end
```

## Mendapatkan Informasi tentang Semua Cluster Amazon RDS dan Snapshots Mereka

Contoh berikut mencantumkan nama (ID) dan status semua kluster Amazon RDS Anda dan nama (ID) serta status snapshot mereka di wilayah tersebut. 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')

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
```



## Mendapatkan Informasi tentang Semua Grup Keamanan Amazon RDS

Contoh berikut mencantumkan nama semua grup keamanan Amazon RDS Anda di us-west-2 wilayah tersebut.

### Note

Grup keamanan Amazon RDS hanya berlaku saat Anda menggunakan platform Amazon EC2-Classic. Jika Anda menggunakan Amazon EC2-VPC, gunakan grup keamanan VPC. Keduanya ditunjukkan dalam contoh.

### Warning

Kami pensiun EC2-Classic pada 15 Agustus 2022. Kami menyarankan Anda bermigrasi dari EC2-Classic ke VPC. [Untuk informasi selengkapnya, lihat Memigrasi dari EC2-Classic ke VPC di Panduan Pengguna Amazon EC2 untuk Instans Linux atau Panduan Pengguna Amazon EC2 untuk Instans Windows.](#) Lihat juga posting blog [EC2-Classic Networking is Retiring - Inilah Cara Mempersiapkan.](#)

```
# 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
```

## Mendapatkan Informasi tentang Semua Grup Subnet Amazon RDS

Contoh berikut mencantumkan nama dan status semua grup subnet Amazon RDS Anda di wilayah tersebutus-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')

rds.db_subnet_groups.each do |s|
  puts s.name
  puts ' ' + s.subnet_group_status
end
```

## Mendapatkan Informasi tentang Semua Grup Parameter Amazon RDS

Contoh berikut mencantumkan nama dan deskripsi semua grup parameter Amazon RDS Anda di wilayah tersebutus-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')

rds.db_parameter_groups.each do |p|
  puts p.db_parameter_group_name
  puts ' ' + p.description
end
```

## Membuat Snapshot dari Instans Amazon RDS

Contoh berikut membuat snapshot untuk instance Amazon RDS yang diwakili oleh instance\_name di wilayah tersebut. us-west-2

### Note

Jika instance Anda adalah anggota klaster, Anda tidak dapat membuat snapshot instance. Sebagai gantinya, Anda harus membuat snapshot dari cluster (lihat [Membuat Snapshot dari Amazon RDS Cluster](#)).

```
# 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}"
```

## Membuat Snapshot dari Amazon RDS Cluster

Contoh berikut membuat snapshot untuk klaster Amazon RDS yang diwakili oleh `cluster_name` di wilayah tersebut. `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}"
```

## Contoh Amazon SES Menggunakan AWS SDK for Ruby

Amazon Simple Email Service (Amazon SES) adalah platform email yang menyediakan cara mudah dan hemat biaya bagi Anda untuk mengirim dan menerima email menggunakan alamat email dan domain Anda sendiri. Anda dapat menggunakan contoh berikut untuk mengakses Amazon SES menggunakan AWS SDK for Ruby. Untuk informasi selengkapnya tentang Amazon SES, lihat [dokumentasi Amazon SES](#).

### Topik

- [Daftar Alamat Email Amazon SES yang Valid](#)
- [Memverifikasi Alamat Email di Amazon SES](#)
- [Mengirim Pesan ke Alamat Email di Amazon SES](#)
- [Mendapatkan Statistik Amazon SES](#)

## Daftar Alamat Email Amazon SES yang Valid

Contoh berikut menunjukkan cara menggunakan AWS SDK for Ruby untuk mencantumkan alamat email Amazon SES yang valid.

```
# 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
```

Lihat [contoh lengkapnya](#) di GitHub.

## Memverifikasi Alamat Email di Amazon SES

Contoh berikut menunjukkan cara menggunakan AWS SDK for Ruby untuk memverifikasi alamat email Amazon SES.

```
# 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
```

Lihat [contoh lengkapnya](#) di GitHub.

## Mengirim Pesan ke Alamat Email di Amazon SES

Contoh berikut menunjukkan cara menggunakan AWS SDK for Ruby untuk mengirim pesan ke alamat email Amazon SES.

```
# 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>.'
```



```
]
},
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
```

Lihat [contoh lengkapnya](#) di GitHub.

## Mendapatkan Statistik Amazon SES

Contoh berikut menunjukkan cara menggunakan AWS SDK for Ruby untuk mendapatkan statistik tentang Amazon SES. Gunakan informasi ini untuk menghindari kerusakan reputasi Anda ketika email dipantulkan atau ditolak.

```
# 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
end
```

Lihat [contoh lengkapnya](#) di GitHub.

## Contoh Amazon SNS Menggunakan AWS SDK for Ruby

Amazon Simple Notification Service (Amazon SNS) adalah layanan web yang memungkinkan aplikasi, pengguna akhir, dan perangkat untuk langsung mengirim dan menerima notifikasi dari cloud.

Anda dapat menggunakan contoh berikut untuk mengakses Amazon SNS menggunakan AWS SDK for Ruby. Untuk informasi selengkapnya tentang Amazon SNS, lihat dokumentasi [Amazon SNS](#).

## Topik

- [Mendapatkan Informasi tentang Semua Topik Amazon SNS](#)
- [Membuat Topik Amazon SNS](#)
- [Mendapatkan Informasi tentang Semua Langganan dalam Topik Amazon SNS](#)
- [Membuat Langganan di Topik Amazon SNS](#)
- [Mengirim Pesan ke Semua Pelanggan Topik Amazon SNS](#)
- [Mengaktifkan Sumber Daya untuk Mempublikasikan ke Topik Amazon SNS](#)

## Mendapatkan Informasi tentang Semua Topik Amazon SNS

Contoh berikut mencantumkan ARN topik Amazon SNS Anda di us-west-2 wilayah tersebut.

```
# 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
```

## Membuat Topik Amazon SNS

Contoh berikut membuat topik `MyGroovyTopic` di us-west-2 wilayah dan menampilkan topik ARN yang dihasilkan.

```
# 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
```

## Mendapatkan Informasi tentang Semua Langganan dalam Topik Amazon SNS

Contoh berikut mencantumkan alamat email langganan Amazon SNS untuk topik dengan `arn:aws:sns:us-west-2:123456789:MyGroovyTopic` ARN di wilayah tersebut. `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-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
```

## Membuat Langganan di Topik Amazon SNS

Contoh berikut membuat langganan untuk topik dengan ARN `arn:aws:sns:us-west-2:123456789:MyGroovyTopic` untuk pengguna yang memiliki alamat email `MyGroovyUser@MyGroovy.com` di `us-west-2` wilayah tersebut, dan menampilkan ARN yang dihasilkan. Awalnya nilai ARN sedang menunggu konfirmasi. Ketika pengguna mengonfirmasi alamat email mereka, nilai ini menjadi ARN sejati.

```
# 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
```

## Mengirim Pesan ke Semua Pelanggan Topik Amazon SNS

Contoh berikut mengirimkan pesan “Halo!” untuk semua pelanggan topik Amazon SNS dengan ARN `arn:aws:sns:us-west-2:123456789:MyGroovyTopic`

```
# 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!'
})
```

## Mengaktifkan Sumber Daya untuk Mempublikasikan ke Topik Amazon SNS

Contoh berikut memungkinkan sumber daya dengan ARN `my-resource-arn` untuk mempublikasikan ke topik dengan ARN `my-topic-arn` di wilayah tersebut. `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-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
})
```

## Contoh Amazon SQS Menggunakan AWS SDK for Ruby

Amazon Simple Queue Service (Amazon SQS) adalah layanan antrian pesan yang dikelola sepenuhnya yang memudahkan untuk memisahkan dan menskalakan layanan mikro, sistem terdistribusi, dan aplikasi tanpa server. Anda dapat menggunakan contoh berikut untuk mengakses Amazon SQS menggunakan AWS SDK for Ruby. Untuk informasi selengkapnya tentang Amazon SQS, lihat dokumentasi [Amazon SQS](#).

### Topik

- [Mendapatkan Informasi tentang Semua Antrian di Amazon SQS](#)
- [Membuat Antrian di Amazon SQS](#)
- [Bekerja dengan Antrian di Amazon SQS](#)
- [Mengirim Pesan di Amazon SQS](#)
- [Mengirim dan Menerima Pesan di Amazon SQS](#)

- [Menerima Pesan di Amazon SQS](#)
- [Menerima Pesan Menggunakan Polling Panjang di Amazon SQS](#)
- [Mengaktifkan Polling Panjang di Amazon SQS](#)
- [Menerima Pesan Menggunakan QueuePoller Kelas di Amazon SQS](#)
- [Mengarahkan Surat Mati di Amazon SQS](#)
- [Menghapus Antrian di Amazon SQS](#)
- [Mengaktifkan Sumber Daya untuk Menerbitkan ke Antrian di Amazon SQS](#)
- [Bekerja dengan Antrian Surat Mati di Amazon SQS](#)
- [Menentukan Batas Waktu Visibilitas Pesan di Amazon SQS](#)

## Mendapatkan Informasi tentang Semua Antrian di Amazon SQS

Contoh berikut mencantumkan URL, ARN, pesan yang tersedia, dan pesan dalam penerbangan antrian Amazon SQS Anda di wilayah tersebut. us-west-2

```
# 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__
```

## Membuat Antrian di Amazon SQS

Contoh berikut membuat antrian Amazon SQS bernama MyGroovyQueue di us-west-2 wilayah dan menampilkan URL-nya.

```
# 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__
```

## Bekerja dengan Antrian di Amazon SQS

Amazon SQS menyediakan antrian host yang sangat skalabel untuk menyimpan pesan saat mereka melakukan perjalanan antar aplikasi atau layanan mikro. Untuk mempelajari lebih lanjut tentang antrian, lihat [Cara Kerja Antrian Amazon SQS](#).

Dalam contoh ini, Anda menggunakan AWS SDK for Ruby dengan Amazon SQS untuk:

1. Dapatkan daftar antrian Anda dengan menggunakan [Aws::SQS::Client#list\\_queues](#)
2. Buat antrian dengan menggunakan [Aws::SQS::Client#create\\_queue](#).
3. Dapatkan URL antrian dengan menggunakan [Aws::SQS::Client#get\\_queue\\_url](#).
4. Hapus antrian dengan menggunakan [Aws::SQS::Client#delete\\_queue](#).

### Prasyarat

Sebelum menjalankan kode contoh, Anda perlu menginstal dan mengkonfigurasi AWS SDK for Ruby, seperti yang dijelaskan dalam:

- [Memasang AWS SDK for Ruby](#)
- [Mengkonfigurasi AWS SDK for Ruby](#)

### Contoh

```
# 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)
```

## Mengirim Pesan di Amazon SQS

Contoh berikut mengirimkan pesan “Hello world” melalui antrian Amazon SQS dengan URL di wilayah tersebut. us-west-2

```
# 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__

```

Contoh berikut mengirimkan pesan “Hello world” dan “Bagaimana cuaca?” melalui antrian Amazon SQS dengan URL di wilayah tersebut. us-west-2

### Note

Jika antrian Anda adalah antrian FIFO, Anda harus menyertakan `message_group_id` parameter selain parameter `dan id.message_body`

```

# 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__
```

## Mengirim dan Menerima Pesan di Amazon SQS

Setelah Anda membuat antrian di Amazon SQS, Anda dapat mengirim pesan ke sana dan kemudian menggunakannya. Untuk mempelajari lebih lanjut, lihat [Tutorial: Mengirim Pesan ke Antrian dan Tutorial Amazon SQS: Menerima dan Menghapus Pesan dari Antrian Amazon SQS](#).

Dalam contoh ini, Anda menggunakan AWS SDK for Ruby dengan Amazon SQS untuk:

1. Kirim pesan ke antrian dengan menggunakan [Aws: :SQS: :Client](#) #send\_message.

### Note

Jika antrian Anda adalah antrian FIFO, Anda harus menyertakan `message_group_id` parameter selain parameter `danid`. `message_body`

1. Terima pesan dalam antrian dengan menggunakan [Aws: :SQS: :Client](#) #receive\_message.
2. Menampilkan informasi tentang pesan.
3. Hapus pesan dari antrian dengan menggunakan [Aws: :SQS: :Client](#) #delete\_message.

### Prasyarat

Sebelum menjalankan kode contoh, Anda perlu menginstal dan mengkonfigurasi AWS SDK for Ruby, seperti yang dijelaskan dalam:

- [Memasang AWS SDK for Ruby](#)
- [Mengkonfigurasi AWS SDK for Ruby](#)

Anda juga perlu membuat antrian `my-queue`, yang dapat Anda lakukan di konsol Amazon SQS.



## Contoh

```
# 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"
      },
    }
  })
end
```

```
        "WeeksOn" => {
          string_value: "6",
          data_type: "Number"
        }
      }
    })
  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
end
```

## Menerima Pesan di Amazon SQS

Contoh berikut menampilkan isi hingga 10 pesan dalam antrian Amazon SQS dengan URL di wilayah tersebut. us-west-2

**Note**

`receive_message` tidak menjamin untuk mendapatkan semua pesan (lihat [Properti Antrian Terdistribusi](#)), dan secara default tidak menghapus pesan.

```
# 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
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:
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__
```

## Menerima Pesan Menggunakan Polling Panjang di Amazon SQS

Contoh berikut menunggu hingga 10 detik untuk menampilkan badan hingga 10 pesan dalam antrian Amazon SQS dengan URL di wilayah tersebut. `us-west-2`

Jika Anda tidak menentukan waktu tunggu, nilai defaultnya adalah 0 (Amazon SQS tidak menunggu).

```
# 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
```

## Mengaktifkan Polling Panjang di Amazon SQS

Polling panjang membantu menurunkan biaya Anda menggunakan Amazon SQS dengan mengurangi jumlah respons kosong dan menghilangkan respons kosong palsu. Untuk informasi selengkapnya tentang polling panjang, lihat Polling Panjang [Amazon SQS](#).

Dalam contoh ini, Anda menggunakan AWS SDK for Ruby dengan Amazon SQS untuk:

1. Buat antrian dan atur untuk polling panjang dengan menggunakan [Aws::SQS::Client#create\\_queue](#).
2. Tetapkan polling panjang untuk antrian yang ada dengan menggunakan [Aws::SQS::Client#set\\_queue\\_attributes](#).
3. Setel polling panjang saat menerima pesan untuk antrian dengan menggunakan [Aws::SQS::Client#receive\\_message](#).

### Prasyarat

Sebelum menjalankan kode contoh, Anda perlu menginstal dan mengkonfigurasi AWS SDK for Ruby, seperti yang dijelaskan dalam:

- [Memasang AWS SDK for Ruby](#)

- [Mengkonfigurasi AWS SDK for Ruby](#)

Anda juga perlu membuat antrian yang ada-antrian dan antrian terima, yang dapat Anda lakukan di konsol Amazon SQS.

### Contoh

```
# 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
  (polls indefinitely by default).
```

```

}) do |messages|
  messages.each do |message|
    puts "Message body: #{message.body}"
  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

```

## Menerima Pesan Menggunakan QueuePoller Kelas di Amazon SQS

Contoh berikut menggunakan kelas `QueuePoller` utilitas untuk menampilkan isi semua pesan dalam antrian Amazon SQS dengan URL di `us-west-2` wilayah tersebut, dan menghapus pesan. Setelah sekitar 15 detik tidak aktif, waktu skrip habis.

```

# 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
```

Contoh berikut mengulang melalui antrian Amazon SQS dengan URLURL, dan menunggu hingga detik durasi.

Anda bisa mendapatkan URL yang benar dengan menjalankan contoh Amazon SQS di [Mendapatkan Informasi tentang Semua Antrian di Amazon SQS](#).

```
# 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
```

Contoh berikut mengulang melalui antrean Amazon SQS dengan URLURL, dan memberi Anda hingga detik batas waktu visibilitas untuk memproses pesan, yang diwakili oleh metode `do_something`

```
# 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
```

Contoh berikut mengulang melalui antrean Amazon SQS dengan URLURL, dan mengubah detik batas waktu visibilitas, untuk setiap pesan yang memerlukan pemrosesan tambahan oleh metode `do_something2`

```
# 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
```

## Mengarahkan Surat Mati di Amazon SQS

Contoh berikut mengalihkan setiap huruf mati dari antrian dengan URL ke antrian dengan 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-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}\"}"
}
```

## Menghapus Antrian di Amazon SQS

Contoh berikut menghapus antrian Amazon SQS dengan URL di wilayah tersebut. 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')

sqs.delete_queue(queue_url: URL)
```

## Mengaktifkan Sumber Daya untuk Menerbitkan ke Antrian di Amazon SQS

Contoh berikut memungkinkan sumber daya dengan ARN my-resource-arn untuk mempublikasikan ke antrian dengan ARN my-queue-arn dan URL di wilayah tersebut. my-queue-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-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
  }
})
```

## Bekerja dengan Antrian Surat Mati di Amazon SQS

Amazon SQS menyediakan dukungan untuk antrian surat mati. Antrian surat mati adalah antrian yang dapat ditargetkan antrian (sumber) lain untuk pesan yang tidak dapat diproses dengan sukses. Anda dapat menyisahkan dan mengisolasi pesan-pesan ini dalam antrian surat mati untuk menentukan mengapa pemrosesan mereka tidak berhasil. Untuk informasi selengkapnya tentang antrian surat mati, lihat [Menggunakan Antrian Surat Mati Amazon SQS](#).

Dalam contoh ini, Anda menggunakan AWS SDK for Ruby dengan Amazon SQS untuk:

1. Buat antrian yang mewakili antrian huruf mati dengan menggunakan [Aws: :SQS: :Client](#) `#create_queue`.
2. Kaitkan antrian huruf mati dengan antrian yang ada dengan menggunakan [Aws: :SQS: :Client](#) `#set_queue_attributes`.
3. Kirim pesan ke antrian yang ada dengan menggunakan [Aws: :SQS: :Client](#) `#send_message`.
4. Polling antrian dengan menggunakan [Aws: :SQS::](#) `QueuePoller`
5. Terima pesan dalam antrian surat mati dengan menggunakan [Aws: :SQS: :Client](#) `#receive_message`.

## Prasyarat

Sebelum menjalankan kode contoh, Anda perlu menginstal dan mengkonfigurasi AWS SDK for Ruby, seperti yang dijelaskan dalam:

- [Memasang AWS SDK for Ruby](#)
- [Mengkonfigurasi AWS SDK for Ruby](#)

Anda juga perlu menggunakan AWS Management Console untuk membuat antrian yang ada, antrian saya.

### Note

Demi kesederhanaan, kode contoh ini tidak menunjukkan [Aws: :SQS: :Client](#) `#add_permission`. Dalam skenario dunia nyata, Anda harus selalu membatasi akses ke tindakan seperti `SendMessage`, `ReceiveMessage` `DeleteMessage`, dan `DeleteQueue`. Tidak melakukannya dapat menyebabkan pengungkapan informasi, penolakan layanan, atau injeksi pesan ke dalam antrian Anda.

## Contoh

```
# 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
  }
})

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}"
end
```



```
else
  puts "\nNo messages received."
end
```

## Menentukan Batas Waktu Visibilitas Pesan di Amazon SQS

Di Amazon SQS, segera setelah pesan diterima, pesan tetap dalam antrian. Untuk mencegah konsumen lain memproses pesan lagi, Amazon SQS menetapkan batas waktu visibilitas. Ini adalah periode waktu di mana Amazon SQS mencegah komponen konsumsi lainnya menerima dan memproses pesan. Untuk mempelajari lebih lanjut, lihat Batas [Waktu Visibilitas](#).

Dalam contoh ini, Anda menggunakan AWS SDK for Ruby dengan Amazon SQS untuk:

1. Dapatkan URL antrian yang ada dengan menggunakan [Aws: :SQS: :Client #get\\_queue\\_url](#).
2. Terima hingga 10 pesan dengan menggunakan [Aws: :SQS: :Client #receive\\_message](#).
3. Tentukan interval waktu di mana pesan tidak terlihat setelah diterima, dengan menggunakan [Aws: :SQS: :Client #change\\_message\\_visibility](#).

### Prasyarat

Sebelum menjalankan kode contoh, Anda perlu menginstal dan mengkonfigurasi AWS SDK for Ruby, seperti yang dijelaskan dalam:

- [Memasang AWS SDK for Ruby](#)
- [Mengkonfigurasi AWS SDK for Ruby](#)

Anda juga perlu membuat antrian my-queue, yang dapat Anda lakukan di konsol Amazon SQS.

### Contoh

```
# 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

  # 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
```

## WorkDocs Contoh Amazon

Anda dapat menggunakan contoh berikut untuk mengakses Amazon WorkDocs (Amazon WorkDocs) menggunakan AWS SDK for Ruby. Untuk informasi selengkapnya tentang Amazon WorkDocs, lihat [WorkDocs dokumentasi Amazon](#).

Anda memerlukan ID organisasi Anda untuk menggunakan contoh-contoh ini. Dapatkan ID organisasi Anda dari AWS konsol menggunakan langkah-langkah berikut:

- Pilih AWS Directory Service
- Pilih Directories

ID organisasi Directory ID sesuai dengan WorkDocs situs Amazon Anda.

Contoh

Topik

- [Daftar Pengguna](#)
- [Daftar Dokumen Pengguna](#)

### Daftar Pengguna

Contoh berikut mencantumkan nama, alamat email, dan folder root dari semua pengguna dalam organisasi. Pilih Copy untuk menyimpan kode secara lokal, atau lihat tautan ke contoh lengkap di akhir topik ini.

1. Memerlukan modul AWS SDK for Ruby dan buat WorkDocs klien Amazon.
2. Panggil `describe_users` dengan ID organisasi Anda, dan dapatkan semua nama pengguna dalam urutan menaik.
  1. Menampilkan informasi tentang pengguna.

```
# 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
```

Lihat [contoh lengkapnya](#) di GitHub.

## Daftar Dokumen Pengguna

Contoh berikut mencantumkan dokumen untuk pengguna. Pilih Copy untuk menyimpan kode secara lokal, atau lihat tautan ke contoh lengkap di akhir topik ini.

1. Memerlukan AWS modul SDK for Ruby.
2. Buat metode pembantu untuk mendapatkan folder root pengguna.
3. Buat WorkDocs klien Amazon.

4. Dapatkan folder root untuk pengguna itu.
5. Panggil `describe_folder_contents` untuk mendapatkan konten folder dalam urutan menaik.
6. Menampilkan nama, ukuran (dalam byte), tanggal modifikasi terakhir, ID dokumen dan ID versi untuk setiap dokumen di folder root pengguna.

```
# 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

  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
```

Lihat [contoh lengkapnya](#) di GitHub.

# Contoh kode SDK for Ruby

Contoh kode dalam topik ini menunjukkan cara menggunakan AWS SDK for Ruby with AWS.

Tindakan merupakan kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Meskipun tindakan menunjukkan cara memanggil setiap fungsi layanan, Anda dapat melihat tindakan dalam konteks pada skenario yang terkait dan contoh lintas layanan.

Skenario adalah contoh kode yang menunjukkan cara untuk menyelesaikan tugas tertentu dengan memanggil beberapa fungsi dalam layanan yang sama.

Contoh lintas layanan adalah contoh aplikasi yang bekerja di beberapa Layanan AWS.

Contoh-contoh

- [Tindakan dan skenario menggunakan SDK for Ruby](#)
- [Contoh lintas layanan menggunakan SDK for Ruby](#)

## Tindakan dan skenario menggunakan SDK for Ruby

Contoh kode berikut menunjukkan cara melakukan tindakan dan mengimplementasikan skenario umum dengan menggunakan AWS SDK for Ruby with Layanan AWS.

Tindakan merupakan kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Meskipun tindakan menunjukkan cara memanggil setiap fungsi layanan, Anda dapat melihat tindakan dalam konteks pada skenario yang terkait dan contoh lintas layanan.

Skenario adalah contoh kode yang menunjukkan cara untuk menyelesaikan tugas tertentu dengan memanggil beberapa fungsi dalam layanan yang sama.

Layanan

- [CloudTrail contoh menggunakan SDK for Ruby](#)
- [CloudWatch contoh menggunakan SDK for Ruby](#)
- [Contoh DynamoDB menggunakan SDK for Ruby](#)
- [Contoh Amazon EC2 menggunakan SDK for Ruby](#)
- [Contoh Elastic Beanstalk menggunakan SDK for Ruby](#)

- [EventBridge contoh menggunakan SDK for Ruby](#)
- [AWS Glue contoh menggunakan SDK for Ruby](#)
- [Contoh IAM menggunakan SDK for Ruby](#)
- [Contoh Kinesis menggunakan SDK for Ruby](#)
- [AWS KMS contoh menggunakan SDK for Ruby](#)
- [Contoh Lambda menggunakan SDK for Ruby](#)
- [Contoh Amazon Polly menggunakan SDK for Ruby](#)
- [Contoh Amazon RDS menggunakan SDK for Ruby](#)
- [Contoh Amazon S3 menggunakan SDK for Ruby](#)
- [Contoh Amazon SES menggunakan SDK for Ruby](#)
- [Contoh Amazon SES API v2 menggunakan SDK for Ruby](#)
- [Contoh Amazon SNS menggunakan SDK for Ruby](#)
- [Contoh Amazon SQS menggunakan SDK for Ruby](#)
- [AWS STS contoh menggunakan SDK for Ruby](#)
- [WorkDocs Contoh Amazon menggunakan SDK for Ruby](#)

## CloudTrail contoh menggunakan SDK for Ruby

Contoh kode berikut menunjukkan cara melakukan tindakan dan menerapkan skenario umum dengan menggunakan AWS SDK for Ruby with CloudTrail.

Tindakan merupakan kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Meskipun tindakan menunjukkan cara memanggil setiap fungsi layanan, Anda dapat melihat tindakan dalam konteks pada skenario yang terkait dan contoh lintas layanan.

Skenario adalah contoh kode yang menunjukkan cara untuk menyelesaikan tugas tertentu dengan memanggil beberapa fungsi dalam layanan yang sama.

Setiap contoh menyertakan tautan ke GitHub, di mana Anda dapat menemukan petunjuk tentang cara mengatur dan menjalankan kode dalam konteks.

### Topik

- [Tindakan](#)



## Tindakan

Buat jalan setapak

Contoh kode berikut menunjukkan cara membuat AWS CloudTrail jejak.

SDK for Ruby

### Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
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


```

- Untuk detail API, lihat [CreateTrail](#) di Referensi AWS SDK for Ruby API.

## Hapus jejak

Contoh kode berikut menunjukkan cara menghapus AWS CloudTrail jejak.

## SDK for Ruby

 Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).


```
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
```

- Untuk detail API, lihat [DeleteTrail](#) di Referensi AWS SDK for Ruby API.

## Daftar acara jejak

Contoh kode berikut menunjukkan cara membuat daftar peristiwa AWS CloudTrail jejak.

## SDK for Ruby

 Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
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
end

```

- Untuk detail API, lihat [LookupEvents](#) di Referensi AWS SDK for Ruby API.

## Daftar jalur

Contoh kode berikut menunjukkan bagaimana untuk daftar AWS CloudTrail jejak.

## SDK for Ruby

### Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```

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

```

- Untuk detail API, lihat [ListTrails](#) di Referensi AWS SDK for Ruby API.

## CloudWatch contoh menggunakan SDK for Ruby

Contoh kode berikut menunjukkan cara melakukan tindakan dan menerapkan skenario umum dengan menggunakan AWS SDK for Ruby with CloudWatch.

Tindakan merupakan kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Meskipun tindakan menunjukkan cara memanggil setiap fungsi layanan, Anda dapat melihat tindakan dalam konteks pada skenario yang terkait dan contoh lintas layanan.

Skenario adalah contoh kode yang menunjukkan cara untuk menyelesaikan tugas tertentu dengan memanggil beberapa fungsi dalam layanan yang sama.

Setiap contoh menyertakan tautan ke GitHub, di mana Anda dapat menemukan petunjuk tentang cara mengatur dan menjalankan kode dalam konteks.

Topik

- [Tindakan](#)

### Tindakan

Buat alarm metrik

Contoh kode berikut menunjukkan cara membuat atau memperbarui CloudWatch alarm Amazon dan mengaitkannya dengan metrik yang ditentukan, ekspresi matematika metrik, model deteksi anomali, atau kueri Wawasan Metrik.

SDK for Ruby

#### Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
# 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
```

- Untuk detail API, lihat [PutMetricAlarm](#) di Referensi AWS SDK for Ruby API.

## Mendeskripsikan alarm

Contoh kode berikut menunjukkan cara mendeskripsikan CloudWatch alarm Amazon.

## SDK for Ruby

 Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

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

- Untuk detail API, lihat [DescribeAlarms](#) di Referensi AWS SDK for Ruby API.

Jelaskan alarm untuk metrik

Contoh kode berikut menunjukkan cara mendeskripsikan CloudWatch alarm Amazon untuk metrik.



## SDK for Ruby

 Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
#
# @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
end
```

```

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

run_me if $PROGRAM_NAME == __FILE__

```

- Untuk detail API, lihat [DescribeAlarmsForMetric](#) di Referensi AWS SDK for Ruby API.

## Menonaktifkan tindakan alarm

Contoh kode berikut menunjukkan cara menonaktifkan tindakan CloudWatch alarm Amazon.

### SDK for Ruby

#### Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```

# 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
end

```

```
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
end

run_me if $PROGRAM_NAME == __FILE__

```

- Untuk detail API, lihat [DisableAlarmActions](#) di Referensi AWS SDK for Ruby API.

## Membuat daftar metrik

Contoh kode berikut menunjukkan cara membuat daftar metadata untuk metrik Amazon CloudWatch . Untuk mendapatkan data untuk metrik, gunakan `GetMetricStatistics` tindakan `GetMetricData` atau.

## SDK for Ruby

### Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```

# 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__
```

- Untuk detail API, lihat [ListMetrics](#) di Referensi AWS SDK for Ruby API.

Masukkan data ke dalam metrik

Contoh kode berikut menunjukkan cara mempublikasikan titik data metrik ke Amazon CloudWatch.

## SDK for Ruby

 Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
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
```

- Untuk detail API, lihat [PutMetricData](#) di Referensi AWS SDK for Ruby API.

## Contoh DynamoDB menggunakan SDK for Ruby

Contoh kode berikut menunjukkan cara melakukan tindakan dan mengimplementasikan skenario umum dengan menggunakan AWS SDK for Ruby with DynamoDB.

Tindakan merupakan kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Meskipun tindakan menunjukkan cara memanggil setiap fungsi layanan, Anda dapat melihat tindakan dalam konteks pada skenario yang terkait dan contoh lintas layanan.

Skenario adalah contoh kode yang menunjukkan cara untuk menyelesaikan tugas tertentu dengan memanggil beberapa fungsi dalam layanan yang sama.

Setiap contoh menyertakan tautan ke GitHub, di mana Anda dapat menemukan petunjuk tentang cara mengatur dan menjalankan kode dalam konteks.

## Topik

- [Tindakan](#)
- [Skenario](#)

## Tindakan

### Buat tabel

Contoh kode berikut menunjukkan cara membuat tabel DynamoDB.

### SDK for Ruby

#### Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
# 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
    ],
    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

```

- Untuk detail API, lihat [CreateTable](#) di Referensi AWS SDK for Ruby API.

## Hapus tabel

Contoh kode berikut menunjukkan cara menghapus tabel DynamoDB.

## SDK for Ruby

### Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```

# 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

# 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
```

- Untuk detail API, lihat [DeleteTable](#) di Referensi AWS SDK for Ruby API.

## Menghapus item dalam tabel

Contoh kode berikut menunjukkan cara menghapus item dari tabel DynamoDB.

### SDK for Ruby

#### Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
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
end
```

```
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:")
  puts("\t#{e.code}: #{e.message}")
  raise
end
```

- Untuk detail API, lihat [DeleteItem](#) di Referensi AWS SDK for Ruby API.

## Mendapatkan item dari tabel

Contoh kode berikut menunjukkan bagaimana untuk mendapatkan item dari tabel DynamoDB.

## SDK for Ruby

### Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
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
```

- Untuk detail API, lihat [GetItem](#) di Referensi AWS SDK for Ruby API.

Dapatkan informasi tentang tabel

Contoh kode berikut menunjukkan bagaimana untuk mendapatkan informasi tentang tabel DynamoDB.

SDK for Ruby

#### Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
# 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}")
  raise
end

```

- Untuk detail API, lihat [DescribeTable](#) di Referensi AWS SDK for Ruby API.

## Mencantumkan tabel

Contoh kode berikut menunjukkan bagaimana daftar tabel DynamoDB.

## SDK for Ruby

### Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

Tentukan apakah tabel ada.

```

# 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}")
  raise
end

```

- Untuk detail API, lihat [ListTables](#) di Referensi AWS SDK for Ruby API.

## Menempatkan item dalam tabel

Contoh kode berikut menunjukkan bagaimana menempatkan item dalam tabel DynamoDB.

### SDK for Ruby

#### Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
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]})
  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
end

```

- Untuk detail API, lihat [PutItem](#) di Referensi AWS SDK for Ruby API.

## Mengkueri Tabel

Contoh kode berikut menunjukkan bagaimana untuk query tabel DynamoDB.

### SDK for Ruby

#### Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```

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}")
    raise
  else
    response.items
  end
end

```

- Untuk detail API, lihat [Query](#) di Referensi API AWS SDK for Ruby .

## Menjalankan pernyataan PartiQL

Contoh kode berikut menunjukkan bagaimana menjalankan pernyataan PartiQL pada tabel DynamoDB.

### SDK for Ruby

#### Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

Pilih satu item menggunakan PartiQL.

```

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]
    }
    @dynamodb.client.execute_statement(request)
  end
end

```

Perbarui satu item menggunakan PartiQL.

```

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

```

Tambahkan satu item menggunakan PartiQL.

```

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
end

```

Hapus satu item menggunakan PartiQL.

```
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
end
```

- Untuk detail API, lihat [ExecuteStatement](#) di Referensi AWS SDK for Ruby API.

Jalankan batch pernyataan PartiQL

Contoh kode berikut menunjukkan bagaimana menjalankan batch pernyataan PartiQL pada tabel DynamoDB.

SDK for Ruby

#### Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

Baca batch item menggunakan PartiQL.

```

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
    @dynamodb.client.batch_execute_statement({statements: request_items})
  end
end

```

Hapus batch item menggunakan PartiQL.

```

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
```

- Untuk detail API, lihat [BatchExecuteStatement](#) di Referensi AWS SDK for Ruby API.

## Memindai tabel

Contoh kode berikut menunjukkan bagaimana untuk memindai tabel DynamoDB.

## SDK for Ruby

### Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
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
rescue Aws::DynamoDB::Errors::ServiceError => e
  puts("Couldn't scan for movies. Here's why:")
  puts("\t#{e.code}: #{e.message}")
  raise
else
  movies
end

```

- Untuk detail API, lihat [Scan](#) di Referensi API AWS SDK for Ruby .

## Memperbarui item dalam tabel

Contoh kode berikut menunjukkan cara memperbarui item dalam tabel DynamoDB.

### SDK for Ruby

#### Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```

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")
    puts("\t#{e.code}: #{e.message}")
    raise
  else
    response.attributes
  end
end

```

- Untuk detail API, lihat [UpdateItem](#) di Referensi AWS SDK for Ruby API.

## Menulis sekumpulan item

Contoh kode berikut menunjukkan bagaimana menulis batch item DynamoDB.

### SDK for Ruby

#### Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
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
end
```

- Untuk detail API, lihat [BatchWriteItem](#) di Referensi AWS SDK for Ruby API.

## Skenario

Memulai tabel, item, dan kueri

Contoh kode berikut ini menunjukkan cara untuk melakukan:

- Buat tabel yang dapat menyimpan data film.
- Masukkan, dapatkan, dan perbarui satu film dalam tabel tersebut.
- Tulis data film ke tabel dari file JSON sampel.
- Kueri untuk film yang dirilis pada tahun tertentu.
- Pindai film yang dirilis dalam suatu rentang tahun.
- Hapus film dari tabel, lalu hapus tabel tersebut.

SDK for Ruby

### Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

Buat kelas yang merangkum tabel DynamoDB.

```
# 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

```

Buat fungsi pembantu untuk mengunduh dan mengekstrak file JSON sampel.

```

# 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

```

Jalankan skenario interaktif untuk membuat tabel dan melakukan tindakan pada tabel tersebut.

```
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
```

- Lihat detail API di topik-topik berikut dalam Referensi API AWS SDK for Ruby .

- [BatchWriteItem](#)
- [CreateTable](#)
- [DeleteItem](#)
- [DeleteTable](#)
- [DescribeTable](#)
- [GetItem](#)
- [PutItem](#)
- [Query](#)
- [Scan](#)
- [UpdateItem](#)

## Melakukan kueri pada tabel menggunakan batch pernyataan PartiQL

Contoh kode berikut ini menunjukkan cara untuk melakukan:

- Dapatkan batch item dengan menjalankan beberapa pernyataan SELECT.
- Tambahkan batch item dengan menjalankan beberapa pernyataan INSERT.
- Perbarui batch item dengan menjalankan beberapa pernyataan UPDATE.
- Hapus batch item dengan menjalankan beberapa pernyataan DELETE.

### SDK for Ruby

#### Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

Jalankan skenario yang membuat tabel dan menjalankan batch kueri PartiQL.

```
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
end

```

- Untuk detail API, lihat [BatchExecuteStatement](#) di Referensi AWS SDK for Ruby API.

## Melakukan kueri tabel menggunakan PartiQL

Contoh kode berikut ini menunjukkan cara untuk melakukan:

- Dapatkan item dengan menjalankan pernyataan SELECT.
- Tambahkan item dengan menjalankan pernyataan INSERT.
- Perbarui item dengan menjalankan pernyataan UPDATE.
- Hapus item dengan menjalankan pernyataan DELETE.

## SDK for Ruby

### Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

Jalankan skenario yang membuat tabel dan menjalankan kueri PartiQL.

```

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 "\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
```

- Untuk detail API, lihat [ExecuteStatement](#) di Referensi AWS SDK for Ruby API.

## Contoh Amazon EC2 menggunakan SDK for Ruby

Contoh kode berikut menunjukkan cara melakukan tindakan dan menerapkan skenario umum dengan menggunakan Amazon EC2. AWS SDK for Ruby

Tindakan merupakan kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Meskipun tindakan menunjukkan cara memanggil setiap fungsi layanan, Anda dapat melihat tindakan dalam konteks pada skenario yang terkait dan contoh lintas layanan.

Skenario adalah contoh kode yang menunjukkan cara untuk menyelesaikan tugas tertentu dengan memanggil beberapa fungsi dalam layanan yang sama.

Setiap contoh menyertakan tautan ke GitHub, di mana Anda dapat menemukan petunjuk tentang cara mengatur dan menjalankan kode dalam konteks.

Topik

- [Tindakan](#)

### Tindakan

Mengalokasikan alamat IP Elastis

Contoh kode berikut menunjukkan cara mengalokasikan alamat IP Elastis untuk Amazon EC2.

SDK for Ruby

#### Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
# 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
```

- Untuk detail API, lihat [AllocateAddress](#) di Referensi AWS SDK for Ruby API.

## Mengaitkan alamat IP Elastis dengan instans

Contoh kode berikut menunjukkan cara mengaitkan alamat IP Elastis dengan instans Amazon EC2.

### SDK for Ruby

#### Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
# 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

```

- Untuk detail API, lihat [AssociateAddress](#) di Referensi AWS SDK for Ruby API.

## Buat Amazon Virtual Private Cloud (Amazon VPC)

Contoh kode berikut ini menunjukkan cara membuat Amazon Virtual Private Cloud (Amazon VPC).

### SDK for Ruby

#### Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```

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__
```

- Untuk detail API, lihat [CreateVpc](#) Referensi AWS SDK for Ruby API.

## Membuat tabel rute

Contoh kode berikut menunjukkan cara membuat tabel rute dan mengasosiasikan dengan Amazon EC2 subnet.

## SDK for Ruby

 Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
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
```

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

- Untuk detail API, lihat [CreateRouteTable](#) di Referensi AWS SDK for Ruby API.

## Membuat grup keamanan

Contoh kode berikut menunjukkan cara membuat grup keamanan Amazon EC2.

## SDK for Ruby

### Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
# 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?(:ipv6_ranges) && perm.ipv6_ranges.count.positive?
    print ", CIDR IPv6: #{perm.ipv6_ranges[0].cidr_ipv6}"
  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"
  end
  exit 1
end

```



```
# 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

run_me if $PROGRAM_NAME == __FILE__
```

- Untuk detail API, lihat [CreateSecurityGroup](#) di Referensi AWS SDK for Ruby API.

## Membuat pasangan kunci

Contoh kode berikut menunjukkan cara membuat security key pair untuk Amazon EC2.

## SDK for Ruby

### Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
# 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}'."
  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
end
```

```

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__
```

- Untuk detail API, lihat [CreateKeyPair](#) di Referensi AWS SDK for Ruby API.

## Membuat subnet

Contoh kode berikut ini menunjukkan cara membuat subnet Amazon EC2.

### SDK for Ruby

#### Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
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
  )
  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
end

run_me if $PROGRAM_NAME == __FILE__
```

- Untuk detail API, lihat [CreateSubnet](#) di Referensi AWS SDK for Ruby API.

## Menjelaskan Wilayah

Contoh kode berikut menunjukkan cara mendeskripsikan Wilayah Amazon EC2.

### SDK for Ruby

#### Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
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
end
```

```

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__

```

- Untuk detail API, lihat [DescribeRegions](#) di Referensi AWS SDK for Ruby API.

## Menjelaskan instans

Contoh kode berikut menunjukkan cara mendeskripsikan instans Amazon EC2.

### SDK for Ruby

#### Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```

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

```

```
    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__
```

- Untuk detail API, lihat [DescribeInstances](#) di Referensi AWS SDK for Ruby API.

## Merilis alamat IP Elastis

Contoh kode berikut menunjukkan cara melepaskan alamat IP Elastis.

## SDK for Ruby

### Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
# 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
```

- Untuk detail API, lihat [ReleaseAddress](#) di Referensi AWS SDK for Ruby API.

## Memulai instans

Contoh kode berikut menunjukkan cara memulai instans Amazon EC2.

## SDK for Ruby

### Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
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__
```

- Untuk detail API, lihat [StartInstances](#) di Referensi AWS SDK for Ruby API.

## Menghentikan instans

Contoh kode berikut menunjukkan cara menghentikan instans Amazon EC2.



## SDK for Ruby

 Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
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__
```

- Untuk detail API, lihat [StopInstances](#) di Referensi AWS SDK for Ruby API.

## Akhiri instans

Contoh kode berikut menunjukkan cara menghentikan instans Amazon EC2.

## SDK for Ruby

### Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
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

# 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__
```

- Untuk detail API, lihat [TerminateInstances](#) di Referensi AWS SDK for Ruby API.

## Contoh Elastic Beanstalk menggunakan SDK for Ruby

Contoh kode berikut menunjukkan cara melakukan tindakan dan menerapkan skenario umum dengan menggunakan AWS SDK for Ruby with Elastic Beanstalk.

Tindakan merupakan kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Meskipun tindakan menunjukkan cara memanggil setiap fungsi layanan, Anda dapat melihat tindakan dalam konteks pada skenario yang terkait dan contoh lintas layanan.

Skenario adalah contoh kode yang menunjukkan cara untuk menyelesaikan tugas tertentu dengan memanggil beberapa fungsi dalam layanan yang sama.

Setiap contoh menyertakan tautan ke GitHub, di mana Anda dapat menemukan petunjuk tentang cara mengatur dan menjalankan kode dalam konteks.

Topik

- [Tindakan](#)

### Tindakan

Jelaskan aplikasi

Contoh kode berikut menunjukkan cara mendeskripsikan AWS Elastic Beanstalk aplikasi.

SDK for Ruby

#### Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
# Class to manage Elastic Beanstalk applications
class ElasticBeanstalkManager
  def initialize(eb_client, logger: Logger.new($stdout))
    @eb_client = eb_client
    @logger = logger
  end
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

# 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
```

- Untuk detail API, lihat [DescribeApplications](#) di Referensi AWS SDK for Ruby API.

## Daftar tumpukan

Contoh kode berikut menunjukkan bagaimana daftar AWS Elastic Beanstalk tumpukan.

## SDK for Ruby

 Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
# 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
```

- Untuk detail API, lihat [ListAvailableSolutionStacks](#) di Referensi AWS SDK for Ruby API.

## Perbarui aplikasi

Contoh kode berikut menunjukkan cara memperbarui AWS Elastic Beanstalk aplikasi.

## SDK for Ruby

### Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
# 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
    )
  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
end

```

- Untuk detail API, lihat [UpdateApplication](#) di Referensi AWS SDK for Ruby API.

## EventBridge contoh menggunakan SDK for Ruby

Contoh kode berikut menunjukkan cara melakukan tindakan dan menerapkan skenario umum dengan menggunakan AWS SDK for Ruby with EventBridge.

Tindakan merupakan kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Meskipun tindakan menunjukkan cara memanggil setiap fungsi layanan, Anda dapat melihat tindakan dalam konteks pada skenario yang terkait dan contoh lintas layanan.

Skenario adalah contoh kode yang menunjukkan cara untuk menyelesaikan tugas tertentu dengan memanggil beberapa fungsi dalam layanan yang sama.

Setiap contoh menyertakan tautan ke GitHub, di mana Anda dapat menemukan petunjuk tentang cara mengatur dan menjalankan kode dalam konteks.

Topik

- [Skenario](#)

## Skenario

Buat dan picu aturan

Contoh kode berikut menunjukkan cara membuat dan memicu aturan di Amazon EventBridge.

SDK for Ruby

### Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

Panggil fungsi dalam urutan yang benar.

```
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"
```

Memeriksa apakah topik Amazon Simple Notification Service (Amazon SNS) yang ditentukan ada di antara topik yang disediakan untuk fungsi ini.

```
# 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
  return false
end

```

Memeriksa apakah topik yang ditentukan ada di antara yang tersedia untuk penelepon di Amazon SNS.

```

# 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
  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
end
puts "Topic not found."
return false
rescue StandardError => e
  puts "Topic not found: #{e.message}"
  return false
end
end

```

Buat topik di Amazon SNS dan kemudian berlangganan alamat email untuk menerima pemberitahuan tentang topik itu.

```

# 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

```

Periksa apakah peran yang ditentukan AWS Identity and Access Management (IAM) ada di antara yang disediakan untuk fungsi ini.

```

# 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

```

Periksa apakah peran yang ditentukan ada di antara yang tersedia untuk pemanggil di IAM.

```

# 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
  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

```

## Buat peran dalam IAM.

```

# 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"
        }
      ]
    }
  )
end
```



```

    }
  ]
}.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

```

Memeriksa apakah EventBridge aturan yang ditentukan ada di antara yang disediakan untuk fungsi ini.

```

# 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

```

Memeriksa apakah aturan yang ditentukan ada di antara yang tersedia untuk pemanggil di 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
  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
  puts "Rule not found."
  return false
rescue StandardError => e
  puts "Rule not found: #{e.message}"
  return false
end

```

Buat aturan di 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
```

Periksa untuk melihat apakah grup log yang ditentukan ada di antara yang tersedia untuk pemanggil di Amazon CloudWatch Logs.

```
# 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
```

Buat grup log di CloudWatch Log.

```
# 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

```

Tulis acara ke aliran log di CloudWatch Log.

```

# 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

```

Mulai ulang instance Amazon Elastic Compute Cloud (Amazon EC2) dan tambahkan informasi tentang aktivitas terkait ke aliran log di Log. CloudWatch

```

# 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
  )
end
```



```

)

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

```

Menampilkan informasi tentang aktivitas untuk aturan di 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
end
rescue StandardError => e
```

```
puts "Error getting information about event rule activity: #{e.message}"
end
```

Menampilkan informasi log untuk semua aliran log dalam grup CloudWatch log Log.

```
# 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
      end
    end
  end
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

```

Tampilkan pengingat ke penelepon untuk membersihkan AWS sumber daya terkait secara manual yang tidak lagi mereka butuhkan.

```

# 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."
    end
  end
end
```

```
    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__
```

- Lihat detail API di topik-topik berikut dalam Referensi API AWS SDK for Ruby .
  - [PutEvents](#)
  - [PutRule](#)

## AWS Glue contoh menggunakan SDK for Ruby

Contoh kode berikut menunjukkan cara melakukan tindakan dan menerapkan skenario umum dengan menggunakan AWS SDK for Ruby with AWS Glue.

Tindakan merupakan kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Meskipun tindakan menunjukkan cara memanggil setiap fungsi layanan, Anda dapat melihat tindakan dalam konteks pada skenario yang terkait dan contoh lintas layanan.

Skenario adalah contoh kode yang menunjukkan cara untuk menyelesaikan tugas tertentu dengan memanggil beberapa fungsi dalam layanan yang sama.

Setiap contoh menyertakan tautan ke GitHub, di mana Anda dapat menemukan petunjuk tentang cara mengatur dan menjalankan kode dalam konteks.

Topik

- [Tindakan](#)
- [Skenario](#)

### Tindakan

Buat crawler

Contoh kode berikut menunjukkan cara membuat AWS Glue crawler.

SDK for Ruby

#### Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
# 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
          }
        ]
      }
    )
  rescue Aws::Glue::Errors::GlueException => e
    @logger.error("Glue could not create crawler: \n#{e.message}")
    raise
  end
end

```

- Untuk detail API, lihat [CreateCrawler](#) di Referensi AWS SDK for Ruby API.

Buat definisi pekerjaan

Contoh kode berikut menunjukkan cara membuat definisi AWS Glue pekerjaan.

## SDK for Ruby

 Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
# 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}")
  end
end
```

```
    raise
  end
```

- Untuk detail API, lihat [CreateJob](#) di Referensi AWS SDK for Ruby API.

## Hapus crawler

Contoh kode berikut menunjukkan cara menghapus AWS Glue crawler.

## SDK for Ruby

### Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
# 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
end
```

- Untuk detail API, lihat [DeleteCrawler](#) di Referensi AWS SDK for Ruby API.

## Menghapus database dari Katalog Data

Contoh kode berikut menunjukkan cara menghapus database dari file AWS Glue Data Catalog.

### SDK for Ruby

#### Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
# 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
end
```

- Untuk detail API, lihat [DeleteDatabase](#) di Referensi AWS SDK for Ruby API.

## Hapus definisi pekerjaan

Contoh kode berikut menunjukkan cara menghapus definisi AWS Glue pekerjaan dan semua proses terkait.

### SDK for Ruby

#### Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
# 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
```

- Untuk detail API, lihat [DeleteJob](#) di Referensi AWS SDK for Ruby API.

## Menghapus tabel dari database

Contoh kode berikut menunjukkan cara menghapus tabel dari AWS Glue Data Catalog database.

### SDK for Ruby

#### Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
# 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
```

- Untuk detail API, lihat [DeleteTable](#) di Referensi AWS SDK for Ruby API.

## Dapatkan crawler

Contoh kode berikut menunjukkan cara mendapatkan AWS Glue crawler.

### SDK for Ruby

#### Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
# 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
```

- Untuk detail API, lihat [GetCrawler](#) di Referensi AWS SDK for Ruby API.

## Dapatkan database dari Katalog Data

Contoh kode berikut menunjukkan bagaimana untuk mendapatkan database dari AWS Glue Data Catalog.

### SDK for Ruby

#### Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
# 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
```

- Untuk detail API, lihat [GetDatabase](#) di Referensi AWS SDK for Ruby API.



## Dapatkan pekerjaan

Contoh kode berikut menunjukkan cara menjalankan AWS Glue pekerjaan.

### SDK for Ruby

#### Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
# 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
```

- Untuk detail API, lihat [GetJobRun](#) di Referensi AWS SDK for Ruby API.

## Menjalankan pekerjaan

Contoh kode berikut menunjukkan cara menjalankan AWS Glue pekerjaan.

## SDK for Ruby

 Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
# 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
```

- Untuk detail API, lihat [GetJobRuns](#) di Referensi AWS SDK for Ruby API.

Dapatkan tabel dari database

Contoh kode berikut menunjukkan bagaimana untuk mendapatkan tabel dari database di AWS Glue Data Catalog.

## SDK for Ruby

 Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
# 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
```

- Untuk detail API, lihat [GetTables](#) di Referensi AWS SDK for Ruby API.

## Daftar definisi pekerjaan

Contoh kode berikut menunjukkan bagaimana daftar definisi AWS Glue pekerjaan.

## SDK for Ruby

 Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
# 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
```

- Untuk detail API, lihat [ListJobs](#) di Referensi AWS SDK for Ruby API.

## Mulai crawler

Contoh kode berikut menunjukkan cara memulai AWS Glue crawler.

## SDK for Ruby

 Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
# 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
```

- Untuk detail API, lihat [StartCrawler](#) di Referensi AWS SDK for Ruby API.

## Memulai eksekusi tugas

Contoh kode berikut menunjukkan cara memulai AWS Glue pekerjaan.

## SDK for Ruby

 Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
# 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
```

- Untuk detail API, lihat [StartJobRun](#) di Referensi AWS SDK for Ruby API.

## Skenario

Memulai crawler dan lowongan

Contoh kode berikut ini menunjukkan cara untuk melakukan:

- Buat crawler yang merayapi bucket Amazon S3 publik dan membuat database metadata berformat CSV.
- Daftar informasi tentang database dan tabel di situs Anda AWS Glue Data Catalog.
- Buat pekerjaan untuk mengekstrak data CSV dari bucket S3, mengubah data, dan memuat output berformat JSON ke bucket S3 lain.
- Buat daftar informasi tentang menjalankan pekerjaan, melihat data yang diubah, dan membersihkan sumber daya.

Untuk informasi selengkapnya, lihat [Tutorial: Memulai AWS Glue Studio](#).

SDK for Ruby

### Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

Buat kelas yang membungkus AWS Glue fungsi yang digunakan dalam skenario.

```
# 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}")
end
```



```
    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}/"
    }
  )
end
```

```

    }
  )
  response.job_run_id
rescue Aws::Glue::Errors::GlueException => e
  @logger.error("Glue could not start job run #{name}: \n#{e.message}")
  raise
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
end
rescue Aws::S3::Errors::S3UploadFailedError => e
  @logger.error("S3 could not upload job script: \n#{e.message}")
  raise
```

```
end

end
```

Buat kelas yang menjalankan skenario.

```
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
  end
end
```

```
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

  # 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

  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 "#          EXAMPLE CODE DEMO:
          #".yellow
  puts "#          AWS Glue
          #".yellow
  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

```

Buat skrip ETL yang digunakan oleh AWS Glue untuk mengekstrak, mengubah, dan memuat data selama pekerjaan berjalan.

```

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()
```

- Lihat detail API di topik-topik berikut dalam Referensi API AWS SDK for Ruby .

- [CreateCrawler](#)
- [CreateJob](#)
- [DeleteCrawler](#)
- [DeleteDatabase](#)
- [DeleteJob](#)
- [DeleteTable](#)
- [GetCrawler](#)
- [GetDatabase](#)
- [GetDatabases](#)
- [GetJob](#)
- [GetJobRun](#)
- [GetJobRuns](#)
- [GetTables](#)
- [ListJobs](#)
- [StartCrawler](#)
- [StartJobRun](#)

## Contoh IAM menggunakan SDK for Ruby

Contoh kode berikut menunjukkan cara melakukan tindakan dan mengimplementasikan skenario umum dengan menggunakan AWS SDK for Ruby with IAM.

Tindakan merupakan kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Meskipun tindakan menunjukkan cara memanggil setiap fungsi layanan, Anda dapat melihat tindakan dalam konteks pada skenario yang terkait dan contoh lintas layanan.

Skenario adalah contoh kode yang menunjukkan cara untuk menyelesaikan tugas tertentu dengan memanggil beberapa fungsi dalam layanan yang sama.

Setiap contoh menyertakan tautan ke GitHub, di mana Anda dapat menemukan petunjuk tentang cara mengatur dan menjalankan kode dalam konteks.

Topik

- [Tindakan](#)
- [Skenario](#)

### Tindakan

Lampirkan kebijakan ke peran

Contoh kode berikut menunjukkan cara melampirkan kebijakan IAM ke peran.

SDK for Ruby

#### Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

Modul contoh ini mencantumkan, membuat, melampirkan, dan melepaskan kebijakan peran.

```
# 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
```

- Untuk detail API, lihat [AttachRolePolicy](#) di Referensi AWS SDK for Ruby API.

## Lampirkan kebijakan ke pengguna

Contoh kode berikut menunjukkan cara melampirkan kebijakan IAM ke pengguna.

### Warning

Untuk menghindari risiko keamanan, jangan gunakan pengguna IAM untuk otentikasi saat mengembangkan perangkat lunak yang dibuat khusus atau bekerja dengan data nyata. Sebaliknya, gunakan federasi dengan penyedia identitas seperti [AWS IAM Identity Center](#).

## SDK for Ruby

### Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
# 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
```

- Untuk detail API, lihat [AttachUserPolicy](#) di Referensi AWS SDK for Ruby API.

## Buat kebijakan

Contoh kode berikut menunjukkan cara membuat kebijakan IAM.

## SDK for Ruby

 Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

Modul contoh ini mencantumkan, membuat, melampirkan, dan melepaskan kebijakan peran.

```
# 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
```

- Untuk detail API, lihat [CreatePolicy](#) di Referensi AWS SDK for Ruby API.

## Membuat peran

Contoh kode berikut menunjukkan cara membuat peran IAM.

## SDK for Ruby

### Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
# 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

```

- Untuk detail API, lihat [CreateRole](#) di Referensi AWS SDK for Ruby API.

Buat peran tertaut layanan

Contoh kode berikut menunjukkan cara membuat peran terkait layanan IAM.

SDK for Ruby

#### Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```

# 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
end

```

```
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
```

- Untuk detail API, lihat [CreateServiceLinkedRole](#) di Referensi AWS SDK for Ruby API.

## Buat pengguna

Contoh kode berikut menunjukkan cara membuat pengguna IAM.

### Warning

Untuk menghindari risiko keamanan, jangan gunakan pengguna IAM untuk otentikasi saat mengembangkan perangkat lunak yang dibuat khusus atau bekerja dengan data nyata. Sebaliknya, gunakan federasi dengan penyedia identitas seperti [AWS IAM Identity Center](#).

## SDK for Ruby

### Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
# 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
```

- Untuk detail API, lihat [CreateUser](#) di Referensi AWS SDK for Ruby API.

## Buat kunci akses

Contoh kode berikut menunjukkan cara membuat kunci akses IAM.

### Warning

Untuk menghindari risiko keamanan, jangan gunakan pengguna IAM untuk otentikasi saat mengembangkan perangkat lunak yang dibuat khusus atau bekerja dengan data nyata. Sebaliknya, gunakan federasi dengan penyedia identitas seperti [AWS IAM Identity Center](#).

## SDK for Ruby

### Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

Modul contoh ini mencantumkan, membuat, menonaktifkan, dan menghapus kunci akses.

```
# 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
end
```

```
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
  )
  true
rescue StandardError => e
  @logger.error("Error deleting access key: #{e.message}")
  false
end
end
```

- Untuk detail API, lihat [CreateAccessKey](#) di Referensi AWS SDK for Ruby API.

Buat alias untuk akun

Contoh kode berikut menunjukkan cara membuat alias untuk akun IAM.

SDK for Ruby

 Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

## Buat daftar, buat, dan hapus alias akun.

```
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)
  end
end
```



```
    true
  rescue Aws::IAM::Errors::ServiceError => e
    @logger.error("Error deleting account alias: #{e.message}")
    false
  end
end
```

- Untuk detail API, lihat [CreateAccountAlias](#) di Referensi AWS SDK for Ruby API.

## Membuat kebijakan inline untuk pengguna

Contoh kode berikut menunjukkan cara membuat kebijakan IAM sebaris untuk pengguna.

### Warning

Untuk menghindari risiko keamanan, jangan gunakan pengguna IAM untuk otentikasi saat mengembangkan perangkat lunak yang dibuat khusus atau bekerja dengan data nyata. Sebaliknya, gunakan federasi dengan penyedia identitas seperti [AWS IAM Identity Center](#).

## SDK for Ruby

### Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
# 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}.")
end
```

```

    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}")
  end
end

```

- Untuk detail API, lihat [PutUserPolicy](#) di Referensi AWS SDK for Ruby API.

## Hapus peran

Contoh kode berikut menunjukkan cara menghapus peran IAM.

## SDK for Ruby

### Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```

# 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
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
end

```

- Untuk detail API, lihat [DeleteRole](#) di Referensi AWS SDK for Ruby API.

## Hapus sertifikat server

Contoh kode berikut menunjukkan cara menghapus sertifikat server IAM.

### SDK for Ruby

#### Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

Daftar, perbarui, dan hapus sertifikat server.

```

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
```

- Untuk detail API, lihat [DeleteServerCertificate](#) di Referensi AWS SDK for Ruby API.

## Hapus peran tertaut layanan

Contoh kode berikut menunjukkan cara menghapus peran terkait layanan IAM.

### SDK for Ruby

#### Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
# 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}")
    raise
  end
end
```

- Untuk detail API, lihat [DeleteServiceLinkedRole](#) di Referensi AWS SDK for Ruby API.

## Hapus pengguna

Contoh kode berikut menunjukkan cara menghapus pengguna IAM.

### Warning

Untuk menghindari risiko keamanan, jangan gunakan pengguna IAM untuk otentikasi saat mengembangkan perangkat lunak yang dibuat khusus atau bekerja dengan data nyata. Sebaliknya, gunakan federasi dengan penyedia identitas seperti [AWS IAM Identity Center](#).

## SDK for Ruby

 Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).


```
# 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
```

- Untuk detail API, lihat [DeleteUser](#) di Referensi AWS SDK for Ruby API.

## Hapus kunci akses

Contoh kode berikut menunjukkan cara menghapus kunci akses IAM.

 Warning

Untuk menghindari risiko keamanan, jangan gunakan pengguna IAM untuk otentikasi saat mengembangkan perangkat lunak yang dibuat khusus atau bekerja dengan data nyata. Sebaliknya, gunakan federasi dengan penyedia identitas seperti [AWS IAM Identity Center](#).

## SDK for Ruby

 Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

Modul contoh ini mencantumkan, membuat, menonaktifkan, dan menghapus kunci akses.

```
# 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
  end
end
```



```
@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
```

- Untuk detail API, lihat [DeleteAccessKey](#) di Referensi AWS SDK for Ruby API.

## Hapus alias akun

Contoh kode berikut menunjukkan cara menghapus alias akun IAM.

## SDK for Ruby

### Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

Buat daftar, buat, dan hapus alias akun.

```
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
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
```

- Untuk detail API, lihat [DeleteAccountAlias](#) di Referensi AWS SDK for Ruby API.

## Menghapus kebijakan inline dari pengguna

Contoh kode berikut menunjukkan cara menghapus kebijakan IAM inline dari pengguna.

### Warning

Untuk menghindari risiko keamanan, jangan gunakan pengguna IAM untuk otentikasi saat mengembangkan perangkat lunak yang dibuat khusus atau bekerja dengan data nyata. Sebaliknya, gunakan federasi dengan penyedia identitas seperti [AWS IAM Identity Center](#).

## SDK for Ruby

 Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
# 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
```

- Untuk detail API, lihat [DeleteUserPolicy](#) di Referensi AWS SDK for Ruby API.

Lepaskan kebijakan dari peran

Contoh kode berikut menunjukkan cara melepaskan kebijakan IAM dari peran.

## SDK for Ruby

 Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

Modul contoh ini mencantumkan, membuat, melampirkan, dan melepaskan kebijakan peran.

```
# 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

```

- Untuk detail API, lihat [DetachRolePolicy](#) di Referensi AWS SDK for Ruby API.

## Melepaskan kebijakan dari pengguna

Contoh kode berikut menunjukkan cara melepaskan kebijakan IAM dari pengguna.

### Warning

Untuk menghindari risiko keamanan, jangan gunakan pengguna IAM untuk otentikasi saat mengembangkan perangkat lunak yang dibuat khusus atau bekerja dengan data nyata. Sebaliknya, gunakan federasi dengan penyedia identitas seperti [AWS IAM Identity Center](#).

## SDK for Ruby

### Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```

# 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
end

```

- Untuk detail API, lihat [DetachUserPolicy](#) di Referensi AWS SDK for Ruby API.

## Dapatkan kebijakan

Contoh kode berikut menunjukkan cara mendapatkan kebijakan IAM.

## SDK for Ruby

### Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```

# 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

```

- Untuk detail API, lihat [GetPolicy](#) di Referensi AWS SDK for Ruby API.



## Dapatkan peran

Contoh kode berikut menunjukkan cara mendapatkan peran IAM.

### SDK for Ruby

#### Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
# 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
```

- Untuk detail API, lihat [GetRole](#) di Referensi AWS SDK for Ruby API.

## Dapatkan pengguna

Contoh kode berikut menunjukkan cara mendapatkan pengguna IAM.

**⚠ Warning**

Untuk menghindari risiko keamanan, jangan gunakan pengguna IAM untuk otentikasi saat mengembangkan perangkat lunak yang dibuat khusus atau bekerja dengan data nyata. Sebaliknya, gunakan federasi dengan penyedia identitas seperti [AWS IAM Identity Center](#).

## SDK for Ruby

**ℹ Note**

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
# 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
```

- Untuk detail API, lihat [GetUser](#) di Referensi AWS SDK for Ruby API.

Dapatkan kebijakan kata sandi akun

Contoh kode berikut menunjukkan cara mendapatkan kebijakan kata sandi akun IAM.

## SDK for Ruby

 Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
# 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
```

- Untuk detail API, lihat [GetAccountPasswordPolicy](#) di Referensi AWS SDK for Ruby API.

## Daftar penyedia SALL

Contoh kode berikut menunjukkan bagaimana untuk daftar penyedia SALL untuk IAM.

## SDK for Ruby

 Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
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
```

- Untuk detail API, lihat [ListSAMLProviders](#) di AWS SDK for Ruby Referensi API.

## Daftar kunci akses pengguna

Contoh kode berikut menunjukkan bagaimana untuk daftar kunci akses IAM pengguna.

**⚠ Warning**

Untuk menghindari risiko keamanan, jangan gunakan pengguna IAM untuk otentikasi saat mengembangkan perangkat lunak yang dibuat khusus atau bekerja dengan data nyata. Sebaliknya, gunakan federasi dengan penyedia identitas seperti [AWS IAM Identity Center](#).

## SDK for Ruby

**ℹ Note**

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

Modul contoh ini mencantumkan, membuat, menonaktifkan, dan menghapus kunci akses.

```
# 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

```

- Untuk detail API, lihat [ListAccessKeys](#) di Referensi AWS SDK for Ruby API.

## Daftar alias akun

Contoh kode berikut menunjukkan bagaimana untuk daftar alias akun IAM.

### SDK for Ruby

#### Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

Buat daftar, buat, dan hapus alias akun.

```

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
```

- Untuk detail API, lihat [ListAccountAliases](#) di Referensi AWS SDK for Ruby API.

## Daftar grup

Contoh kode berikut menunjukkan bagaimana daftar grup IAM.



## SDK for Ruby

 Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
# 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
```

- Untuk detail API, lihat [ListGroups](#) di Referensi AWS SDK for Ruby API.

Daftar kebijakan inline untuk peran

Contoh kode berikut menunjukkan cara membuat daftar kebijakan inline untuk peran IAM.

## SDK for Ruby

 Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).


```
# 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
```

- Untuk detail API, lihat [ListRolePolicies](#) di Referensi AWS SDK for Ruby API.

## Daftar kebijakan

Contoh kode berikut menunjukkan cara membuat daftar kebijakan IAM.

## SDK for Ruby

 Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

Modul contoh ini mencantumkan, membuat, melampirkan, dan melepaskan kebijakan peran.

```
# 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
```

- Untuk detail API, lihat [ListPolicies](#) di Referensi AWS SDK for Ruby API.

## Daftar kebijakan yang dilampirkan pada peran

Contoh kode berikut menunjukkan cara membuat daftar kebijakan yang dilampirkan ke peran IAM.

### SDK for Ruby

#### Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

Modul contoh ini mencantumkan, membuat, melampirkan, dan melepaskan kebijakan peran.

```
# 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
end
```

```
# @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

```

- Untuk detail API, lihat [ListAttachedRolePolicies](#) di Referensi AWS SDK for Ruby API.

## Daftar peran

Contoh kode berikut menunjukkan bagaimana untuk daftar peran IAM.

## SDK for Ruby

### Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```

# 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
    end
  end
end

```

```

        @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
end

```

- Untuk detail API, lihat [ListRoles](#) di Referensi AWS SDK for Ruby API.

## Daftar sertifikat server

Contoh kode berikut menunjukkan bagaimana untuk daftar sertifikat server IAM.

### SDK for Ruby

#### Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

Daftar, perbarui, dan hapus sertifikat server.

```

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
```

- Untuk detail API, lihat [ListServerCertificates](#) di Referensi AWS SDK for Ruby API.

## Daftar pengguna

Contoh kode berikut menunjukkan bagaimana untuk daftar pengguna IAM.

### Warning

Untuk menghindari risiko keamanan, jangan gunakan pengguna IAM untuk otentikasi saat mengembangkan perangkat lunak yang dibuat khusus atau bekerja dengan data nyata. Sebaliknya, gunakan federasi dengan penyedia identitas seperti [AWS IAM Identity Center](#).

## SDK for Ruby

### Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
# 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
end
```

```
    end
  end
  users
rescue Aws::IAM::Errors::ServiceError => e
  @logger.error("Error listing users: #{e.message}")
  []
end
```

- Untuk detail API, lihat [ListUsers](#) di Referensi AWS SDK for Ruby API.

## Memperbarui sertifikat server

Contoh kode berikut menunjukkan cara memperbarui sertifikat server IAM.

### SDK for Ruby

#### Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

Daftar, perbarui, dan hapus sertifikat server.

```
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
```

- Untuk detail API, lihat [UpdateServerCertificate](#) di Referensi AWS SDK for Ruby API.

## Memperbarui pengguna

Contoh kode berikut menunjukkan cara memperbarui pengguna IAM.

### Warning

Untuk menghindari risiko keamanan, jangan gunakan pengguna IAM untuk otentikasi saat mengembangkan perangkat lunak yang dibuat khusus atau bekerja dengan data nyata. Sebaliknya, gunakan federasi dengan penyedia identitas seperti [AWS IAM Identity Center](#).

## SDK for Ruby

### Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
# 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
```

- Untuk detail API, lihat [UpdateUser](#) di Referensi AWS SDK for Ruby API.

## Skenario

Buat pengguna dan ambil peran

Contoh kode berikut menunjukkan cara membuat pengguna dan mengambil peran.

### Warning

Untuk menghindari risiko keamanan, jangan gunakan pengguna IAM untuk otentikasi saat mengembangkan perangkat lunak yang dibuat khusus atau bekerja dengan data nyata. Sebaliknya, gunakan federasi dengan penyedia identitas seperti [AWS IAM Identity Center](#).

- Buat pengguna tanpa izin.
- Buat peran yang memberikan izin untuk mencantumkan bucket Amazon S3 untuk akun tersebut.
- Tambahkan kebijakan agar pengguna dapat mengambil peran tersebut.
- Asumsikan peran dan daftar bucket S3 menggunakan kredensial sementara, lalu bersihkan sumber daya.

## SDK for Ruby

### Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

Buat pengguna IAM dan peran yang memberikan izin untuk mencantumkan bucket Amazon S3. Pengguna hanya memiliki hak untuk mengambil peran. Setelah mengasumsikan peran, gunakan kredensial sementara untuk membuat daftar bucket untuk akun.

```
# 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
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
    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

# 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__
```

- Lihat detail API di topik-topik berikut dalam Referensi API AWS SDK for Ruby .
  - [AttachRolePolicy](#)
  - [CreateAccessKey](#)
  - [CreatePolicy](#)
  - [CreateRole](#)
  - [CreateUser](#)
  - [DeleteAccessKey](#)
  - [DeletePolicy](#)
  - [DeleteRole](#)
  - [DeleteUser](#)
  - [DeleteUserPolicy](#)
  - [DetachRolePolicy](#)
  - [PutUserPolicy](#)

## Contoh Kinesis menggunakan SDK for Ruby

Contoh kode berikut menunjukkan cara melakukan tindakan dan mengimplementasikan skenario umum dengan menggunakan AWS SDK for Ruby Kinesis with.

Tindakan merupakan kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Meskipun tindakan menunjukkan cara memanggil setiap fungsi layanan, Anda dapat melihat tindakan dalam konteks pada skenario yang terkait dan contoh lintas layanan.

Skenario adalah contoh kode yang menunjukkan cara untuk menyelesaikan tugas tertentu dengan memanggil beberapa fungsi dalam layanan yang sama.

Setiap contoh menyertakan tautan ke GitHub, di mana Anda dapat menemukan petunjuk tentang cara mengatur dan menjalankan kode dalam konteks.

## Topik

- [Contoh nirserver](#)

## Contoh nirserver

Memanggil fungsi Lambda dari pemicu Kinesis

Contoh kode berikut menunjukkan bagaimana menerapkan fungsi Lambda yang menerima peristiwa yang dipicu oleh menerima catatan dari aliran Kinesis. Fungsi mengambil payload Kinesis, mendekode dari Base64, dan mencatat konten rekaman.

## SDK for Ruby

### Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di repositori [contoh Nirserver](#).

Mengonsumsi acara Kinesis dengan Lambda menggunakan Ruby.

```
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
```

## Melaporkan kegagalan item batch untuk fungsi Lambda dengan pemicu Kinesis

Contoh kode berikut menunjukkan bagaimana menerapkan respons batch paralel untuk fungsi Lambda yang menerima peristiwa dari aliran Kinesis. Fungsi melaporkan kegagalan item batch dalam respons, memberi sinyal ke Lambda untuk mencoba lagi pesan tersebut nanti.

### SDK for Ruby

#### Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di repositori [contoh Nirserver](#).

## Melaporkan kegagalan item batch Kinesis dengan Lambda menggunakan Ruby.

```
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.
    end
  end
end
```

```
        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
end
```

## AWS KMS contoh menggunakan SDK for Ruby

Contoh kode berikut menunjukkan cara melakukan tindakan dan menerapkan skenario umum dengan menggunakan AWS SDK for Ruby with AWS KMS.

Tindakan merupakan kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Meskipun tindakan menunjukkan cara memanggil setiap fungsi layanan, Anda dapat melihat tindakan dalam konteks pada skenario yang terkait dan contoh lintas layanan.

Skenario adalah contoh kode yang menunjukkan cara untuk menyelesaikan tugas tertentu dengan memanggil beberapa fungsi dalam layanan yang sama.

Setiap contoh menyertakan tautan ke GitHub, di mana Anda dapat menemukan petunjuk tentang cara mengatur dan menjalankan kode dalam konteks.

Topik

- [Tindakan](#)

Tindakan

Buat kunci

Contoh kode berikut menunjukkan cara membuat AWS KMS key.



## SDK for Ruby

 Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
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
```

- Untuk detail API, lihat [CreateKey](#) di Referensi AWS SDK for Ruby API.

## Dekripsi ciphertext

Contoh kode berikut menunjukkan cara mendekripsi ciphertext yang dienkrpsi oleh kunci KMS.

## SDK for Ruby

 Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
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
```

- Untuk detail API, lihat [Mendekripsi](#) di Referensi AWS SDK for Ruby API.

Enkripsi teks menggunakan kunci

Contoh kode berikut menunjukkan cara mengenkripsi teks menggunakan kunci KMS.

## SDK for Ruby

 Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
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*")
```

- Untuk detail API, lihat [Enkripsi](#) di Referensi AWS SDK for Ruby API.

Enkripsi ulang ciphertext dari satu kunci ke kunci lainnya

Contoh kode berikut menunjukkan cara mengenkripsi ulang ciphertext dari satu kunci KMS ke yang lain.

SDK for Ruby

 Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
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*")
```

- Untuk detail API, lihat [ReEncrypt](#) di Referensi AWS SDK for Ruby API.

## Contoh Lambda menggunakan SDK for Ruby

Contoh kode berikut menunjukkan cara melakukan tindakan dan mengimplementasikan skenario umum dengan menggunakan Lambda AWS SDK for Ruby with.

Tindakan merupakan kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Meskipun tindakan menunjukkan cara memanggil setiap fungsi layanan, Anda dapat melihat tindakan dalam konteks pada skenario yang terkait dan contoh lintas layanan.

Skenario adalah contoh kode yang menunjukkan cara untuk menyelesaikan tugas tertentu dengan memanggil beberapa fungsi dalam layanan yang sama.

Setiap contoh menyertakan tautan ke GitHub, di mana Anda dapat menemukan petunjuk tentang cara mengatur dan menjalankan kode dalam konteks.

### Topik

- [Tindakan](#)
- [Skenario](#)
- [Contoh nirserver](#)

## Tindakan

Buat fungsi

Contoh kode berikut menunjukkan cara membuat fungsi Lambda.

SDK for Ruby

### Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
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})
  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
```

- Untuk detail API, lihat [CreateFunction](#) di Referensi AWS SDK for Ruby API.

## Hapus fungsi

Contoh kode berikut menunjukkan cara menghapus fungsi Lambda.

## SDK for Ruby

### Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
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
```

- Untuk detail API, lihat [DeleteFunction](#) di Referensi AWS SDK for Ruby API.

Dapatkan fungsi

Contoh kode berikut menunjukkan cara mendapatkan fungsi Lambda.

SDK for Ruby

#### Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
class LambdaWrapper
  attr_accessor :lambda_client

  def initialize
    @lambda_client = Aws::Lambda::Client.new
    @logger = Logger.new($stdout)
    @logger.level = Logger::WARN
  end
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
```

- Untuk detail API, lihat [GetFunction](#) di Referensi AWS SDK for Ruby API.

## Memanggil fungsi

Contoh kode berikut menunjukkan cara memanggil fungsi Lambda.

### SDK for Ruby

#### Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
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
```

- Untuk detail API, lihat [Memanggil di Referensi AWS SDK for Ruby API](#).

## Daftar fungsi

Contoh kode berikut menunjukkan bagaimana untuk daftar fungsi Lambda.

## SDK for Ruby

### Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
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
  end
end
```

```

    end
  end
  functions
rescue Aws::Lambda::Errors::ServiceException => e
  @logger.error("There was an error executing #{function_name}:\n #{e.message}")
end

```

- Untuk detail API, lihat [ListFunctions](#) di Referensi AWS SDK for Ruby API.

## Perbarui kode fungsi

Contoh kode berikut menunjukkan cara memperbarui kode fungsi Lambda.

## SDK for Ruby

### Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```

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

```

- Untuk detail API, lihat [UpdateFunctionCode](#) di Referensi AWS SDK for Ruby API.

## Perbarui konfigurasi fungsi

Contoh kode berikut menunjukkan cara memperbarui konfigurasi fungsi Lambda.

### SDK for Ruby

#### Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```

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})
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

```

- Untuk detail API, lihat [UpdateFunctionConfiguration](#) di Referensi AWS SDK for Ruby API.

## Skenario


Memulai dengan fungsi

Contoh kode berikut ini menunjukkan cara untuk melakukan:

- Buat peran IAM dan fungsi Lambda, lalu unggah kode handler.
- Panggil fungsi dengan satu parameter dan dapatkan hasil.
- Perbarui kode fungsi dan konfigurasikan dengan variabel lingkungan.
- Panggil fungsi dengan parameter baru dan dapatkan hasil. Tampilkan log eksekusi yang dikembalikan.
- Buat daftar fungsi untuk akun Anda, lalu bersihkan sumber daya.

Untuk informasi selengkapnya, lihat [Membuat fungsi Lambda dengan konsol](#).

## SDK for Ruby

 Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

Siapkan izin IAM prasyarat untuk fungsi Lambda yang mampu menulis log.

```
# 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
    end
  end
end
```

```

        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
end

```

Tentukan handler Lambda yang menambah angka yang disediakan sebagai parameter pemanggilan.

```

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
```

Zip fungsi Lambda Anda ke dalam paket penerapan.

```
# 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
```

## Buat fungsi Lambda baru.

```

# 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})
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

```

## Panggil fungsi Lambda Anda dengan parameter runtime opsional.

```

# 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

```

Perbarui konfigurasi fungsi Lambda Anda untuk menyuntikkan variabel lingkungan baru.

```

# 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})
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

```

Perbarui kode fungsi Lambda Anda dengan paket penerapan berbeda yang berisi kode berbeda.

```

# 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

```

Daftar semua fungsi Lambda yang ada menggunakan paginator bawaan.

```

# 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

```

Hapus fungsi Lambda tertentu.

```
# 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
```

- Lihat detail API di topik-topik berikut dalam Referensi API AWS SDK for Ruby .
  - [CreateFunction](#)
  - [DeleteFunction](#)
  - [GetFunction](#)
  - [Memohon](#)
  - [ListFunctions](#)
  - [UpdateFunctionCode](#)
  - [UpdateFunctionConfiguration](#)

## Contoh nirserver

### Memanggil fungsi Lambda dari pemicu Kinesis

Contoh kode berikut menunjukkan bagaimana menerapkan fungsi Lambda yang menerima peristiwa yang dipicu oleh menerima catatan dari aliran Kinesis. Fungsi mengambil payload Kinesis, mendekode dari Base64, dan mencatat konten rekaman.

### SDK for Ruby

#### Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di repositori [contoh Nirserver](#).

## Mengkonsumsi acara Kinesis dengan Lambda menggunakan Ruby.

```
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
```

## Memanggil fungsi Lambda dari pemicu Amazon SNS

Contoh kode berikut menunjukkan cara menerapkan fungsi Lambda yang menerima peristiwa yang dipicu dengan menerima pesan dari topik SNS. Fungsi mengambil pesan dari parameter peristiwa dan mencatat konten setiap pesan.

### SDK for Ruby

#### Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di repositori [contoh Nirserver](#).

## Mengkonsumsi acara SNS dengan Lambda menggunakan Ruby.

```
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
```

## Memanggil fungsi Lambda dari pemicu Amazon SQS

Contoh kode berikut menunjukkan bagaimana menerapkan fungsi Lambda yang menerima peristiwa yang dipicu oleh menerima pesan dari antrian SQS. Fungsi mengambil pesan dari parameter peristiwa dan mencatat konten setiap pesan.

### SDK for Ruby

#### Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di repositori [contoh Nirserver](#).

## Mengkonsumsi acara SQS dengan Lambda menggunakan Ruby.

```
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']}"
  end
end
```

```
# TODO: Do interesting work based on the new message
rescue StandardError => err
  puts "An error occurred"
  raise err
end
end
```

## Melaporkan kegagalan item batch untuk fungsi Lambda dengan pemicu Kinesis

Contoh kode berikut menunjukkan bagaimana menerapkan respons batch paralel untuk fungsi Lambda yang menerima peristiwa dari aliran Kinesis. Fungsi melaporkan kegagalan item batch dalam respons, memberi sinyal ke Lambda untuk mencoba lagi pesan tersebut nanti.

### SDK for Ruby

#### Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di repositori [contoh Nirserver](#).

## Melaporkan kegagalan item batch Kinesis dengan Lambda menggunakan Ruby.

```
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.
    end
  end
end
```

```

    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
end

```

## Melaporkan kegagalan item batch untuk fungsi Lambda dengan pemicu Amazon SQS

Contoh kode berikut menunjukkan cara mengimplementasikan respons batch sebagian untuk fungsi Lambda yang menerima peristiwa dari antrian SQS. Fungsi melaporkan kegagalan item batch dalam respons, memberi sinyal ke Lambda untuk mencoba lagi pesan tersebut nanti.

### SDK for Ruby

#### Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di repositori [contoh Nirserver](#).

## Melaporkan kegagalan item batch SQS dengan Lambda menggunakan Ruby.

```

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
```

## Contoh Amazon Polly menggunakan SDK for Ruby

Contoh kode berikut menunjukkan cara melakukan tindakan dan menerapkan skenario umum dengan menggunakan Amazon Polly. AWS SDK for Ruby

Tindakan merupakan kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Meskipun tindakan menunjukkan cara memanggil setiap fungsi layanan, Anda dapat melihat tindakan dalam konteks pada skenario yang terkait dan contoh lintas layanan.

Skenario adalah contoh kode yang menunjukkan cara untuk menyelesaikan tugas tertentu dengan memanggil beberapa fungsi dalam layanan yang sama.

Setiap contoh menyertakan tautan ke GitHub, di mana Anda dapat menemukan petunjuk tentang cara mengatur dan menjalankan kode dalam konteks.

Topik

- [Tindakan](#)

## Tindakan

Dapatkan suara yang tersedia untuk sintesis

Contoh kode berikut menunjukkan cara mendapatkan suara Amazon Polly yang tersedia untuk sintesis.



## SDK for Ruby

 Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
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
```

- Untuk detail API, lihat [DescribeVoices](#) di Referensi AWS SDK for Ruby API.

## Daftar leksikon pengucapan

Contoh kode berikut menunjukkan cara mencantumkan leksikon pengucapan Amazon Polly.

## SDK for Ruby

 Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
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
```

- Untuk detail API, lihat [ListLexicons](#) di Referensi AWS SDK for Ruby API.

## Sintesis ucapan dari teks

Contoh kode berikut menunjukkan cara mensintesis ucapan dari teks dengan Amazon Polly.

## SDK for Ruby

 Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
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
```

- Untuk detail API, lihat [SynthesizeSpeech](#) di Referensi AWS SDK for Ruby API.

## Contoh Amazon RDS menggunakan SDK for Ruby

Contoh kode berikut menunjukkan cara melakukan tindakan dan menerapkan skenario umum dengan menggunakan RDS AWS SDK for Ruby with Amazon.

Tindakan merupakan kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Meskipun tindakan menunjukkan cara memanggil setiap fungsi layanan, Anda dapat melihat tindakan dalam konteks pada skenario yang terkait dan contoh lintas layanan.

Skenario adalah contoh kode yang menunjukkan cara untuk menyelesaikan tugas tertentu dengan memanggil beberapa fungsi dalam layanan yang sama.

Setiap contoh menyertakan tautan ke GitHub, di mana Anda dapat menemukan petunjuk tentang cara mengatur dan menjalankan kode dalam konteks.

Topik

- [Tindakan](#)

## Tindakan

Buat cuplikan instans basis data

Contoh kode berikut menunjukkan cara membuat snapshot dari instans Amazon RDS DB.

SDK for Ruby

### Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
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
```

- Lihat detail API di [CreateDBSnapshot](#) dalam Referensi API AWS SDK for Ruby .

Jelaskan instans basis data

Contoh kode berikut menunjukkan cara mendeskripsikan instans Amazon RDS DB.

## SDK for Ruby

 Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
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
```

- Lihat detail API di [DescribeDBInstances](#) dalam Referensi API AWS SDK for Ruby .

Jelaskan grup parameter basis data

Contoh kode berikut menunjukkan cara mendeskripsikan kelompok parameter Amazon RDS DB.

## SDK for Ruby

 Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```

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

```

- Untuk detail API, lihat [DescribeDB ParameterGroups](#) di Referensi AWS SDK for Ruby API.

Jelaskan parameter dalam grup parameter basis data

Contoh kode berikut menunjukkan cara mendeskripsikan parameter dalam grup parameter Amazon RDS DB.

SDK for Ruby

#### Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```

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

```

- Lihat detail API di [DescribeDBParameters](#) dalam Referensi API AWS SDK for Ruby .

Jelaskan cuplikan dari instans basis data

Contoh kode berikut menunjukkan cara mendeskripsikan snapshot dari instans Amazon RDS DB.

SDK for Ruby

#### Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```

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
end

```



```
        })
    end
    instance_snapshots
  rescue Aws::Errors::ServiceError => e
    puts "Couldn't list instance snapshots:\n #{e.message}"
  end
end
```

- Lihat detail API di [DescribeDBSnapshots](#) dalam Referensi API AWS SDK for Ruby .

## Contoh Amazon S3 menggunakan SDK for Ruby

Contoh kode berikut menunjukkan cara melakukan tindakan dan menerapkan skenario umum dengan menggunakan Amazon S3. AWS SDK for Ruby

Tindakan merupakan kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Meskipun tindakan menunjukkan cara memanggil setiap fungsi layanan, Anda dapat melihat tindakan dalam konteks pada skenario yang terkait dan contoh lintas layanan.

Skenario adalah contoh kode yang menunjukkan cara untuk menyelesaikan tugas tertentu dengan memanggil beberapa fungsi dalam layanan yang sama.

Setiap contoh menyertakan tautan ke GitHub, di mana Anda dapat menemukan petunjuk tentang cara mengatur dan menjalankan kode dalam konteks.

Topik


- [Tindakan](#)
- [Skenario](#)

### Tindakan

Tambahkan aturan CORS ke bucket

Contoh kode berikut menunjukkan cara menambahkan aturan cross-origin resource sharing (CORS) ke bucket S3.

## SDK for Ruby

 Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
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}"
  end
end
```

```

    false
  end

end

```

- Untuk detail API, lihat [PutBucketCors](#) di Referensi AWS SDK for Ruby API.

## Menambahkan kebijakan ke bucket

Contoh kode berikut menunjukkan cara menambahkan kebijakan ke bucket S3.

### SDK for Ruby

#### Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```

# 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

  # Sets a policy on a bucket.
  #
  def set_policy(policy)
    @bucket_policy.put(policy: policy)
    true
  rescue Aws::Errors::ServiceError => e
    puts "Couldn't set the policy for #{@bucket_policy.bucket.name}. Here's why:
#{e.message}"
    false
  end
end

```

- Untuk detail API, lihat [PutBucketPolicy](#) di Referensi AWS SDK for Ruby API.

Menyalin objek dari satu bucket ke bucket lainnya

Contoh kode berikut menunjukkan cara menyalin objek S3 dari satu ember ke ember lainnya.

SDK for Ruby

#### Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

Salin objek.

```
require "aws-sdk-s3"

# Wraps Amazon S3 object actions.
class ObjectCopyWrapper
  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 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)
  end
end
```

```

    rescue Aws::Errors::ServiceError => e
      puts "Couldn't copy #{@source_object.key} to #{target_object_key}. Here's why:
#{e.message}"
    end
  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"

  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__

```

Salin objek dan tambahkan enkripsi sisi server ke objek tujuan.

```

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)
  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__

```

- Untuk detail API, lihat [CopyObject](#) di Referensi AWS SDK for Ruby API.

## Buat bucket

Contoh kode berikut menunjukkan cara membuat bucket S3.

### SDK for Ruby

#### Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
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!"
    end
  end
end
```

```

    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__

```

- Untuk detail API, lihat [CreateBucket](#) di Referensi AWS SDK for Ruby API.

## Menghapus aturan CORS dari bucket

Contoh kode berikut menunjukkan cara menghapus aturan CORS dari bucket S3.

### SDK for Ruby

#### Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```

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

```

- Untuk detail API, lihat [DeleteBucketCors](#) di Referensi AWS SDK for Ruby API.

## Menghapus kebijakan dari bucket

Contoh kode berikut menunjukkan cara menghapus kebijakan dari bucket S3.

### SDK for Ruby

#### Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```

# 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

```

- Untuk detail API, lihat [DeleteBucketPolicy](#) di Referensi AWS SDK for Ruby API.

## Menghapus bucket kosong

Contoh kode berikut menunjukkan cara menghapus bucket S3 kosong.

### SDK for Ruby

#### Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```

# 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

```

```

rescue Aws::Errors::ServiceError => e
  puts("Couldn't empty and delete bucket #{bucket.name}.")
  puts("\t#{e.code}: #{e.message}")
  raise
end

```

- Untuk detail API, lihat [DeleteBucket](#) di Referensi AWS SDK for Ruby API.

## Menghapus beberapa objek

Contoh kode berikut menunjukkan cara menghapus beberapa objek dari bucket S3.

### SDK for Ruby

#### Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```

# 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
rescue Aws::Errors::ServiceError => e
  puts("Couldn't empty and delete bucket #{bucket.name}.")
  puts("\t#{e.code}: #{e.message}")
  raise
end

```

- Untuk detail API, lihat [DeleteObjects](#) di Referensi AWS SDK for Ruby API.

## Menentukan keberadaan dan jenis konten suatu objek

Contoh kode berikut menunjukkan cara menentukan keberadaan dan jenis konten objek dalam bucket S3.

### SDK for Ruby

#### Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
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

# 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__

```

- Untuk detail API, lihat [HeadObject](#) di Referensi AWS SDK for Ruby API.

## Mendapatkan aturan CORS untuk bucket

Contoh kode berikut menunjukkan cara mendapatkan aturan cross-origin resource sharing (CORS) untuk bucket S3.

### SDK for Ruby

#### Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```

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
  end
end

```

```
rescue Aws::Errors::ServiceError => e
  puts "Couldn't get CORS configuration for #{@bucket_cors.bucket.name}. Here's
why: #{e.message}"
  nil
end


end
```

- Untuk detail API, lihat [GetBucketCors](#) di Referensi AWS SDK for Ruby API.

Mendapatkan objek dari bucket

Contoh kode berikut menunjukkan cara membaca data dari objek dalam bucket S3.

SDK for Ruby

 Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

Dapatkan objek.

```
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__

```

Dapatkan objek dan laporkan status enkripsi sisi servernya.

```

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__

```

- Untuk detail API, lihat [GetObject](#) di Referensi AWS SDK for Ruby API.

Dapatkan kebijakan untuk bucket

Contoh kode berikut menunjukkan cara mendapatkan kebijakan untuk bucket S3.

SDK for Ruby

#### Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```

# 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

```

- Untuk detail API, lihat [GetBucketPolicy](#) di Referensi AWS SDK for Ruby API.

## Buat daftar bucket

Contoh kode berikut menunjukkan cara membuat daftar bucket S3.

## SDK for Ruby

### Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```

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__
```

- Untuk detail API, lihat [ListBuckets](#) di Referensi AWS SDK for Ruby API.

## Membuat daftar objek dalam bucket

Contoh kode berikut menunjukkan cara membuat daftar objek dalam bucket S3.

### SDK for Ruby

#### Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
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__
```

- Untuk detail API, lihat [ListObjectsV2](#) di Referensi AWS SDK for Ruby API.

## Menetapkan konfigurasi situs web untuk bucket

Contoh kode berikut menunjukkan cara mengatur konfigurasi situs web untuk bucket S3.

### SDK for Ruby

#### Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
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
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__
```

- Untuk detail API, lihat [PutBucketWebsite](#) di Referensi AWS SDK for Ruby API.

## Mengunggah objek ke bucket

Contoh kode berikut menunjukkan cara mengunggah objek ke bucket S3.

## SDK for Ruby

### Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

Unggah file menggunakan pengunggah terkelola (`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

  # 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__

```

## Unggah file menggunakan Object.put.

```

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

  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__

```

Unggah file menggunakan `Object.put` dan tambahkan enkripsi sisi server.

```

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
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__
```

- Untuk detail API, lihat [PutObject](#) di Referensi AWS SDK for Ruby API.

## Skenario

Membuat URL yang telah ditetapkan sebelumnya

Contoh kode berikut menunjukkan cara membuat URL presigned untuk Amazon S3 dan mengunggah objek.



## SDK for Ruby

 Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
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
end

run_demo if $PROGRAM_NAME == __FILE__
```

## Memulai bucket dan objek

Contoh kode berikut ini menunjukkan cara untuk melakukan:

- Membuat bucket dan mengunggah file ke dalamnya.
- Mengunduh objek dari bucket.
- Menyalin objek ke subfolder di bucket.
- Membuat daftar objek dalam bucket.
- Menghapus objek bucket dan bucket tersebut.

## SDK for Ruby

### Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
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}.")
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
```

```
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

# 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!")
end
```

```
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__
```

- Lihat detail API di topik-topik berikut dalam Referensi API AWS SDK for Ruby .
  - [CopyObject](#)
  - [CreateBucket](#)
  - [DeleteBucket](#)
  - [DeleteObjects](#)
  - [GetObject](#)
  - [ListObjectsV2](#)
  - [PutObject](#)

## Contoh Amazon SES menggunakan SDK for Ruby

Contoh kode berikut menunjukkan cara melakukan tindakan dan menerapkan skenario umum AWS SDK for Ruby dengan menggunakan Amazon SES.

Tindakan merupakan kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Meskipun tindakan menunjukkan cara memanggil setiap fungsi layanan, Anda dapat melihat tindakan dalam konteks pada skenario yang terkait dan contoh lintas layanan.

Skenario adalah contoh kode yang menunjukkan cara untuk menyelesaikan tugas tertentu dengan memanggil beberapa fungsi dalam layanan yang sama.

Setiap contoh menyertakan tautan ke GitHub, di mana Anda dapat menemukan petunjuk tentang cara mengatur dan menjalankan kode dalam konteks.

### Topik

- [Tindakan](#)

## Tindakan

Dapatkan status identitas

Contoh kode berikut menunjukkan cara mendapatkan status identitas Amazon SES.

SDK for Ruby

### Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
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
```

- Untuk detail API, lihat [GetIdentityVerificationAttributes](#) di Referensi AWS SDK for Ruby API.

## Daftar identitas

Contoh kode berikut menunjukkan cara mencantumkan identitas Amazon SES.

### SDK for Ruby

#### Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
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
```

- Untuk detail API, lihat [ListIdentities](#) di Referensi AWS SDK for Ruby API.



## Kirim email

Contoh kode berikut menunjukkan cara mengirim email dengan Amazon SES.

### SDK for Ruby

#### Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
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
```

- Untuk detail API, lihat [SendEmail](#) di Referensi AWS SDK for Ruby API.

## Verifikasi identitas email

Contoh kode berikut menunjukkan cara memverifikasi identitas email dengan Amazon SES.

### SDK for Ruby

#### Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
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
```

- Untuk detail API, lihat [VerifyEmailIdentity](#) di Referensi AWS SDK for Ruby API.

## Contoh Amazon SES API v2 menggunakan SDK for Ruby

Contoh kode berikut menunjukkan cara melakukan tindakan dan menerapkan skenario umum dengan menggunakan AWS SDK for Ruby with Amazon SES API v2.

Tindakan merupakan kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Meskipun tindakan menunjukkan cara memanggil setiap fungsi layanan, Anda dapat melihat tindakan dalam konteks pada skenario yang terkait dan contoh lintas layanan.

Skenario adalah contoh kode yang menunjukkan cara untuk menyelesaikan tugas tertentu dengan memanggil beberapa fungsi dalam layanan yang sama.

Setiap contoh menyertakan tautan ke GitHub, di mana Anda dapat menemukan petunjuk tentang cara mengatur dan menjalankan kode dalam konteks.

Topik

- [Tindakan](#)

### Tindakan

Kirim email

Contoh kode berikut menunjukkan cara mengirim email Amazon SES API v2.

SDK for Ruby

#### Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
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)
```

- Untuk detail API, lihat [SendEmail](#) di Referensi AWS SDK for Ruby API.

## Contoh Amazon SNS menggunakan SDK for Ruby

Contoh kode berikut menunjukkan cara melakukan tindakan dan menerapkan skenario umum dengan menggunakan AWS SDK for Ruby dengan Amazon SNS.

Tindakan merupakan kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Meskipun tindakan menunjukkan cara memanggil setiap fungsi layanan, Anda dapat melihat tindakan dalam konteks pada skenario yang terkait dan contoh lintas layanan.

Skenario adalah contoh kode yang menunjukkan cara untuk menyelesaikan tugas tertentu dengan memanggil beberapa fungsi dalam layanan yang sama.

Setiap contoh menyertakan tautan ke GitHub, di mana Anda dapat menemukan petunjuk tentang cara mengatur dan menjalankan kode dalam konteks.

## Topik

- [Tindakan](#)
- [Contoh nirserver](#)

## Tindakan

Buat topik

Contoh kode berikut menunjukkan cara membuat topik Amazon SNS.

SDK for Ruby

### Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
# 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

  # 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
```

```

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

```

- Untuk informasi selengkapnya, silakan lihat [Panduan Developer AWS SDK for Ruby](#).
- Untuk detail API, lihat [CreateTopic](#) di Referensi AWS SDK for Ruby API.

Buat daftar pelanggan suatu topik

Contoh kode berikut menunjukkan cara mengambil daftar pelanggan topik Amazon SNS.

SDK for Ruby

#### Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```

# 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
```

- Untuk informasi selengkapnya, silakan lihat [Panduan Developer AWS SDK for Ruby](#).
- Untuk detail API, lihat [ListSubscriptions](#) di Referensi AWS SDK for Ruby API.

## Daftar topik

Contoh kode berikut menunjukkan cara membuat daftar topik Amazon SNS.

## SDK for Ruby

### Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).



```
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__
```

- Untuk informasi selengkapnya, silakan lihat [Panduan Developer AWS SDK for Ruby](#).
- Untuk detail API, lihat [ListTopics](#) di Referensi AWS SDK for Ruby API.

Publikasikan ke topik

Contoh kode berikut menunjukkan cara mempublikasikan pesan ke topik Amazon SNS.

## SDK for Ruby

 Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
# 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.")
end
```

```

unless message_sender.send_message(topic_arn, message)
  @logger.error("Message sending failed. Stopping program.")
  exit 1
end
end
end

```

- Untuk informasi selengkapnya, silakan lihat [Panduan Developer AWS SDK for Ruby](#).
- Untuk detail API, lihat [Publikasikan](#) di Referensi AWS SDK for Ruby API.

Tetapkan atribut topik

Contoh kode berikut menunjukkan cara mengatur atribut topik Amazon SNS.

SDK for Ruby

#### Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```

# 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)
  end
end

```

```

    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)
end

```


```
enabler.enable_resource(topic_arn, resource_arn, policy_name)
end
```

- Untuk informasi selengkapnya, silakan lihat [Panduan Developer AWS SDK for Ruby](#).
- Untuk detail API, lihat [SetTopicAttributes](#) di Referensi AWS SDK for Ruby API.

Berlangganan alamat email ke suatu topik

Contoh kode berikut menunjukkan cara berlangganan alamat email ke topik Amazon SNS.

SDK for Ruby

 Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
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

  # 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
```

- Untuk informasi selengkapnya, silakan lihat [Panduan Developer AWS SDK for Ruby](#).
- Untuk detail API, lihat [Berlangganan](#) di Referensi AWS SDK for Ruby API.

## Contoh nirserver

### Memanggil fungsi Lambda dari pemicu Amazon SNS

Contoh kode berikut menunjukkan cara menerapkan fungsi Lambda yang menerima peristiwa yang dipicu dengan menerima pesan dari topik SNS. Fungsi mengambil pesan dari parameter peristiwa dan mencatat konten setiap pesan.

## SDK for Ruby

### Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di repositori [contoh Nirserver](#).

Mengonsumsi acara SNS dengan Lambda menggunakan Ruby.

```
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
```

## Contoh Amazon SQS menggunakan SDK for Ruby

Contoh kode berikut menunjukkan cara melakukan tindakan dan menerapkan skenario umum dengan menggunakan Amazon SQS. AWS SDK for Ruby

Tindakan merupakan kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Meskipun tindakan menunjukkan cara memanggil setiap fungsi layanan, Anda dapat melihat tindakan dalam konteks pada skenario yang terkait dan contoh lintas layanan.

Skenario adalah contoh kode yang menunjukkan cara untuk menyelesaikan tugas tertentu dengan memanggil beberapa fungsi dalam layanan yang sama.

Setiap contoh menyertakan tautan ke GitHub, di mana Anda dapat menemukan petunjuk tentang cara mengatur dan menjalankan kode dalam konteks.

### Topik

- [Tindakan](#)
- [Contoh nirserver](#)

## Tindakan

Ubah visibilitas batas waktu pesan

Contoh kode berikut menunjukkan cara mengubah visibilitas batas waktu pesan Amazon SQS.

SDK for Ruby

### Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
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

```

- Untuk detail API, lihat [ChangeMessageVisibility](#) di Referensi AWS SDK for Ruby API.

## Membuat antrian

Contoh kode berikut menunjukkan cara membuat antrian Amazon SQS.

### SDK for Ruby

#### Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```

# 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__
```

- Untuk detail API, lihat [CreateQueue](#) di Referensi AWS SDK for Ruby API.

## Hapus antrian

Contoh kode berikut menunjukkan cara menghapus antrian Amazon SQS.

## SDK for Ruby

### Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
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)
```

- Untuk detail API, lihat [DeleteQueue](#) di Referensi AWS SDK for Ruby API.

## Daftar antrian

Contoh kode berikut menunjukkan cara membuat daftar antrian Amazon SQS.

## SDK for Ruby

### Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
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
```

- Untuk detail API, lihat [ListQueues](#) di Referensi AWS SDK for Ruby API.

Menerima pesan dari antrian

Contoh kode berikut menunjukkan cara menerima pesan dari antrian Amazon SQS.

SDK for Ruby

**Note**

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
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__
```

- Untuk detail API, lihat [ReceiveMessage](#) di Referensi AWS SDK for Ruby API.

## Mengirim sekumpulan pesan ke antrian

Contoh kode berikut menunjukkan cara mengirim sekumpulan pesan ke antrian Amazon SQS.

### SDK for Ruby

#### Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
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
end
```

```
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
```

- Untuk detail API, lihat [SendMessageBatch](#) di Referensi AWS SDK for Ruby API.



## Mengirim Pesan Antrian

Contoh kode berikut menunjukkan cara mengirim pesan ke antrian Amazon SQS.

### SDK for Ruby

#### Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
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__
```

- Untuk detail API, lihat [SendMessage](#) di Referensi AWS SDK for Ruby API.

## Contoh nirserver

### Memanggil fungsi Lambda dari pemicu Amazon SQS

Contoh kode berikut menunjukkan bagaimana menerapkan fungsi Lambda yang menerima peristiwa yang dipicu oleh menerima pesan dari antrian SQS. Fungsi mengambil pesan dari parameter peristiwa dan mencatat konten setiap pesan.

### SDK for Ruby

#### Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di repositori [contoh Nirserver](#).

Mengonsumsi acara SQS dengan Lambda menggunakan Ruby.

```
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
```

## Melaporkan kegagalan item batch untuk fungsi Lambda dengan pemicu Amazon SQS

Contoh kode berikut menunjukkan cara mengimplementasikan respons batch sebagian untuk fungsi Lambda yang menerima peristiwa dari antrian SQS. Fungsi melaporkan kegagalan item batch dalam respons, memberi sinyal ke Lambda untuk mencoba lagi pesan tersebut nanti.

### SDK for Ruby

#### Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di repositori [contoh Nirserver](#).

## Melaporkan kegagalan item batch SQS dengan Lambda menggunakan Ruby.

```
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
```

## AWS STS contoh menggunakan SDK for Ruby

Contoh kode berikut menunjukkan cara melakukan tindakan dan menerapkan skenario umum dengan menggunakan AWS SDK for Ruby with AWS STS.

Tindakan merupakan kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Meskipun tindakan menunjukkan cara memanggil setiap fungsi layanan, Anda dapat melihat tindakan dalam konteks pada skenario yang terkait dan contoh lintas layanan.

Skenario adalah contoh kode yang menunjukkan cara untuk menyelesaikan tugas tertentu dengan memanggil beberapa fungsi dalam layanan yang sama.

Setiap contoh menyertakan tautan ke GitHub, di mana Anda dapat menemukan petunjuk tentang cara mengatur dan menjalankan kode dalam konteks.

Topik


- [Tindakan](#)

### Tindakan

Asumsikan peran

Contoh kode berikut menunjukkan bagaimana untuk mengambil peran dengan AWS STS.

## SDK for Ruby

 Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
# 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
```

- Untuk detail API, lihat [AssumeRole](#) di Referensi AWS SDK for Ruby API.

## WorkDocs Contoh Amazon menggunakan SDK for Ruby

Contoh kode berikut menunjukkan cara melakukan tindakan dan menerapkan skenario umum AWS SDK for Ruby dengan menggunakan Amazon WorkDocs.

Tindakan merupakan kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Meskipun tindakan menunjukkan cara memanggil setiap fungsi layanan, Anda dapat melihat tindakan dalam konteks pada skenario yang terkait dan contoh lintas layanan.

Skenario adalah contoh kode yang menunjukkan cara untuk menyelesaikan tugas tertentu dengan memanggil beberapa fungsi dalam layanan yang sama.

Setiap contoh menyertakan tautan ke GitHub, di mana Anda dapat menemukan petunjuk tentang cara mengatur dan menjalankan kode dalam konteks.

Topik

- [Tindakan](#)

### Tindakan

Jelaskan isi folder root

Contoh kode berikut menunjukkan cara mendeskripsikan isi folder WorkDocs root Amazon.

SDK for Ruby

#### Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
# 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
  end
end
```

```

    @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}"
    @logger.info ""
  end
rescue Aws::WorkDocs::Errors::ServiceError => e
  @logger.error "Error listing folder contents: #{e.message}"
  exit(1)
end

```

- Untuk detail API, lihat [DescribeRootFolders](#) di Referensi AWS SDK for Ruby API.

Jelaskan pengguna

Contoh kode berikut menunjukkan cara mendeskripsikan WorkDocs pengguna Amazon.

SDK for Ruby

#### Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
# 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}"
  exit(1)
end
```

- Untuk detail API, lihat [DescribeUsers](#) di Referensi AWS SDK for Ruby API.

## Contoh lintas layanan menggunakan SDK for Ruby

Contoh aplikasi berikut menggunakan AWS SDK for Ruby untuk bekerja di beberapa Layanan AWS.

Contoh lintas layanan menargetkan pengalaman tingkat lanjut untuk membantu Anda mulai membangun aplikasi.

Contoh-contoh

- [Buat aplikasi yang menganalisis umpan balik pelanggan dan mensintesis audio](#)



# Buat aplikasi yang menganalisis umpan balik pelanggan dan mensintesis audio

## SDK for Ruby

Aplikasi contoh ini menganalisis dan menyimpan kartu umpan balik pelanggan. Secara khusus, ini memenuhi kebutuhan hotel fiktif di New York City. Hotel menerima umpan balik dari para tamu dalam berbagai bahasa dalam bentuk kartu komentar fisik. Umpan balik itu diunggah ke aplikasi melalui klien web. Setelah gambar kartu komentar diunggah, langkah-langkah berikut terjadi:

- Teks diekstraksi dari gambar menggunakan Amazon Textract.
- Amazon Comprehend menentukan sentimen teks yang diekstraksi dan bahasanya.
- Teks yang diekstraksi diterjemahkan ke bahasa Inggris menggunakan Amazon Translate.
- Amazon Polly mensintesis file audio dari teks yang diekstraksi.

Aplikasi lengkap dapat digunakan dengan AWS CDK Untuk kode sumber dan petunjuk penerapan, lihat proyek di [GitHub](#).

Layanan yang digunakan dalam contoh ini

- Amazon Comprehend
- Lambda
- Amazon Polly
- Amazon Textract
- Amazon Translate

# Keamanan untuk AWS SDK for Ruby

Keamanan cloud di Amazon Web Services (AWS) merupakan prioritas tertinggi. Sebagai pelanggan AWS, Anda mendapatkan manfaat dari pusat data dan arsitektur jaringan yang dibangun untuk memenuhi persyaratan organisasi yang paling sensitif terhadap keamanan. Keamanan adalah tanggung jawab bersama antara Anda AWS dan Anda. [Model Tanggung Jawab Bersama](#) menggambarkan ini sebagai Keamanan dari Cloud dan Keamanan dalam Cloud.

Security of the Cloud - AWS bertanggung jawab untuk melindungi infrastruktur yang menjalankan semua layanan yang ditawarkan di AWS Cloud dan memberi Anda layanan yang dapat Anda gunakan dengan aman. Tanggung jawab keamanan kami adalah prioritas tertinggi di AWS, dan efektivitas keamanan kami secara teratur diuji dan diverifikasi oleh auditor pihak ketiga sebagai bagian dari [Program AWS Kepatuhan](#).

Keamanan di Cloud — Tanggung jawab Anda ditentukan oleh yang Layanan AWS Anda gunakan, dan faktor-faktor lain termasuk sensitivitas data Anda, persyaratan organisasi Anda, dan hukum dan peraturan yang berlaku.

## Topik

- [Perlindungan Data di AWS SDK for Ruby](#)
- [Identity and Access Management untuk AWS SDK for Ruby](#)
- [Validasi Kepatuhan untuk AWS SDK for Ruby](#)
- [Ketahanan untuk AWS SDK for Ruby](#)
- [Keamanan Infrastruktur untuk AWS SDK for Ruby](#)
- [Menerapkan versi TLS minimum di AWS SDK for Ruby](#)
- [Migrasi Klien Enkripsi Amazon S3](#)

## Perlindungan Data di AWS SDK for Ruby

[Model tanggung jawab AWS bersama model](#) berlaku untuk perlindungan data di. Seperti yang dijelaskan dalam model AWS ini, bertanggung jawab untuk melindungi infrastruktur global yang menjalankan semua AWS Cloud. Anda bertanggung jawab untuk memelihara kendali atas isi yang dihost pada infrastruktur ini. Anda juga bertanggung jawab atas tugas-tugas konfigurasi dan manajemen keamanan untuk Layanan AWS yang Anda gunakan. Lihat informasi yang lebih lengkap

tentang privasi data dalam [Pertanyaan Umum Privasi Data](#). Lihat informasi tentang perlindungan data di Eropa di pos blog [Model Tanggung Jawab Bersama dan GDPR AWS](#) di Blog Keamanan AWS .

Untuk tujuan perlindungan data, kami menyarankan Anda melindungi Akun AWS kredensial dan mengatur pengguna individu dengan AWS IAM Identity Center atau AWS Identity and Access Management (IAM). Dengan cara itu, setiap pengguna hanya diberi izin yang diperlukan untuk memenuhi tanggung jawab tugasnya. Kami juga menyarankan supaya Anda mengamankan data dengan cara-cara berikut:

- Gunakan autentikasi multi-faktor (MFA) pada setiap akun.
- Gunakan SSL/TLS untuk berkomunikasi dengan sumber daya. AWS Kami mensyaratkan TLS 1.2 dan menganjurkan TLS 1.3.
- Siapkan API dan pencatatan aktivitas pengguna dengan AWS CloudTrail.
- Gunakan solusi AWS enkripsi, bersama dengan semua kontrol keamanan default di dalamnya Layanan AWS.
- Gunakan layanan keamanan terkelola lanjut seperti Amazon Macie, yang membantu menemukan dan mengamankan data sensitif yang disimpan di Amazon S3.
- Jika Anda memerlukan modul kriptografi tervalidasi FIPS 140-2 saat mengakses AWS melalui antarmuka baris perintah atau API, gunakan titik akhir FIPS. Lihat informasi yang lebih lengkap tentang titik akhir FIPS yang tersedia di [Standar Pemrosesan Informasi Federal \(FIPS\) 140-2](#).

Kami sangat merekomendasikan agar Anda tidak pernah memasukkan informasi identifikasi yang sensitif, seperti nomor rekening pelanggan Anda, ke dalam tanda atau bidang isian bebas seperti bidang Nama. Ini termasuk saat Anda bekerja dengan atau lainnya Layanan AWS menggunakan konsol, API AWS CLI, atau AWS SDK. Data apa pun yang Anda masukkan ke dalam tanda atau bidang isian bebas yang digunakan untuk nama dapat digunakan untuk log penagihan atau log diagnostik. Saat Anda memberikan URL ke server eksternal, kami sangat menganjurkan supaya Anda tidak menyertakan informasi kredensial di dalam URL untuk memvalidasi permintaan Anda ke server itu.

## Identity and Access Management untuk AWS SDK for Ruby

AWS Identity and Access Management (IAM) adalah layanan Amazon Web Services (AWS) yang membantu administrator mengontrol akses ke AWS sumber daya dengan aman. Administrator IAM mengontrol siapa yang dapat terautentikasi (masuk) dan berwenang (memiliki izin) untuk

menggunakan sumber daya Layanan AWS. IAM adalah Layanan AWS yang dapat Anda gunakan tanpa biaya tambahan.

Untuk menggunakan AWS SDK for Ruby AWS untuk mengakses, Anda AWS memerlukan akun AWS dan kredensial. Untuk meningkatkan keamanan akun AWS Anda, kami menyarankan Anda menggunakan pengguna IAM untuk menyediakan kredensial akses alih-alih menggunakan kredensial akun AWS Anda.

Untuk detail tentang bekerja dengan IAM, lihat [IAM](#).

Untuk gambaran umum tentang pengguna IAM dan mengapa mereka penting untuk keamanan akun Anda, lihat [Kredensial keamanan AWS](#) dalam [Referensi Umum Amazon Web Services](#).

AWS SDK for Ruby [mengikuti model tanggung jawab bersama](#) melalui layanan Amazon Web Services ( ) tertentu yang didukungnya. Untuk informasi Layanan AWS keamanan, lihat [halaman dokumentasi Layanan AWS keamanan](#) dan [Layanan AWS yang berada dalam lingkup upaya AWS kepatuhan oleh program kepatuhan](#).

## Validasi Kepatuhan untuk AWS SDK for Ruby

AWS SDK for Ruby [mengikuti model tanggung jawab bersama](#) melalui layanan Amazon Web Services ( ) tertentu yang didukungnya. Untuk informasi Layanan AWS keamanan, lihat [halaman dokumentasi Layanan AWS keamanan](#) dan [Layanan AWS yang berada dalam lingkup upaya AWS kepatuhan oleh program kepatuhan](#).

Keamanan dan kepatuhan layanan Amazon Web Services (AWS) dinilai oleh auditor pihak ketiga sebagai bagian dari beberapa program AWS kepatuhan. Ini termasuk SOC, PCI, FedRAMP, HIPAA, dan lainnya. AWS menyediakan daftar yang sering diperbarui Layanan AWS dalam lingkup program kepatuhan khusus di [AWS Layanan dalam Lingkup oleh Program Kepatuhan](#).

Laporan audit pihak ketiga tersedia untuk Anda unduh AWS Artifact. Untuk informasi selengkapnya, lihat [Mengunduh Laporan di AWS Artifak](#).

Untuk informasi selengkapnya tentang program AWS kepatuhan, lihat [Program AWS Kepatuhan](#).

Tanggung jawab kepatuhan Anda saat menggunakan AWS SDK for Ruby untuk Layanan AWS mengakses suatu ditentukan oleh sensitivitas data Anda, tujuan kepatuhan organisasi Anda, serta hukum dan peraturan yang berlaku. Jika penggunaan Anda tunduk pada kepatuhan terhadap standar seperti HIPAA, PCI, atau FedRAMP, menyediakan sumber daya untuk membantu: Layanan AWS AWS

- [Panduan Memulai Cepat Keamanan dan Kepatuhan — Panduan](#) penerapan yang membahas pertimbangan arsitektur dan memberikan langkah-langkah untuk menerapkan lingkungan dasar yang berfokus pada keamanan dan berfokus pada kepatuhan. AWS
- [Arsitektur untuk Whitepaper Keamanan dan Kepatuhan HIPAA — Whitepaper](#) yang menjelaskan bagaimana perusahaan dapat menggunakan untuk membuat aplikasi yang sesuai dengan HIPAA. AWS
- [AWS Sumber Daya Kepatuhan](#) — Kumpulan buku kerja dan panduan yang mungkin berlaku untuk industri dan lokasi Anda.
- [AWS Config](#) — Layanan yang menilai seberapa baik konfigurasi sumber daya Anda mematuhi praktik internal, pedoman industri, dan peraturan.
- [AWS Security Hub](#) — Pandangan komprehensif tentang status keamanan Anda di dalamnya AWS yang membantu Anda memeriksa kepatuhan Anda terhadap standar industri keamanan dan praktik terbaik.

## Ketahanan untuk AWS SDK for Ruby

Infrastruktur global Amazon Web Services (AWS) dibangun di sekitar Wilayah AWS dan Availability Zones.

Wilayah AWS menyediakan beberapa Availability Zone yang terpisah secara fisik dan terisolasi, yang terhubung dengan latensi rendah, throughput tinggi, dan jaringan yang sangat redundan.

Dengan Zona Ketersediaan, Anda dapat merancang dan mengoperasikan aplikasi dan basis data yang melakukan secara otomatis pindah saat gagal/failover di antara zona-zona tanpa terputus. Zona Ketersediaan lebih sangat tersedia, lebih toleran kesalahan, dan lebih dapat diskalakan daripada infrastruktur pusat data tunggal atau multi tradisional.

Untuk informasi selengkapnya tentang Wilayah AWS dan Availability Zone, lihat [Infrastruktur AWS Global](#).

AWS SDK for Ruby [mengikuti model tanggung jawab bersama](#) melalui layanan Amazon Web Services ( ) tertentu yang didukungnya. Untuk informasi Layanan AWS keamanan, lihat [halaman dokumentasi Layanan AWS keamanan](#) dan [Layanan AWS yang berada dalam lingkup upaya AWS kepatuhan oleh program kepatuhan](#).

## Keamanan Infrastruktur untuk AWS SDK for Ruby

AWS SDK for Ruby [mengikuti model tanggung jawab bersama](#) melalui layanan Amazon Web Services (AWS) tertentu yang didukung. Untuk informasi Layanan AWS keamanan, lihat [halaman dokumentasi Layanan AWS keamanan](#) dan [Layanan AWS yang berada dalam lingkup upaya AWS kepatuhan oleh program kepatuhan](#).

Untuk informasi tentang proses AWS keamanan, lihat [AWS whitepaper: Ringkasan Proses Keamanan](#).

## Menerapkan versi TLS minimum di AWS SDK for Ruby

Komunikasi antara AWS SDK for Ruby dan diamankan menggunakan Secure Sockets Layer (SSL) atau Transport Layer Security (TLS). Semua versi SSL, dan versi TLS lebih awal dari 1.2, memiliki kerentanan yang dapat membahayakan keamanan komunikasi Anda. Untuk alasan ini, Anda harus memastikan bahwa Anda menggunakan AWS SDK for Ruby dengan versi Ruby yang mendukung TLS versi 1.2 atau yang lebih baru.

Ruby menggunakan perpustakaan OpenSSL untuk mengamankan koneksi HTTP. Versi Ruby yang didukung (1.9.3 dan yang lebih baru) diinstal melalui [manajer paket](#) sistem (yum, dan lainnya) apt, [installer resmi](#), atau [manajer](#) Ruby (rbenv, RVM, dan lainnya) biasanya menggabungkan OpenSSL 1.0.1 atau yang lebih baru, yang mendukung TLS 1.2.

Ketika digunakan dengan versi Ruby yang didukung dengan OpenSSL 1.0.1 atau yang lebih baru, SDK for AWS Ruby lebih memilih TLS 1.2, dan menggunakan versi terbaru SSL atau TLS yang didukung oleh klien dan server. Ini selalu setidaknya TLS 1.2 untuk Layanan AWS. (SDK menggunakan `Net::HTTP` kelas Ruby dengan.) `use_ssl=true`

## Memeriksa versi OpenSSL

Untuk memastikan instalasi Ruby Anda menggunakan OpenSSL 1.0.1 atau yang lebih baru, masukkan perintah berikut.

```
ruby -r openssl -e 'puts OpenSSL::OPENSSL_VERSION'
```

Cara alternatif untuk mendapatkan versi OpenSSL adalah dengan menanyakan `openssl` executable secara langsung. Pertama, cari executable yang sesuai menggunakan perintah berikut.

```
ruby -r rbconfig -e 'puts RbConfig::CONFIG["configure_args"]'
```

Output harus `--with-openssl-dir=/path/to/openssl` menunjukkan lokasi instalasi OpenSSL. Catat jalan ini. Untuk memeriksa versi OpenSSL, masukkan perintah berikut.

```
cd /path/to/openssl
bin/openssl version
```

Metode terakhir ini mungkin tidak bekerja dengan semua instalasi Ruby.

## Meningkatkan dukungan TLS

[Jika versi OpenSSL yang digunakan oleh instalasi Ruby Anda lebih awal dari 1.0.1, tingkatkan instalasi Ruby atau OpenSSL Anda menggunakan manajer paket sistem, penginstal Ruby, atau manajer Ruby, seperti yang dijelaskan dalam panduan instalasi Ruby.](#) Jika Anda menginstal Ruby [dari sumber](#), instal [OpenSSL terbaru](#) terlebih dahulu, lalu teruskan saat menjalankan. `--with-openssl-dir=/path/to/upgraded/openssl ./configure`

## Migrasi Klien Enkripsi Amazon S3

Topik ini menunjukkan cara memigrasikan aplikasi Anda dari klien enkripsi Amazon Simple Storage Service (Amazon S3) ke Versi 1 (V1) Amazon Simple Storage Service (Amazon S3) ke Versi 2 (V2), dan memastikan ketersediaan aplikasi selama proses migrasi.

### Ikhtisar Migrasi

Migrasi ini terjadi dalam dua fase:

1. Perbarui klien yang ada untuk membaca format baru. Pertama, gunakan versi terbaru AWS SDK for Ruby ke aplikasi Anda. Ini akan memungkinkan klien enkripsi V1 yang ada untuk mendekripsi objek yang ditulis oleh klien V2 baru. Jika aplikasi Anda menggunakan beberapa AWS SDK, Anda harus memutakhirkan setiap SDK secara terpisah.
2. Migrasikan enkripsi dan dekripsi klien ke V2. Setelah semua klien enkripsi V1 Anda dapat membaca format baru, Anda dapat memigrasikan klien enkripsi dan dekripsi yang ada ke versi V2 masing-masing.

### Perbarui Klien yang Ada untuk Membaca Format Baru

Klien enkripsi V2 menggunakan algoritma enkripsi yang tidak didukung oleh versi klien yang lebih lama. Langkah pertama dalam migrasi adalah memperbarui klien dekripsi V1 Anda ke rilis SDK terbaru. Setelah menyelesaikan langkah ini, klien V1 aplikasi Anda akan dapat mendekripsi objek

yang dienkripsi oleh klien enkripsi V2. Lihat detail di bawah untuk setiap versi utama AWS SDK for Ruby.

## Perbarui AWS SDK for Ruby Versi 3

Versi 3 adalah versi terbaru dari AWS SDK For Ruby. Untuk menyelesaikan migrasi ini, Anda perlu menggunakan permata versi 1.76.0 atau yang lebih baru. `aws-sdk-s3`

Menginstal dari baris perintah

Untuk proyek yang menginstal `aws-sdk-s3` permata, gunakan opsi versi untuk memverifikasi bahwa versi minimum 1.76.0 diinstal.

```
gem install aws-sdk-s3 -v '>= 1.76.0'
```

### Menggunakan Gemfiles

Untuk proyek yang menggunakan Gemfile untuk mengelola dependensi, atur versi minimum `aws-sdk-s3` permata ke 1.76.0. Sebagai contoh:

```
gem 'aws-sdk-s3', '>= 1.76.0'
```

1. Ubah Gemfile Anda.
2. Jalankan `bundle update aws-sdk-s3`. Untuk memverifikasi versi Anda, jalankan `bundle info aws-sdk-s3`.

## Upgdate AWS SDK for Ruby Versi 2

Versi 2 AWS SDK for Ruby [akan memasuki](#) mode pemeliharaan pada 21 November 2021. Untuk menyelesaikan migrasi ini, Anda perlu menggunakan permata `aws-sdk` versi 2.11.562 atau yang lebih baru.

Menginstal dari baris perintah

Untuk proyek yang menginstal `aws-sdk` permata, dari baris perintah, gunakan opsi versi untuk memverifikasi bahwa versi minimum 2.11.562 diinstal.

```
gem install aws-sdk -v '>= 2.11.562'
```

### Menggunakan Gemfiles



Untuk proyek yang menggunakan Gemfile untuk mengelola dependensi, atur versi minimum `aws-sdk` permata ke 2.11.562. Sebagai contoh:

```
gem 'aws-sdk', '>= 2.11.562'
```

1. Ubah Gemfile Anda. Jika Anda memiliki file Gemfile.lock, hapus atau perbarui.
2. Jalankan `bundle update aws-sdk`. Untuk memverifikasi versi Anda, jalankan `bundle info aws-sdk`.

## Migrasi Klien Enkripsi dan Dekripsi ke V2

Setelah memperbarui klien Anda untuk membaca format enkripsi baru, Anda dapat memperbarui aplikasi Anda ke klien enkripsi dan dekripsi V2. Langkah-langkah berikut menunjukkan kepada Anda bagaimana berhasil memigrasikan kode Anda dari V1 ke V2.

Sebelum memperbarui kode Anda untuk menggunakan klien enkripsi V2, pastikan Anda telah mengikuti langkah-langkah sebelumnya dan menggunakan `aws-sdk-s3` permata versi 2.11.562 atau yang lebih baru.

### Note

Saat mendekripsi dengan AES-GCM, baca seluruh objek sampai akhir sebelum Anda mulai menggunakan data yang didekripsi. Ini untuk memverifikasi bahwa objek belum dimodifikasi sejak dienkripsi.

## Mengkonfigurasi Klien Enkripsi V2

`EncryptionV2::Client` membutuhkan konfigurasi tambahan. Untuk informasi konfigurasi terperinci, lihat [dokumentasi EncryptionV2::Client](#) atau contoh yang diberikan nanti dalam topik ini.

1. Metode pembungkus kunci dan algoritma enkripsi konten harus ditentukan pada konstruksi klien. Saat membuat yang baru `EncryptionV2::Client`, Anda perlu memberikan nilai untuk `key_wrap_schema` dan `content_encryption_schema`.

`key_wrap_schema`- Jika Anda menggunakan AWS KMS, ini harus diatur ke `:kms_context`. Jika Anda menggunakan kunci simetris (AES), itu harus diatur ke `:aes_gcm`. Jika Anda menggunakan kunci asimetris (RSA), itu harus diatur ke `:rsa_oaep_sha1`.

`content_encryption_schema`- Ini harus diatur ke:`aes_gcm_no_padding`.

2. `security_profile` harus ditentukan pada konstruksi klien. Saat membuat yang baru `EncryptionV2::Client`, Anda perlu memberikan nilai untuk `security_profile`. Parameter `security_profile` menentukan dukungan untuk membaca objek yang ditulis menggunakan V1 yang lebih lama. `Encryption::Client` Ada dua nilai `:v2` dan `:v2_and_legacy`. Untuk mendukung migrasi, atur `security_profile` ke `:v2_and_legacy`. Gunakan: `v2` hanya untuk pengembangan aplikasi baru.

3. AWS KMS key ID diberlakukan secara default. Di `V1Encryption::Client`, yang `kms_key_id` digunakan untuk membuat klien tidak diberikan kepada `AWS KMS Decrypt` call `AWS KMS` bisa mendapatkan informasi ini dari metadata dan menambahkannya ke gumpalan ciphertext simetris. Dalam `V2, EncryptionV2::Client`, `kms_key_id` diteruskan ke panggilan `AWS KMS` Dekripsi, dan panggilan gagal jika tidak cocok dengan kunci yang digunakan untuk mengenkripsi objek. Jika kode Anda sebelumnya mengandalkan tidak menyetel yang spesifik `kms_key_id`, setel `kms_key_id: :kms_allow_decrypt_with_any_cmk` pada pembuatan klien atau setel `kms_allow_decrypt_with_any_cmk: true` `get_object` panggilan.

## Contoh: Menggunakan Kunci Symmetric (AES)

### Pra-migrasi

```
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)
```

### Pasca-migrasi

```
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
```

## Contoh: Menggunakan AWS KMS dengan kms\_key\_id

### Pra-migrasi

```
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)
```

### Pasca-migrasi

```
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
```

## Contoh: Menggunakan AWS KMS tanpa kms\_key\_id

### Pra-migrasi

```
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)
```

### Pasca-migrasi

```
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_cmek:
  true) # To allow decrypting with any cmk
```

## Alternatif Pasca Migrasi

Jika Anda hanya membaca dan mendekripsi (tidak pernah menulis dan mengenkripsi) objek menggunakan klien enkripsi S2, gunakan kode ini.

```
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
```

# Riwayat Dokumen

Tabel berikut menjelaskan perubahan penting dalam panduan ini. Untuk notifikasi tentang pembaruan dokumentasi ini, Anda dapat berlangganan ke [umpan RSS](#).

Perubahan	Deskripsi	Tanggal
<a href="#">Daftar Isi</a>	Daftar isi yang diperbarui untuk membuat contoh kode lebih mudah diakses.	1 Juni 2023 Juni 2023 Juni 2023
<a href="#">Praktik terbaik.</a>	Panduan yang diperbarui untuk menyelaraskan dengan praktik terbaik IAM. Untuk informasi selengkapnya tentang <a href="#">keamanan dalam IAM</a> . Pembaruan untuk Memulai.	8 Mei 2023 Mei 2023 Mei 2023
<a href="#">Pembaruan umum</a>	Memperbarui halaman selamat datang untuk sumber daya eksternal yang relevan. Juga diperbarui versi Ruby minimum yang diperlukan untuk v2.3. AWS Key Management Service Bagian yang diperbarui untuk mencerminkan pembaruan terminologi. Informasi penggunaan yang diperbarui pada utilitas REPL untuk kejelasan.	8 Agustus 2022 2020 2022 Agustus 2022 Mei 2022
<a href="#">Memperbaiki tautan yang rusak</a>	Tetap contoh rusak link. Menghapus halaman Tips dan Trik berlebihan; mengarahkan ulang ke konten contoh Amazon EC2. Termasuk daftar	3 Agustus 2022 2020 2022 Agustus 2022 Mei 2022

contoh kode yang tersedia di GitHub dalam repositori Contoh Kode.

[Mendapatkan Informasi tentang Semua Grup Keamanan Amazon RDS](#)

Catatan tambahan tentang pensiun EC2-Classic.

26 Juli 2022 2022 2022 Juli 2022 2022

[Metrik SDK](#)

Menghapus informasi tentang mengaktifkan Metrik SDK untuk Support Perusahaan, yang telah usang.

28 Januari 2022 2020 2022 Januari 2022 Mei 2022

Terjemahan disediakan oleh mesin penerjemah. Jika konten terjemahan yang diberikan bertentangan dengan versi bahasa Inggris aslinya, utamakan versi bahasa Inggris.