



Panduan Developer

AWS Serverless Application Repository



AWS Serverless Application Repository: Panduan Developer

Copyright © 2023 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 di antara para pelanggan, atau dengan cara apa pun yang menghina atau mendiskreditkan Amazon. Semua merek dagang lain yang tidak dimiliki oleh Amazon adalah milik dari pemiliknya masing-masing, yang mungkin atau tidak berafiliasi dengan, terhubung ke, atau disponsori oleh Amazon.

Table of Contents

ApaAWS Serverless Application Repository?	1
Langkah Selanjutnya	1
Mulai Cepat: Penerbitan Aplikasi	3
Gambaran Umum	3
Aplikasi Hello World	3
Sebelum Anda Memulai	4
Langkah 1: Inisialisasikan Aplikasi	4
Langkah 2: Uji Aplikasi secara Lokal	5
Langkah 3: Buat Package aplikasi	6
Langkah 4: Publikasikan Aplikasi	8
Langkah Selanjutnya	8
Informasi Selengkapnya	9
Publikasi Aplikasi	10
Menggunakan AWS SAM dengan AWS Serverless Application Repository	11
DidukungAWSSumber Daya diAWS Serverless Application Repository	11
Templat Kebijakan	12
Daftar DidukungAWSSumber Daya	12
Cara Mempublikasikan Aplikasi	19
Menerbitkan Aplikasi (AWS CLI)	19
Menerbitkan Aplikasi Baru (Konsol)	20
Berbagi Aplikasi	24
Membatalkan Berbagi Aplikasi	27
Menghapus Aplikasi	28
Menerbitkan Versi Aplikasi Baru	29
Lencana Penulis Terverifikasi	30
Meminta Lencana Penulis Terverifikasi	31
Lapisan Lambda	31
Cara Kerjanya	32
Contoh	32
Men-deploy Aplikasi	34
Izin Deployment Aplikasi	34
Kemampuan Aplikasi	35
Menemukan dan Mengakui Kemampuan Aplikasi (Konsol)	36
Melihat Kemampuan Aplikasi (AWS CLI)	36

Cara Menyebarkan Aplikasi	36
Menerapkan Aplikasi Baru (Konsol)	37
Menerapkan Aplikasi Baru (AWS CLI)	38
Menghapus Tumpukan Aplikasi	39
Memperbarui Aplikasi	40
Keamanan	42
Perlindungan Data	43
Enkripsi Saat Data Berpindah	44
Enkripsi saat Data Tidak Berpindah	44
Identity and Access Management	44
Penonton	45
Mengautentikasi dengan Identitas	45
Mengelola Akses Menggunakan Kebijakan	49
Bagaimana AWS Serverless Application Repository Bekerja dengan IAM	51
Contoh Kebijakan Berbasis Identitas	57
Contoh Kebijakan Aplikasi	66
AWS Serverless Application Repository Referensi Izin API	72
Pemecahan Masalah	74
Pencatatan dan Pemantauan	77
Mencatat AWS Serverless Application Repository panggilan API dengan AWS CloudTrail	77
Validasi Kepatuhan	81
Ketahanan	82
Keamanan Infrastruktur	82
Quotas	84
Pemecahan Masalah	85
Anda Tidak Bisa Membuat Aplikasi Publik	85
Kuota Terlampaui	86
File Readme yang Diperbarui Tidak Muncul Segera	86
Anda Tidak Dapat Menyebarkan Aplikasi Karena Izin IAM Tidak Cukup	86
Anda Tidak Dapat Menyebarkan Aplikasi yang Sama Dua Kali	86
Mengapa Aplikasi Saya Tidak Tersedia untuk Umum	87
Menghubungi Support	87
Operasi	88
Sumber daya	90
Applications	90
URI	90

Metode HTTP	90
Skema	92
Properti	96
Lihat juga	114
ApplicationId Aplikasi	115
URI	115
Metode HTTP	115
Skema	119
Properti	121
Lihat juga	135
Aplikasi ApplicationId Changesets	136
URI	136
Metode HTTP	136
Skema	137
Properti	139
Lihat juga	147
Applications applicationId Dependencies	148
URI	148
Metode HTTP	148
Skema	150
Properti	152
Lihat juga	155
Kebijakan ApplicationID Aplikasi	155
URI	155
Metode HTTP	156
Skema	158
Properti	160
Lihat juga	163
Applications applicationId Templates	164
URI	164
Metode HTTP	164
Skema	166
Properti	167
Lihat juga	171
Applications applicationId Templates templateId	172
URI	172

Metode HTTP	172
Skema	174
Properti	175
Lihat juga	179
Applications applicationId Unshare	180
URI	180
Metode HTTP	180
Skema	181
Properti	183
Lihat juga	185
Versi ApplicationId Aplikasi	186
URI	186
Metode HTTP	186
Skema	188
Properti	189
Lihat juga	193
Aplikasi ApplicationID Versi SemanticVersion	193
URI	193
Metode HTTP	193
Skema	195
Properti	197
Lihat juga	206
Riwayat Dokumen	208
AWSGlosarium	212
.....	ccxiii

ApaAWS Serverless Application Repository?

ParameterAWS Serverless Application Repositorymemudahkan pengembang dan perusahaan untuk dengan cepat menemukan, menyebarkan, dan mempublikasikan aplikasi tanpa server diAWSCloud. Untuk informasi selengkapnya tentang aplikasi tanpa server, lihat[Komputasi dan Aplikasi Tanpa Server](#)padaAWSsitus web.

Anda dapat dengan mudah mempublikasikan aplikasi, membagikannya secara publik dengan komunitas secara luas, atau secara pribadi dalam tim Anda atau di seluruh organisasi Anda. Untuk mempublikasikan aplikasi tanpa server (atau aplikasi), Anda dapat menggunakanAWS Management Console, yangAWS SAMantarmuka baris perintah (AWS SAMCLI), atauAWSSDK untuk mengunggah kode Anda. Seiring dengan kode Anda, Anda mengunggah file manifes sederhana, juga dikenal sebagaiAWS Serverless Application Model(AWS SAM) templat. Untuk informasi selengkapnya tentang AWS SAM, lihat [Panduan Developer AWS Serverless Application Model](#).

ParameterAWS Serverless Application Repositorysangat terintegrasi denganAWS Lambdakonsol. Integrasi ini berarti bahwa pengembang dari semua tingkatan dapat memulai dengan komputasi tanpa server tanpa perlu mempelajari sesuatu yang baru. Anda dapat menggunakan kata kunci kategori untuk menelusuri aplikasi seperti backend web dan seluler, aplikasi pemrosesan data, atau chatbots. Anda juga dapat mencari aplikasi berdasarkan nama, penerbit, atau sumber acara. Untuk menggunakan aplikasi, Anda cukup memilihnya, mengkonfigurasi bidang yang diperlukan, dan menerapkannya dengan beberapa klik.

Dalam panduan ini, Anda dapat mempelajari tentang dua cara untuk bekerja denganAWS Serverless Application Repository:

- [Publikasi Aplikasi](#)— Mengkonfigurasi dan meng-upload aplikasi untuk membuat mereka tersedia untuk pengembang lain, dan mempublikasikan versi baru dari aplikasi.
- [Men-deploy Aplikasi](#)— Browse untuk aplikasi dan melihat informasi tentang mereka, termasuk kode sumber dan file readme. Instal, konfigurasi, dan sebarkan aplikasi yang Anda pilih.

Langkah Selanjutnya

- Untuk tutorial tentang penerbitan contoh aplikasi keAWS Serverless Application Repository, Lihat[Mulai Cepat: Penerbitan Aplikasi](#).

- Untuk petunjuk tentang penerapan aplikasi dari AWS Serverless Application Repository, Lihat [Cara Menyebarkan Aplikasi](#).

Mulai Cepat: Penerbitan Aplikasi

Panduan ini memandu Anda melalui langkah-langkah untuk men-download, membangun, menguji dan mempublikasikan contoh aplikasi tanpa server untuk AWS Serverless Application Repository menggunakan AWS SAM CLI. Anda dapat menggunakan contoh aplikasi ini sebagai titik awal untuk mengembangkan dan memublikasikan aplikasi tanpa server Anda sendiri.

Gambaran Umum

Langkah-langkah berikut menguraikan cara mengunduh, membangun, dan mempublikasikan contoh aplikasi tanpa server:

1. Inisialisasi. Download contoh aplikasi dari template menggunakan `sam init`.
2. Uji Lokal. Uji aplikasi secara lokal menggunakan `sam local invoke` dan/atau `sam local start-api`. Perhatikan bahwa dengan perintah ini, meskipun fungsi Lambda Anda dipanggil secara lokal, ia membaca dari dan menulis ke AWS sumber daya di AWS Cloud.
3. Package. Ketika Anda puas dengan fungsi Lambda Anda, bundel fungsi Lambda, AWS SAM template, dan dependensi apa pun ke dalam paket AWS CloudFormation deployment menggunakan `sam package`. Pada langkah ini Anda juga akan memasukkan informasi tentang aplikasi yang akan diunggah ke AWS Serverless Application Repository.
4. Publikasikan. Publikasikan aplikasi ke AWS Serverless Application Repository menggunakan `sam publish`. Pada akhir langkah ini, Anda dapat melihat aplikasi Anda AWS Serverless Application Repository dan menerapkannya ke AWS Cloud menggunakan AWS Serverless Application Repository.

Contoh [Aplikasi Hello World](#) di bagian berikutnya memandu Anda melalui langkah-langkah ini dalam membangun dan menerbitkan aplikasi tanpa server.

Aplikasi Hello World

Dalam latihan ini, Anda mengunduh dan menguji aplikasi tanpa server Hello World yang mewakili backend API sederhana. Ini memiliki titik akhir Amazon API Gateway yang mendukung operasi GET dan fungsi Lambda. Ketika permintaan GET dikirim ke titik akhir, API Gateway akan memanggil fungsi Lambda tersebut. Kemudian, AWS Lambda mengeksekusi fungsi, yang hanya mengembalikan `hello world` pesan.

Aplikasi ini memiliki komponen berikut:

- AWS SAMTemplate yang mendefinisikan duaAWS sumber daya untuk aplikasi Hello World: layanan API Gateway dengan operasi GET, dan fungsi Lambda. Template juga mendefinisikan pemetaan antara operasi GET API Gateway dan fungsi Lambda.
- Kode aplikasi yang ditulis dengan Python.

Sebelum Anda Memulai

Pastikan bahwa Anda memiliki pengaturan yang diperlukan untuk latihan ini:

- Anda harus memilikiAWS akun dengan pengguna IAM yang memiliki izin administrator. Lihat [MengaturAWS Akun](#).
- Anda harus memilikiAWS SAM CLI (antarmuka baris perintah) yang terinstal. Lihat [MemasangAWS SAM CLI](#).
- Anda harus memiliki versi 1.16.77 atau yang lebih baru yangAWS CLI terinstal. Lihat [MemasangAWS Command Line Interface](#).

Langkah 1: Inisialisasikan Aplikasi

Dalam bagian ini, Anda mengunduh aplikasi contoh, yang terdiri dari templat AWS SAM dan kode aplikasi.

Untuk menginisialisasi aplikasi

1. Jalankan perintah berikut pada prompt perintah CLI AWS SAM.

```
sam init --runtime python3.6
```

2. Tinjau isi direktori yang dibuat oleh perintah (sam-app/):
 - `template.yaml`— Mendefinisikan duaAWS sumber daya yang dibutuhkan aplikasi Hello World: fungsi Lambda dan titik akhir API Gateway yang mendukung operasi GET. Templat juga menentukan pemetaan antara dua sumber daya.
 - Konten yang terkait dengan kode aplikasi Hello World:
 - `hello_world/direktori` - Berisi kode aplikasi, yang kembali `hello world` ketika Anda menjalankannya.

Note

Untuk latihan ini, kode aplikasi ditulis dengan Python, dan Anda menentukan runtime dalam `init` perintah. AWS Lambda mendukung bahasa tambahan untuk membuat kode aplikasi. Jika Anda menentukan runtime lain yang didukung, `init` perintah tersebut menyediakan kode Hello World dalam bahasa yang ditentukan, dan `README.md` file yang dapat Anda ikuti untuk bahasa tersebut. Untuk informasi tentang runtime yang didukung, lihat [Lingkungan Eksekusi Lambda dan Pustaka yang Tersedia](#).

Langkah 2: Uji Aplikasi secara Lokal

Sekarang setelah Anda memiliki AWS SAM aplikasi di mesin lokal Anda, ikuti langkah-langkah di bawah ini untuk mengujinya secara lokal.

Untuk menguji aplikasi secara lokal

1. Mulai endpoint API Gateway secara lokal. Anda harus menjalankan perintah berikut dari direktori yang berisi `template.yaml` file tersebut.

```
sam-app> sam local start-api --region us-east-1
```

Perintah mengembalikan endpoint API Gateway, yang dapat Anda kirimkan permintaan untuk pengujian lokal.

2. Uji aplikasi. Salin URL endpoint API Gateway, tempelkan di browser, dan pilih Enter. Contoh URL endpoint API Gateway adalah `http://127.0.0.1:3000/hello`.

API Gateway secara lokal memanggil fungsi Lambda tempat endpoint dipetakan. Fungsi Lambda dijalankan dalam kontainer Docker lokal dan kembali `hello world`. API Gateway mengembalikan respons ke browser yang berisi teks.

Latihan: Ubah string pesan

Setelah berhasil menguji aplikasi sampel, Anda dapat bereksperimen dengan membuat modifikasi sederhana: ubah string pesan yang dikembalikan.

1. Edit `hello_world/app.py` file untuk mengubah string pesan dari `'hello world'` ke `'Hello World!'`.
2. Muat ulang URL pengujian di browser Anda dan amati string baru.

Anda akan melihat bahwa kode baru Anda dimuat secara dinamis, tanpa Anda harus me-restart `local` proses.

Langkah 3: Buat Package aplikasi

Setelah menguji aplikasi Anda secara lokal, Anda gunakan AWS SAM CLI untuk membuat paket deployment dan AWS SAM templat yang dikemas.

Note

Pada langkah-langkah berikut, Anda membuat `file.zip` untuk isi `hello_world/` direktori, yang berisi kode aplikasi. File.zip ini adalah paket penyebaran untuk aplikasi tanpa server Anda. Untuk informasi selengkapnya, lihat [Membuat Package Deployment \(Python\)](#) di Panduan AWS Lambda Pengembang.

Untuk membuat paket deployment Lambda

1. Tambahkan `Metadata` bagian ke file AWS SAM template Anda yang menyediakan informasi aplikasi yang diperlukan. Untuk informasi selengkapnya tentang `Metadata` bagian AWS SAM template, lihat [Properti Bagian Metadata AWS SAM Template](#) di Panduan AWS Serverless Application Model Pengembang.

Berikut adalah `Metadata` bagian contoh:

```
Metadata:
  AWS::ServerlessRepo::Application:
    Name: my-app
    Description: hello world
    Author: user1
    SpdxLicenseId: Apache-2.0
    LicenseUrl: LICENSE.txt
```

```
ReadmeUrl: README.md
Labels: ['tests']
HomePageUrl: https://github.com/user1/my-app-project
SemanticVersion: 0.0.1
SourceCodeUrl: https://github.com/user1/my-app-project
```

ReadmeUrl, LicenseUrl, dan properti dapat berupa referensi ke file lokal (seperti pada contoh di atas), atau dapat berupa tautan ke bucket Amazon S3 yang sudah menampung artefak ini.

2. Buat bucket S3 di lokasi tempat Anda ingin menyimpan kode yang dipaketkan. Jika Anda ingin menggunakan bucket S3 yang sudah ada, lewati langkah ini.

```
sam-app> aws s3 mb s3://bucketname
```

3. Buat paket deployment fungsi Lambda dengan menjalankan perintah `packageAWS` SAM CLI berikut ini.

```
sam-app> sam package \  
  --template-file template.yaml \  
  --output-template-file packaged.yaml \  
  --s3-bucket bucketname
```

Perintah ini melakukan hal berikut:

- Menitsleting konten `aws-sam/hello_world/` direktori dan mengunggahnya ke Amazon S3.
- Mengunggah paket penerapan, file README, dan file LICENSE ke bucket Amazon S3 yang ditentukan oleh `--s3-bucket` opsi.
- Menampilkan file templat baru, yang disebut `packaged.yaml`, yang Anda gunakan di langkah berikutnya untuk memublikasikan aplikasi tersebut AWS Serverless Application Repository. File `packaged.yaml` templat mirip dengan file templat asli (`template.yaml`), tetapi memiliki perbedaan kunci—`ReadmeUrl` properti `int` `CodeUri`, `LicenseUrl`, dan menunjuk pada bucket Amazon S3 dan objek yang mencakup artefaknya masing-masing. Sinppet berikut dari contoh file templat `packaged.yaml` menunjukkan properti `CodeUri`:

```
HelloWorldFunction:
  Type: AWS::Serverless::Function # For more information about function
  resources, see https://github.com/awslabs/serverless-application-model/blob/
  master/versions/2016-10-31.md#awsserverlessfunction
  Properties:
```

```
CodeUri: s3://bucketname/fb77a3647a4f47a352fc0bjectGUID
```

```
...
```

Langkah 4: Publikasikan Aplikasi

Setelah Anda membuat paket deployment tersebut, Anda menggunakannya untuk memublikasikan aplikasi keAWS Serverless Application Repository.

Untuk memublikasikan aplikasi tanpa server keAWS Serverless Application Repository

- Jalankan perintah berikut untuk memublikasikan aplikasi baruAWS Serverless Application Repository dengan versi pertama yang dibuat sebagai 0.0.1.

```
sam-app> sam publish \  
  --template packaged.yaml \  
  --region us-east-1
```

Note

Aplikasi akan dibuat sebagai pribadi secara default. Anda harus berbagi aplikasi sebelumAWS Akun lain akan diizinkan untuk melihat dan men-deploy aplikasi Anda. Lihat Langkah Berikutnya di bawah ini untuk detail selengkapnya tentang berbagi aplikasi Anda.

Langkah Selanjutnya

Sekarang bahwa Anda telah menerbitkan aplikasi sampel Anda, berikut adalah beberapa hal yang mungkin ingin Anda lakukan dengan itu.

- Lihat Aplikasi Anda diAWS Serverless Application Repository - Output dari `publish` perintah akan menyertakan tautan keAWS Serverless Application Repository langsung ke halaman detail aplikasi Anda. Anda juga dapat membuka halamanAWS Serverless Application Repository arahan dan mencari aplikasi Anda.
- Bagikan Aplikasi Anda — Karena aplikasi Anda diatur ke pribadi secara default, aplikasi tersebut tidak terlihat olehAWS Akun lain. Untuk berbagi aplikasi Anda dengan orang lain, Anda harus membuatnya publik atau memberikan izin untuk daftarAWS Akun tertentu. Untuk informasi tentang

berbagi aplikasi Anda menggunakan AWS CLI lihat [AWS Serverless Application Repository Contoh Kebijakan Aplikasi](#). Untuk informasi tentang berbagi aplikasi Anda menggunakan konsol, lihat [Berbagi Aplikasi](#).

Informasi Selengkapnya

Untuk informasi selengkapnya tentang Metadata bagian AWS SAM template, `sam package` dan `sam publish` perintah AWS SAM CLI, lihat [Menerbitkan Aplikasi Menggunakan AWS SAM CLI](#) di Panduan AWS Serverless Application Model Pengembang.

Publikasi Aplikasi

Ketika Anda memublikasikan aplikasi nirserver keAWS Serverless Application Repository, Anda membuatnya tersedia dan di-deploy orang lain.

Anda terlebih dahulu menentukan aplikasi Anda denganAWS Serverless Application Model(AWS SAM) templat. Ketika Anda menentukan aplikasi Anda, Anda harus mempertimbangkan apakah konsumen aplikasi Anda akan diminta untuk mengakui kemampuan aplikasi. Untuk informasi lebih lanjut tentang penggunaanAWS SAMdan mengakui kemampuan, lihat[Menggunakan AWS SAM dengan AWS Serverless Application Repository](#).

Anda dapat memublikasikan aplikasi tanpa server dengan menggunakanAWS Management Console, yangAWS SAMantarmuka baris perintah (AWS SAMCLI), atauAWSSDK. Untuk mempelajari lebih lanjut tentang prosedur penerbitan aplikasi keAWS Serverless Application Repository, lihat[Cara Memublikasikan Aplikasi](#).

Ketika Anda memublikasikan aplikasi Anda, itu awalnya diatur keprivat, yang berarti bahwa itu hanya tersedia untukAWSakun yang membuatnya. Untuk berbagi aplikasi Anda dengan orang lain, Anda harus mengaturnyadibagikan secara pribadi(hanya dibagi dengan set tertentuAWSakun), atauberbagi secara publik(dibagikan dengan semua orang).

Ketika Anda memublikasikan aplikasi keAWS Serverless Application Repositorydan mengaturnya ke publik, layanan ini membuat aplikasi tersedia untuk konsumen di semua Wilayah. Ketika konsumen menyebarkan aplikasi publik ke Wilayah selain Wilayah di mana aplikasi pertama kali diterbitkan, yangAWS Serverless Application Repositoriesalinan artefak penyebaran aplikasi ke bucket Amazon S3 di Wilayah tujuan. Ini memperbarui sumber daya apa pun diAWS SAMtemplate yang menggunakan artefak tersebut sebagai gantinya referensi file dalam bucket Amazon S3 untuk Wilayah tujuan. Artefak penyebaran dapat mencakup kode fungsi Lambda, file definisi API, dan sebagainya.

Note

Privatdandibagikan secara pribadiaplikasi hanya tersedia diAWSWilayah bahwa mereka diciptakan di. Berbagi secara publikaplikasi yang tersedia di semuaAWSWilayah. Untuk mempelajari selengkapnya tentang berbagi aplikasi, lihat[AWS Serverless Application RepositoryContoh Kebijakan Aplikasi](#).

Topik

- [Menggunakan AWS SAM dengan AWS Serverless Application Repository](#)
- [Cara Memublikasikan Aplikasi](#)
- [Lencana Penulis Terverifikasi](#)
- [Lapisan Lambda](#)

Menggunakan AWS SAM dengan AWS Serverless Application Repository

AWS Serverless Application Model (AWS SAM) adalah kerangka kerja sumber terbuka yang dapat Anda gunakan untuk membangun [aplikasi nirserver](#) pada AWS. Untuk informasi lebih lanjut tentang penggunaan AWS SAM untuk membangun aplikasi tanpa server Anda, lihat [AWS Serverless Application Model Panduan Pengembang](#).

Saat membangun aplikasi yang akan dipublikasikan ke AWS Serverless Application Repository, Anda harus mempertimbangkan set didukung AWS Sumber daya dan Template Kebijakan tersedia untuk digunakan. Bagian di bawah mendeskripsikan topik ini secara lebih mendetail.

Didukung AWS Sumber Daya di AWS Serverless Application Repository

Parameter AWS Serverless Application Repository mendukung aplikasi tanpa server yang terdiri dari banyak AWS SAM dan AWS CloudFormation sumber daya. Untuk melihat daftar lengkap AWS sumber daya yang didukung oleh AWS Serverless Application Repository, lihat [Daftar Didukung AWS Sumber Daya](#).

Jika Anda ingin meminta dukungan untuk tambahan AWS sumber daya, kontak [AWS Dukungan](#).

Important

Jika template aplikasi Anda berisi salah satu peran IAM khusus berikut atau kebijakan sumber daya, aplikasi Anda tidak muncul di hasil pencarian secara default. Juga, pelanggan perlu mengakui peran IAM kustom aplikasi atau kebijakan sumber daya sebelum mereka dapat menerapkan aplikasi. Untuk informasi selengkapnya, lihat [Kemampuan Aplikasi](#). Daftar sumber daya yang berlaku ini adalah:

- Peran IAM: [AWS::IAM::Group](#), [AWS::IAM::InstanceProfile](#), [AWS::IAM::Policy](#), dan [AWS::IAM::Peran](#).

- Kebijakan sumber daya: [AWS::Lambda::LayerVersionIzin](#), [AWS::Lambda::Permission](#), [AWS::Events::EventBusKebijakan](#), [AWS::IAM::Kebijakan](#), [AWS::ApplicationAutoMenskalakan::ScalingPolicy](#), [AWS::S3::BucketPolicy](#), [AWS::SQS::QueuePolicy](#), dan [AWS::SNS::TopicPolicy](#).

Jika aplikasi Anda berisi [AWS::Serverless::Application](#) sumber daya, pelanggan perlu mengakui bahwa aplikasi bersiarang sebelum mereka dapat menyebarkan aplikasi. Untuk informasi selengkapnya tentang aplikasi bersiarang, lihat [Aplikasi Bersiarang](#) di [AWS Serverless Application Model Panduan Pengembang](#). Untuk informasi selengkapnya tentang kemampuan pengakuan, lihat [Kemampuan Aplikasi](#).

Templat Kebijakan

AWS SAM menyediakan daftar templat kebijakan untuk ruang lingkup izin fungsi Lambda Anda pada sumber daya yang digunakan oleh aplikasi Anda. Menggunakan template kebijakan tidak memerlukan pengakuan pelanggan tambahan untuk mencari, menelusuri, atau menyebarkan aplikasi.

Untuk daftar standar AWS SAM Templat kebijakan, lihat [AWS SAM Templat Kebijakan](#) di [AWS Serverless Application Model Panduan Pengembang](#).

Daftar Didukung AWS Sumber Daya

Ini adalah daftar lengkap AWS sumber daya yang didukung oleh AWS Serverless Application Repository.

- `AWS::AccessAnalyzer::Analyzer`
- `AWS::AmazonMQ::Broker`
- `AWS::AmazonMQ::Configuration`
- `AWS::AmazonMQ::ConfigurationAssociation`
- `AWS::ApiGateway::Account`
- `AWS::ApiGateway::ApiKey`
- `AWS::ApiGateway::Authorizer`
- `AWS::ApiGateway::BasePathMapping`

- `AWS::ApiGateway::ClientCertificate`
- `AWS::ApiGateway::Deployment`
- `AWS::ApiGateway::DocumentationPart`
- `AWS::ApiGateway::DocumentationVersion`
- `AWS::ApiGateway::DomainName`
- `AWS::ApiGateway::GatewayResponse`
- `AWS::ApiGateway::Method`
- `AWS::ApiGateway::Model`
- `AWS::ApiGateway::RequestValidator`
- `AWS::ApiGateway::Resource`
- `AWS::ApiGateway::RestApi`
- `AWS::ApiGateway::Stage`
- `AWS::ApiGateway::UsagePlan`
- `AWS::ApiGateway::UsagePlanKey`
- `AWS::ApiGateway::VpcLink`
- `AWS::ApiGatewayV2::Api`
- `AWS::ApiGatewayV2::ApiMapping`
- `AWS::ApiGatewayV2::Authorizer`
- `AWS::ApiGatewayV2::DomainName`
- `AWS::ApiGatewayV2::Deployment`
- `AWS::ApiGatewayV2::Integration`
- `AWS::ApiGatewayV2::IntegrationResponse`
- `AWS::ApiGatewayV2::Model`
- `AWS::ApiGatewayV2::Route`
- `AWS::ApiGatewayV2::RouteResponse`
- `AWS::ApiGatewayV2::Stage`
- `AWS::AppSync::ApiKey`
- `AWS::AppSync::DataSource`
- `AWS::AppSync::GraphQLApi`
- `AWS::AppSync::GraphQLSchema`

- `AWS::AppSync::Resolver`
- `AWS::ApplicationAutoScaling::AutoScalingGroup`
- `AWS::ApplicationAutoScaling::LaunchConfiguration`
- `AWS::ApplicationAutoScaling::ScalableTarget`
- `AWS::ApplicationAutoScaling::ScalingPolicy`
- `AWS::Athena::NamedQuery`
- `AWS::Athena::WorkGroup`
- `AWS::CertificateManager::Certificate`
- `AWS::Chatbot::SlackChannelConfiguration`
- `AWS::CloudFormation::CustomResource`
- `AWS::CloudFormation::Interface`
- `AWS::CloudFormation::Macro`
- `AWS::CloudFormation::WaitConditionHandle`
- `AWS::CloudFront::CachePolicy`
- `AWS::CloudFront::CloudFrontOriginAccessIdentity`
- `AWS::CloudFront::Distribution`
- `AWS::CloudFront::Function`
- `AWS::CloudFront::OriginRequestPolicy`
- `AWS::CloudFront::ResponseHeadersPolicy`
- `AWS::CloudFront::StreamingDistribution`
- `AWS::CloudTrail::Trail`
- `AWS::CloudWatch::Alarm`
- `AWS::CloudWatch::AnomalyDetector`
- `AWS::CloudWatch::Dashboard`
- `AWS::CloudWatch::InsightRule`
- `AWS::CodeBuild::Project`
- `AWS::CodeCommit::Repository`
- `AWS::CodePipeline::CustomActionType`
- `AWS::CodePipeline::Pipeline`
- `AWS::CodePipeline::Webhook`

- `AWS::CodeStar::GitHubRepository`
- `AWS::CodeStarNotifications::NotificationRule`
- `AWS::Cognito::IdentityPool`
- `AWS::Cognito::IdentityPoolRoleAttachment`
- `AWS::Cognito::UserPool`
- `AWS::Cognito::UserPoolClient`
- `AWS::Cognito::UserPoolDomain`
- `AWS::Cognito::UserPoolGroup`
- `AWS::Cognito::UserPoolResourceServer`
- `AWS::Cognito::UserPoolUser`
- `AWS::Cognito::UserPoolUserToGroupAttachment`
- `AWS::Config::AggregationAuthorization`
- `AWS::Config::ConfigRule`
- `AWS::Config::ConfigurationAggregator`
- `AWS::Config::ConfigurationRecorder`
- `AWS::Config::DeliveryChannel`
- `AWS::Config::RemediationConfiguration`
- `AWS::DataPipeline::Pipeline`
- `AWS::DynamoDB::Table`
- `AWS::EC2::EIP`
- `AWS::EC2::InternetGateway`
- `AWS::EC2::NatGateway`
- `AWS::EC2::Route`
- `AWS::EC2::RouteTable`
- `AWS::EC2::SecurityGroup`
- `AWS::EC2::SecurityGroupEgress`
- `AWS::EC2::SecurityGroupIngress`
- `AWS::EC2::Subnet`
- `AWS::EC2::SubnetRouteTableAssociation`
- `AWS::EC2::VPC`

- `AWS::EC2::VPCGatewayAttachment`
- `AWS::EC2::VPCPeeringConnection`
- `AWS::ECR::Repository`
- `AWS::Elasticsearch::Domain`
- `AWS::Events::EventBus`
- `AWS::Events::EventBusPolicy`
- `AWS::Events::Rule`
- `AWS::EventSchemas::Discoverer`
- `AWS::EventSchemas::Registry`
- `AWS::EventSchemas::Schema`
- `AWS::Glue::Classifier`
- `AWS::Glue::Connection`
- `AWS::Glue::Crawler`
- `AWS::Glue::Database`
- `AWS::Glue::DevEndpoint`
- `AWS::Glue::Job`
- `AWS::Glue::Partition`
- `AWS::Glue::SecurityConfiguration`
- `AWS::Glue::Table`
- `AWS::Glue::Trigger`
- `AWS::Glue::Workflow`
- `AWS::IAM::Group`
- `AWS::IAM::InstanceProfile`
- `AWS::IAM::ManagedPolicy`
- `AWS::IAM::OIDCProvider`
- `AWS::IAM::Policy`
- `AWS::IAM::Role`
- `AWS::IAM::ServiceLinkedRole`
- `AWS::IoT::Certificate`
- `AWS::IoT::Policy`

- `AWS::IoT::PolicyPrincipalAttachment`
- `AWS::IoT::Thing`
- `AWS::IoT::ThingPrincipalAttachment`
- `AWS::IoT::TopicRule`
- `AWS::KMS::Alias`
- `AWS::KMS::Key`
- `AWS::Kinesis::Stream`
- `AWS::Kinesis::StreamConsumer`
- `AWS::Kinesis::Streams`
- `AWS::KinesisAnalytics::Application`
- `AWS::KinesisAnalytics::ApplicationOutput`
- `AWS::KinesisFirehose::DeliveryStream`
- `AWS::Lambda::Alias`
- `AWS::Lambda::EventInvokeConfig`
- `AWS::Lambda::EventSourceMapping`
- `AWS::Lambda::Function`
- `AWS::Lambda::LayerVersion`
- `AWS::Lambda::LayerVersionPermission`
- `AWS::Lambda::Permission`
- `AWS::Lambda::Version`
- `AWS::Location::GeofenceCollection`
- `AWS::Location::Map`
- `AWS::Location::PlaceIndex`
- `AWS::Location::RouteCalculator`
- `AWS::Location::Tracker`
- `AWS::Location::TrackerConsumer`
- `AWS::Logs::Destination`
- `AWS::Logs::LogGroup`
- `AWS::Logs::LogStream`
- `AWS::Logs::MetricFilter`

- `AWS::Logs::SubscriptionFilter`
- `AWS::Route53::HealthCheck`
- `AWS::Route53::HostedZone`
- `AWS::Route53::RecordSet`
- `AWS::Route53::RecordSetGroup`
- `AWS::S3::Bucket`
- `AWS::S3::BucketPolicy`
- `AWS::SNS::Subscription`
- `AWS::SNS::Topic`
- `AWS::SNS::TopicPolicy`
- `AWS::SQS::Queue`
- `AWS::SQS::QueuePolicy`
- `AWS::SSM::Association`
- `AWS::SSM::Document`
- `AWS::SSM::MaintenanceWindowTask`
- `AWS::SSM::Parameter`
- `AWS::SSM::PatchBaseline`
- `AWS::SSM::ResourceDataSync`
- `AWS::SecretsManager::ResourcePolicy`
- `AWS::SecretsManager::RotationSchedule`
- `AWS::SecretsManager::Secret`
- `AWS::SecretsManager::SecretTargetAttachment`
- `AWS::Serverless::Api`
- `AWS::Serverless::Application`
- `AWS::Serverless::Function`
- `AWS::Serverless::HttpApi`
- `AWS::Serverless::LayerVersion`
- `AWS::Serverless::SimpleTable`
- `AWS::Serverless::StateMachine`
- `AWS::ServiceDiscovery::HttpNamespace`

- `AWS::ServiceCatalog::CloudFormationProvisionedProduct`
- `AWS::ServiceDiscovery::Instance`
- `AWS::ServiceDiscovery::PrivateDnsNamespace`
- `AWS::ServiceDiscovery::PublicDnsNamespace`
- `AWS::ServiceDiscovery::Service`
- `AWS::SES::ReceiptRule`
- `AWS::SES::ReceiptRuleSet`
- `AWS::StepFunctions::Activity`
- `AWS::StepFunctions::StateMachine`
- `AWS::Wisdom::Assistant`
- `AWS::Wisdom::AssistantAssociation`
- `AWS::Wisdom::KnowledgeBase`

Cara Mempublikasikan Aplikasi

Bagian ini memberi Anda prosedur untuk menerbitkan aplikasi tanpa server Anda keAWS Serverless Application Repositorydengan menggunakanAWS SAMCLI atauAWS Management Console. Ini juga menunjukkan cara membagikan aplikasi Anda untuk memungkinkan orang lain menerapkannya, dan menghapus aplikasi Anda dariAWS Serverless Application Repository.

Important

Informasi yang Anda masukkan saat memublikasikan aplikasi tidak dienkripsi. Informasi ini mencakup data seperti nama penulis. Jika Anda memiliki informasi identitas pribadi yang tidak ingin disimpan atau dipublikasikan, sebaiknya Anda tidak memasukkan informasi ini saat menerbitkan aplikasi Anda.

Menerbitkan Aplikasi (AWS CLI)

Cara termudah untuk memublikasikan aplikasi keAWS Serverless Application Repositoryadalah dengan menggunakan satu setAWS SAMPerintah CLI. Untuk informasi selengkapnya, lihat[Menerbitkan Aplikasi MenggunakanAWS SAMCLI](#)diAWS Serverless Application Model(AWS SAMPanduan Developer.

Menerbitkan Aplikasi Baru (Konsol)

Bagian ini menunjukkan cara menggunakan AWS Management Console untuk mempublikasikan aplikasi baru ke AWS Serverless Application Repository. Untuk informasi tentang mempublikasikan versi baru aplikasi yang sudah ada, lihat [Menerbitkan Versi Baru dari Aplikasi yang Ada](#).

Prasyarat

Sebelum Anda mempublikasikan aplikasi ke AWS Serverless Application Repository, Anda memerlukan yang berikut ini:

- Valid AWS akun.
- Valid AWS Serverless Application Model (AWS SAM) template yang mendefinisikan AWS sumber daya yang digunakan. Untuk informasi lebih lanjut tentang AWS SAM template, lihat [AWS SAM Dasar-Dasar Template](#).
- Sebuah paket untuk aplikasi Anda yang Anda buat dengan menggunakan AWS CloudFormation package perintah untuk AWS CLI. Perintah ini paket artefak lokal (jalur lokal) yang Anda AWS SAM referensi template. Untuk detail selengkapnya, lihat [paket](#) di AWS CloudFormation dokumentasi.
- URL yang menunjuk ke kode sumber aplikasi Anda, jika Anda ingin mempublikasikan aplikasi Anda secara publik.
- Sebuah file readme.txt. File ini harus menjelaskan cara pelanggan dapat menggunakan aplikasi Anda, dan cara mengonfigurasinya sebelum men-deploynya sendiri AWS akun.
- Sebuah file license.txt atau pengenalan lisensi yang valid dari [Situs SPDX](#). Perhatikan bahwa lisensi hanya diperlukan jika Anda ingin membagikan aplikasi Anda secara publik. Jika Anda akan menjaga aplikasi Anda tetap pribadi atau hanya membagikannya secara pribadi, Anda tidak perlu menentukan lisensi.
- Kebijakan bucket Amazon S3 yang memberikan layanan izin baca untuk artefak yang diunggah ke Amazon S3 ketika Anda mengemas aplikasi Anda. Untuk mengatur kebijakan ini, ikuti langkah-langkah ini:
 1. Buka konsol Amazon S3 di <https://console.aws.amazon.com/s3/>.
 2. Pilih bucket Amazon S3 yang Anda gunakan untuk mengemas aplikasi Anda.
 3. Pilih tab Izin.
 4. Pilih Kebijakan embertombol.
 5. Tempelkan pernyataan kebijakan berikut ke Editor kebijakan bucket. Pastikan untuk mengganti nama bucket Anda di Resource elemen, dan AWSID akun di Condition elemen. Ekspresi

dalam `Condition` elemen memastikan AWS Serverless Application Repository hanya memiliki izin untuk mengakses aplikasi dari yang ditentukan AWS S3. Untuk informasi selengkapnya tentang pernyataan kebijakan, lihat [Referensi elemen kebijakan JSON IAM](#) di Panduan Pengguna IAM.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "serverlessrepo.amazonaws.com"
      },
      "Action": "s3:GetObject",
      "Resource": "arn:aws:s3:::bucketname/*",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "123456789012"
        }
      }
    }
  ]
}
```

6. Pilih tombol Simpan.

Prosedur

Membuat aplikasi baru di AWS Serverless Application Repository dengan menggunakan prosedur berikut.

Untuk membuat aplikasi baru di AWS Serverless Application Repository

1. Buka [AWS Serverless Application Repository konsol](#) dan pilih **Memublikasikan aplikasi**.
2. Pada **Publikasikan aplikasi** halaman, masukkan informasi aplikasi berikut, lalu pilih **Memublikasikan aplikasi**:

Properti	Diperlukan	Deskripsi
Nama aplikasi	BENAR	Nama aplikasi.

Properti	Diperlukan	Deskripsi
		<p>Panjang minimum=1. Panjang maksimum=140.</p> <p>Pola: “[a-zA-Z0-9\ -] +”;</p>
Penulis	BETUL	<p>Nama penulis yang memublikasikan aplikasi.</p> <p>Panjang minimum=1. Panjang maksimum=127.</p> <p>Pola: “^ [a-z0-9] ([a-z0-9] -(?! -) * [a-z0-9])? \$”;</p>
Halaman Beranda	SALAH	<p>URL dengan informasi selengkapnya tentang aplikasi—misalnya, lokasi Anda GitHub repositori untuk aplikasi.</p>
Deskripsi	BETUL	<p>Deskripsi aplikasi.</p> <p>Panjang minimum=1. Panjang maksimum=256.</p>
Label	SALAH	<p>Label yang meningkatkan penemuan aplikasi di hasil pencarian.</p> <p>Panjang minimum=1. Panjang maksimum=127. Jumlah maksimum label: 10.</p> <p>Pola: “^ [a-zA-Z0-9+\ -_:\ \ /@] +\$”;</p>

Properti	Diperlukan	Deskripsi
Lisensi SPDX (daftar drop-down)	SALAH	Pilih pengenal lisensi yang valid dari drop-down yang berisi lisensi yang tersedia di Situs SPDX . Memilih item di drop-down Lisensi kotak teks di bawahnya. Catatan: Memilih lisensi di drop-down menggantikan isi Lisensi kotak teks, dan membuang suntingan manual yang telah Anda buat.
Lisensi	SALAH	<p>Unggah file lisensi.txt, atau pilih lisensi dari lisensi SPDX drop-down dijelaskan di baris sebelumnya. Memilih lisensi dari lisensi SPDX drop-down secara otomatis mengisi Lisensi kotak teks. Anda dapat mengedit isi kotak teks ini secara manual setelah mengunggah file lisensi atau memilih salah satu dari lisensi SPDX drop-down. Namun, jika lain lisensi SPDX dipilih dari drop-down, setiap suntingan manual yang telah Anda buat dibuang.</p> <p>Ini adalah bidang opsional, tetapi Anda harus memberikan lisensi untuk berbagi aplikasi secara publik.</p>

Properti	Diperlukan	Deskripsi
Readme	SALAH	Upload isi file Readme, yang dapat dalam format teks atau penurunan harga. Isi ini ditampilkan pada halaman detail aplikasi di AWS Serverless Application Repository. Anda dapat mengedit isi kotak teks ini secara manual setelah mengunggah file.
Versi semantik	SALAH	Versi semantik aplikasi. Untuk informasi selengkapnya, lihat Situs Semantic Versioning . Anda harus memberikan nilai untuk properti ini agar dapat membuat aplikasi Anda publik.
Url kode sumber	SALAH	Tautan ke repositori publik untuk kode sumber aplikasi Anda.
Templat SAM	BETUL	Valid AWS Serverless Application Model (AWS SAM) template yang mendefinisikan AWS sumber daya yang digunakan.

Berbagi Aplikasi

Aplikasi yang diterbitkan dapat memiliki izin yang ditetapkan dalam salah satu dari tiga kategori berikut:

- **Pribadi (default)**— Aplikasi yang dibuat dengan akun yang sama, dan belum dibagikan dengan yang lainAWSakun. Hanya konsumen yang berbagiAWSakun memiliki izin untuk menyebarkan aplikasi pribadi.
- **Dibagikan secara pribadi**— Aplikasi yang telah dibagikan oleh penerbit secara eksplisit dengan satu set tertentuAWSakun, atau denganAWSakun dalamAWSOrganisasi. Konsumen memiliki izin untuk menyebarkan aplikasi yang telah dibagikan kepada pelangganAWSakun atauAWSOrganisasi. Untuk informasi selengkapnya tentang AWS Organizations, lihat [Panduan Pengguna AWS Organizations](#).
- **Dibagikan publik**— Aplikasi yang telah dibagikan oleh penerbit dengan semua orang. Semua konsumen memiliki izin untuk menyebarkan aplikasi yang dibagikan secara publik.

Setelah Anda menerbitkan aplikasi keAWS Serverless Application Repository, secara default diatur keprivat. Bagian ini menunjukkan cara untuk membagikan aplikasi secara pribadi dengan spesifikAWSakun atauAWSOrganisasi, atau berbagi secara terbuka dengan semua orang.

Berbagi Aplikasi Melalui Konsol

Anda memiliki dua opsi untuk membagikan aplikasi Anda dengan orang lain: 1) Bagikan dengan spesifikAWSakun atauAWSakun dalamAWSOrganisasi, atau 2) Bagikan secara terbuka dengan semua orang. Untuk informasi selengkapnya tentang AWS Organizations, lihat [Panduan Pengguna AWS Organizations](#).

Opsi 1: Untuk berbagi aplikasi Anda dengan spesifikAWSakun atau akun dalamAWSOrganisasi

1. Buka [konsol AWS Serverless Application Repository](#).
2. Pada panel navigasi, pilihAplikasi yang dipublikasikanuntuk memunculkan daftar aplikasi yang telah Anda buat.
3. Pilih aplikasi yang ingin Anda bagikan.
4. PilihPembagiantab.
5. DiPernyataan kebijakan aplikasibagian, pilihMembuat Pernyataantombol.
6. DiKonfigurasi pernyataanjendela mengisi bidang berdasarkan bagaimana Anda ingin berbagi aplikasi Anda.

Note

Jika Anda berbagi dengan organisasi, Anda hanya dapat menentukan organisasi yang AndaAWSakun adalah anggota dari. Jika Anda mencoba untuk

menentukan AWS Organisasi bahwa Anda bukan anggota, kesalahan akan mengakibatkan.

Untuk berbagi aplikasi Anda dengan AWS Organisasi, Anda harus mengakui bahwa `UnshareApplication` tindakan akan ditambahkan ke pernyataan kebijakan Anda, jika berbagi perlu dicabut di future.

7. Pilih tombol Simpan.

Opsi 2: Untuk berbagi aplikasi Anda secara publik dengan semua orang

1. Buka [konsol AWS Serverless Application Repository](#).
2. Pada panel navigasi, pilih Aplikasi yang dipublikasikan untuk memunculkan daftar aplikasi yang telah Anda buat.
3. Pilih aplikasi yang ingin Anda bagikan.
4. Pilih Pembagian tab.
5. Di Berbagi publik bagian, pilih Mengedit tombol.
6. Di bawah Berbagi publik pilih Diaktifkan Tombol radio.
7. Di kotak teks ketik nama aplikasi Anda, lalu pilih Simpan tombol.

Note

Dalam rangka untuk berbagi aplikasi publik, itu harus memiliki kedua `SemanticVersion` dan `LicenseUrl` set properti.

Berbagi Aplikasi Melalui AWS CLI

Untuk berbagi aplikasi menggunakan AWS CLI Anda memberikan izin menggunakan `put-application-policy` perintah untuk menentukan AWS Akun yang ingin Anda bagikan sebagai prinsipal.

Untuk informasi lebih lanjut tentang berbagi aplikasi Anda dengan menggunakan AWS CLI, lihat [AWS Serverless Application Repository Contoh Kebijakan Aplikasi](#).

Membatalkan Berbagi Aplikasi

Ada dua opsi untuk unsharing aplikasi dari AWS Organisasi:

1. Penerbit aplikasi dapat menghapus izin menggunakan [put-application-policy](#) perintah.
2. Seorang pengguna dari akun pengelola dari AWS Organisasi dapat melakukan [aplikasi unshare](#) operasi pada setiap aplikasi bersama dengan organisasi, bahkan jika aplikasi itu diterbitkan oleh pengguna dari akun yang berbeda.

Note

Ketika aplikasi tidak dibagikan dari AWS Organisasi dengan operasi “unshare application”, tidak dapat dibagi dengan AWS Organisasi lagi.

Untuk informasi selengkapnya tentang AWS Organizations, lihat [Panduan Pengguna AWS Organizations](#).

Publikasi Menghapus Izin

Penerbit Menghapus Izin Melalui Konsol

Untuk unshare aplikasi melalui AWS Management Console, Anda menghapus pernyataan kebijakan yang berbagi dengan yang lain AWS Akun. Untuk melakukannya, ikuti langkah-langkah berikut:

1. Buka [konsol AWS Serverless Application Repository](#).
2. Pilih Aplikasi yang tersedia Di panel navigasi kiri.
3. Pilih aplikasi yang ingin Anda bagikan.
4. Pilih Pembagian tab.
5. Di Pernyataan kebijakan aplikasi bagian, pilih pernyataan kebijakan yang berbagi aplikasi dengan akun yang ingin Anda unshare dari.
6. Pilih Delete (Hapus).
7. Sebuah pesan konfirmasi akan muncul. Pilih Hapus lagi.

Penerbit Menghapus Izin Melalui AWS CLI

Untuk unshare aplikasi melalui AWS CLI, penerbit dapat menghapus atau mengubah izin menggunakan `put-application-policy` perintah untuk membuat aplikasi pribadi, atau berbagi dengan set yang berbeda AWS Akun.

Untuk informasi lebih lanjut tentang perubahan izin menggunakan AWS CLI, lihat [AWS Serverless Application Repository Contoh Kebijakan Aplikasi](#).

Akun manajemen unsharing aplikasi

Akun manajemen unsharing aplikasi dari AWS Organisasi melalui konsol

Untuk membatalkan berbagi aplikasi dari AWS Organisasi melalui AWS Management Console, pengguna dari akun pengelola dapat melakukan hal berikut:

1. Buka [konsol AWS Serverless Application Repository](#).
2. Pilih Aplikasi yang tersedia Di panel navigasi kiri.
3. Dalam ubin aplikasi, pilih Tidak berbagi.
4. Dalam kotak pesan unshare, konfirmasi Anda ingin membatalkan berbagi aplikasi dengan memasukkan ID Organisasi dan nama aplikasi, kemudian memilih Simpan.

Akun manajemen unsharing aplikasi dari AWS Organisasi Melalui AWS CLI

Untuk membatalkan berbagi aplikasi dari AWS Organisasi, pengguna dari akun pengelola dapat menjalankan `aws serverlessrepo unshare-application` perintah.

Perintah berikut unshares aplikasi dari AWS Organisasi, tempat *id aplikasi* Amazon Resource Name (ARN), dan *organisasi-id* adalah AWS ID Organisasi:

```
aws serverlessrepo unshare-application --application-id application-id --organization-id organization-id
```

Menghapus Aplikasi

Anda dapat menghapus aplikasi dari AWS Serverless Application Repository dengan menggunakan salah satu AWS Management Console atau AWS SAM CLI.

Menghapus Aplikasi (Konsol)

Untuk menghapus aplikasi yang dipublikasikan melalui AWS Management Console, lakukan hal berikut.

1. Buka [konsol AWS Serverless Application Repository](#).
2. Untuk Aplikasi saya, pilih aplikasi yang ingin Anda hapus.
3. Di halaman detail aplikasi, pilih Menghapus aplikasi.
4. Pilih Menghapus aplikasi untuk menyelesaikan penghapusan.

Menghapus Aplikasi (AWS CLI)

Untuk menghapus aplikasi yang dipublikasikan menggunakan AWS CLI, jalankan [aws serverlessrepo delete-application](#) perintah.

Perintah berikut akan menghapus aplikasi, tempat *application-id* adalah Amazon Resource Name (ARN) aplikasi:

```
aws serverlessrepo delete-application --application-id application-id
```

Menerbitkan Versi Baru dari Aplikasi yang Ada

Bagian ini menunjukkan cara untuk mempublikasikan versi baru aplikasi yang sudah ada ke AWS Serverless Application Repository dengan menggunakan AWS SAM CLI atau AWS Management Console. Untuk petunjuk tentang penerbitan aplikasi baru, lihat [Cara Mempublikasikan Aplikasi](#).

Menerbitkan Versi Baru dari Aplikasi yang Ada (AWS CLI)

Cara termudah untuk mempublikasikan versi baru dari aplikasi yang ada adalah dengan menggunakan satu set AWS SAM Perintah CLI. Untuk informasi selengkapnya, lihat [Menerbitkan Aplikasi Menggunakan AWS SAM CLI](#) di AWS Serverless Application Model (AWS SAM) Panduan Developer.

Menerbitkan Versi Baru dari Aplikasi yang Ada (Konsol)

Untuk mempublikasikan versi baru aplikasi yang telah Anda publikasikan sebelumnya, ikuti langkah-langkah berikut:

1. Buka [konsol AWS Serverless Application Repository](#).
2. Di panel navigasi, pilih Aplikasi saya untuk memunculkan daftar aplikasi yang telah Anda buat.
3. Pilih aplikasi yang ingin Anda terbitkan untuk versi baru.
4. Pilih Publikasikan versi baru.
5. Masuk Versi, masukkan informasi aplikasi berikut:

Properti	Diperlukan	Deskripsi
Versi semantik	BETUL	Versi semantik aplikasi. Untuk informasi selengkapnya, lihat Situs Semantic Versioning . Anda harus memberikan nilai untuk properti ini agar dapat membuat aplikasi Anda publik.
Url kode sumber	SALAH	Tautan ke repositori publik untuk kode sumber aplikasi Anda.
Templat SAM	BETUL	Valid AWS Serverless Application Model (AWS SAM) template yang mendefinisikan AWS sumber daya yang digunakan.

6. Pilih Publikasikan versi.

Lencana Penulis Terverifikasi

Penulis terverifikasi di AWS Serverless Application Repository adalah orang-orang yang AWS telah membuat tinjauan itikad baik, sebagai penyedia layanan yang masuk akal dan bijaksana, informasi yang diberikan oleh pemohon, dan telah mengkonfirmasi bahwa identitas pemohon adalah seperti yang diklaim.

Aplikasi penulis terverifikasi menampilkan lencana penulis yang diverifikasi, bersama dengan tautan ke profil publik penulis. Lencana penulis diverifikasi ditampilkan di kedua hasil pencarian dan pada halaman detail aplikasi.

Meminta Lencana Penulis Terverifikasi

Anda dapat meminta untuk disetujui sebagai penulis terverifikasi diAWS Serverless Application Repositorydengan mengirim email keserverlessrepo-verified-author@amazon.com. Anda perlu memberikan informasi berikut ini:

- Nama penulis
- ID akun AWS
- Tautan profil yang dapat diakses publik, sepertiGitHubatauLinkedInprofil

Setelah mengirimkan permintaan untuk lencana penulis terverifikasi, Anda dapat mengharapkan tanggapan dariAWSdalam beberapa hari. Anda mungkin diminta untuk informasi tambahan sebelum permintaan Anda disetujui.

Setelah permintaan Anda disetujui, Anda dapat mengharapkan bahwa lencana penulis terverifikasi akan ditampilkan untuk aplikasi Anda dalam sehari.

Note

Lencana penulis diverifikasi ditampilkan untuk semua aplikasi yang cocok dengan keduaAWSakun dan nama penulis. KarenaAWSakun dapat memiliki beberapa penulis, lencana tidak ditampilkan pada aplikasi yang memiliki nama penulis yang berbeda. Untuk memiliki lencana penulis ditampilkan pada aplikasi dengan nama penulis yang berbeda, Anda harus mengirimkan permintaan lain untuk penulis itu.

Lapisan Lambda

Jika Anda telah menerapkan fungsionalitas di layer Lambda, Anda mungkin ingin berbagi layer Anda tanpa hosting instance global dari itu. Berbagi layer dengan cara ini memungkinkan orang lain untuk menyebarkan instance layer Anda ke akun mereka sendiri. Hal ini mencegah aplikasi klien tergantung pada instance global layer Anda. ParameterAWS Serverless Application Repositorymemungkinkan Anda untuk berbagi layer Lambda dengan cara ini dengan mudah.

Untuk informasi selengkapnya tentang lapisan Lambda, lihat [AWS Lambda Lapisan](#) di [AWS Lambda Panduan Pengembang](#).

Cara Kerjanya

Berikut adalah langkah-langkah untuk berbagi lapisan Anda menggunakan AWS Serverless Application Repository. Hal ini memungkinkan salinan layer Anda untuk dibuat di pengguna AWS akun.

1. Mendefinisikan aplikasi nirserver dengan AWS SAM template yang mencakup layer Anda sebagai resource— yaitu, baik [AWS::Serverless::LayerVersion](#) atau [AWS::Lambda::LayerVersion](#) sumber daya.
2. Publikasikan aplikasi Anda ke AWS Serverless Application Repository, dan membaginya (baik secara terbuka maupun pribadi).
3. Pelanggan menyebarkan aplikasi Anda, yang membuat salinan layer Anda sendiri AWS akun. Pelanggan sekarang dapat referensi Amazon Resource Name (ARN) dari lapisan di dalamnya AWS akun dalam aplikasi klien mereka.

Contoh

Berikut ini adalah contoh-contohnya AWS SAM template untuk aplikasi yang berisi layer Lambda yang ingin Anda bagikan:

```
Resources:
  SharedLayer:
    Type: AWS::Serverless::LayerVersion
    Properties:
      LayerName: shared-layer
      ContentUri: source/layer-code/
      CompatibleRuntimes:
        - python3.7
Outputs:
  LayerArn:
    Value: !Ref SharedLayer
```

Ketika pelanggan menyebarkan aplikasi Anda dari AWS Serverless Application Repository, sebuah layer dibuat di AWS akun. Lapisan ARN terlihat seperti berikut ini:

```
arn:aws:lambda:us-east-1:012345678901:layer:shared-layer:1
```

Pelanggan sekarang dapat referensi ARN ini dalam aplikasi klien mereka sendiri, seperti dalam contoh ini:

Resources:**MyFunction:**

Type: `AWS::Serverless::Function`

Properties:

Handler: `index.handler`

Runtime: `python3.7`

CodeUrl: `source/app-code/`

Layers:

- `arn:aws:lambda:us-east-1:012345678901:layer:shared-layer:1`

Men-deploy Aplikasi

Bagian ini membantu Anda mempelajari cara menemukan dan menyebarkan aplikasi tanpa server yang telah dipublikasikan ke AWS Serverless Application Repository. Anda dapat menelusuri aplikasi yang tersedia untuk umum tanpa memiliki AWS akun dengan mengunjungi [situs publik](#). Atau, Anda dapat menelusuri aplikasi dari dalam AWS Lambda konsol.

Beberapa aplikasi memiliki penulis terverifikasi silencana, dengan link ke profil penulis. Seorang penulis dianggap sebagai penulis terverifikasi ketika AWS telah membuat tinjauan itikad baik, sebagai penyedia layanan yang masuk akal dan bijaksana, informasi yang diberikan oleh pemohon, dan telah mengkonfirmasi bahwa identitas pemohon adalah seperti yang diklaim.

Sebelum menerapkan aplikasi dari AWS Serverless Application Repository, lihat topik berikut untuk mempelajari tentang izin penyebaran aplikasi dan kemampuan aplikasi.

Topik

- [Izin Deployment Aplikasi](#)
- [Kemampuan Aplikasi: Peran IAM, Kebijakan Sumber Daya, dan Aplikasi Bersarang](#)
- [Cara Menyebarkan Aplikasi](#)

Izin Deployment Aplikasi

Untuk men-deploy aplikasi di AWS Serverless Application Repository, Anda harus memiliki izin untuk melakukannya. Ada tiga kategori aplikasi yang Anda memiliki izin untuk menyebarkan:

- Privat— Aplikasi yang dibuat dengan akun yang sama, dan belum dibagikan dengan akun lain. Anda memiliki izin untuk menyebarkan aplikasi yang dibuat menggunakan AWS akun.
- Dibagikan secara pribadi— Aplikasi yang penerbit telah secara eksplisit berbagi dengan satu set tertentu AWS akun. Anda memiliki izin untuk men-deploy aplikasi yang telah dibagikan kepada Anda AWS akun.
- Dibagikan kepada publik— Aplikasi yang telah dibagikan oleh penerbit dengan semua orang. Anda memiliki izin untuk menyebarkan aplikasi yang dibagikan secara publik.

Anda hanya dapat mencari dan menelusuri aplikasi yang memiliki izin untuk Anda. Ini termasuk aplikasi yang dibuat menggunakan AWS akun, dibagikan secara pribadi dengan AWS akun, dan dibagikan secara publik. Semua aplikasi lain tidak ditampilkan untuk Anda.

⚠ Important

Aplikasi yang berisi aplikasi bersarang mewarisi pembatasan berbagi aplikasi bersarang. Misalnya, anggaplah aplikasi dibagikan secara publik, tetapi berisi aplikasi nest yang hanya dibagikan secara privat dengan akun AWS yang membuat aplikasi induk. Dalam hal ini, jika Anda AWS akun tidak memiliki izin untuk men-deploy aplikasi bersarang, maka Anda tidak dapat men-deploy aplikasi induk. Untuk informasi selengkapnya tentang aplikasi bersarang, lihat [Aplikasi Bersarang](#) di AWS Serverless Application Model Panduan Pengembang.

Kemampuan Aplikasi: Peran IAM, Kebijakan Sumber Daya, dan Aplikasi Bersarang

Sebelum Anda dapat menerapkan aplikasi, AWS Serverless Application Repository memeriksa template aplikasi untuk peran IAM, kebijakan AWS sumber daya, dan aplikasi bersarang yang ditetapkan template yang harus dibuat. Sumber daya IAM, seperti peran IAM dengan akses penuh, dapat memodifikasi sumber daya apa pun di AWS akun Anda. Oleh karena itu, kami menyarankan Anda meninjau izin yang terkait dengan aplikasi sebelum melanjutkan sehingga Anda tidak secara tidak sengaja membuat sumber daya dengan izin yang ditingkatkan. Untuk memastikan bahwa Anda telah melakukannya, Anda harus mengakui bahwa aplikasi berisi kemampuan sebelum AWS Serverless Application Repository dapat menerapkan aplikasi atas nama Anda.

Aplikasi dapat berisi salah satu dari empat kemampuan berikut: `CAPABILITY_IAM`, `CAPABILITY_NAMED_IAM`, `CAPABILITY_RESOURCE_POLICY`, dan `CAPABILITY_AUTO_EXPAND`.

Sumber daya berikut mengharuskan Anda untuk menentukan `CAPABILITY_IAM` atau `CAPABILITY_NAMED_IAM`: [AWS::IAM::Group](#), [AWS::IAM::InstanceProfile](#), [AWS::IAM::Policy](#), dan [AWS::IAM::Role](#). Jika aplikasi berisi IAM dengan nama kustom, Anda harus menentukan `CAPABILITY_NAMED_IAM`. Untuk contoh cara menentukan kemampuan, lihat [Menemukan dan Mengakui Kemampuan Aplikasi \(AWS CLI\)](#).

Sumber daya berikut mengharuskan Anda untuk menentukan `CAPABILITY_RESOURCE_POLICY`: [AWS::Lambda::LayerVersion](#) izin [AWS::Lambda::Permission](#), [AWS::Events::EventBus](#) Kebijakan, [AWS::IAM::Policy](#) [AWS::ApplicationAutoScaling::ScalingPolicy](#), [AWS::S3::Bucket](#) Policy,, [AWS::SQS::QueuePolicy](#), dan [AWS::SNS::TopicPolicy](#).

Aplikasi yang berisi satu atau beberapa aplikasi bersarang mengharuskan Anda untuk menentukan `CAPABILITY_AUTO_EXPAND`. Untuk informasi selengkapnya tentang aplikasi bersarang, lihat [Aplikasi Bersarang](#) di Panduan AWS Serverless Application Model Pengembang.

Menemukan dan Mengakui Kemampuan Aplikasi (Konsol)

Anda dapat menemukan aplikasi yang tersedia AWS Serverless Application Repository di [AWS Serverless Application Repository situs web](#), atau melalui [konsol Lambda \(pada halaman Buat Fungsi di bawah AWS Serverless Application Repository tab\)](#).

Aplikasi yang memerlukan pengakuan kemampuan untuk membuat peran IAM kustom atau kebijakan sumber daya tidak ditampilkan dalam hasil pencarian secara default. Untuk menelusuri aplikasi yang berisi kemampuan ini, Anda harus memilih kotak centang Tampilkan aplikasi yang membuat IAM role khusus atau kebijakan sumber daya.

Anda dapat meninjau kemampuan aplikasi di bawah tab Izin saat Anda memilih aplikasi. Untuk menyebarkan aplikasi, Anda perlu memilih Saya mengakui aplikasi ini menciptakan peran IAM kustom atau kebijakan sumber daya kotak centang. Jika Anda tidak mengakui kemampuan ini, Anda melihat pesan galat ini: Pengakuan diperlukan. Untuk menyebarkan, centang kotak di bagian Konfigurasi parameter aplikasi.

Melihat Kemampuan Aplikasi (AWS CLI)

Untuk melihat kemampuan aplikasi menggunakan aplikasi AWS CLI, Anda memerlukan Amazon Resource Name (ARN). Anda kemudian dapat jalankan perintah berikut ini:

```
aws serverlessrepo get-application \  
--application-id application-arn
```

Properti respons [requiredCapabilities](#) berisi daftar kemampuan aplikasi yang perlu Anda akui sebelum Anda dapat menyebarkan aplikasi. Perhatikan bahwa jika properti [requiredCapabilities](#) kosong, aplikasi tidak memiliki kemampuan yang diperlukan.

Cara Menyebarkan Aplikasi

Bagian ini menyediakan Anda dengan prosedur untuk menyebarkan aplikasi tanpa server dari AWS Serverless Application Repository dengan menggunakan AWS Management Console atau AWS CLI.

Menerapkan Aplikasi Baru (Konsol)

Bagian ini menunjukkan bagaimana untuk men-deploy aplikasi baru dari AWS Serverless Application Repository menggunakan AWS Management Console. Untuk petunjuk tentang men-deploy versi baru aplikasi yang sudah ada, lihat [Memperbarui Aplikasi](#).

Menjelajahi, Mencari, dan Menerapkan Aplikasi

Temukan, konfigurasi, dan deploy aplikasi di AWS Serverless Application Repository dengan menggunakan prosedur berikut.

Untuk menemukan dan mengkonfigurasi aplikasi di AWS Serverless Application Repository

1. Buka [halaman beranda AWS Serverless Application Repository publik](#), atau buka [AWS Lambdakonsol](#). Pilih Buat fungsi, lalu pilih Jelajahi repositori aplikasi tanpa server.
2. Jelajahi atau cari aplikasi.

Note

Untuk menampilkan aplikasi yang berisi IAM role khusus atau kebijakan sumber daya, pilih kotak centang Tampilkan aplikasi yang membuat IAM role khusus atau kebijakan sumber daya. Untuk informasi selengkapnya tentang peran IAM kustom dan kebijakan sumber daya, lihat [Mengakui Kemampuan Aplikasi](#).

3. Pilih aplikasi untuk melihat detail seperti izin, kemampuan, dan berapa kali aplikasi telah digunakan oleh AWS pelanggan.

Jumlah penyebaran ditampilkan untuk AWS Wilayah tempat Anda mencoba menerapkan aplikasi.

4. Pada halaman detail aplikasi, lihat izin aplikasi dan sumber daya aplikasi dengan melihat AWS SAM template, lisensi, dan file readme. Di halaman ini, Anda juga dapat menemukan tautan URL kode sumber untuk aplikasi yang dibagikan secara publik. Jika aplikasi menyertakan aplikasi bersarang, Anda juga dapat melihat detail aplikasi bersarang di halaman ini.
5. Konfigurasi aplikasi di bagian Pengaturan aplikasi. Untuk panduan tentang mengkonfigurasi aplikasi tertentu, lihat file readme aplikasi itu.

Misalnya, persyaratan konfigurasi mungkin termasuk menentukan nama sumber daya yang Anda inginkan untuk memiliki akses ke aplikasi. Sumber daya semacam itu mungkin berupa tabel Amazon DynamoDB, bucket Amazon S3, atau Amazon API Gateway.

- Pilih Deploy. Melakukan hal ini akan membawa Anda ke halaman status Deployment.

Note

Jika aplikasi memiliki kemampuan yang memerlukan pengakuan, Anda harus memilih Saya mengakui aplikasi ini menciptakan peran IAM kustom atau kebijakan sumber daya kotak centang sebelum menerapkan aplikasi. Jika tidak, akan terjadi kesalahan. Untuk informasi selengkapnya tentang peran IAM kustom dan kebijakan sumber daya, lihat [Mengakui Kemampuan Aplikasi](#).

- Pada halaman status Deployment, Anda dapat melihat kemajuan penerapan Anda. Sambil menunggu penerapan Anda selesai, Anda dapat mencari dan menelusuri aplikasi lain, dan kembali ke halaman ini melalui konsol Lambda.

Setelah aplikasi Anda berhasil diterapkan, Anda dapat meninjau dan mengelola sumber daya yang telah dibuat dengan menggunakan AWS alat yang ada.

Menerapkan Aplikasi Baru (AWS CLI)

Bagian ini menunjukkan bagaimana untuk men-deploy aplikasi baru dari AWS Serverless Application Repository dengan menggunakan AWS CLI. Untuk petunjuk tentang men-deploy versi baru aplikasi yang sudah ada, lihat [Memperbarui Aplikasi](#).

Menemukan dan Mengakui Kemampuan Aplikasi (AWS CLI)

Untuk mengetahui kemampuan aplikasi menggunakan AWS CLI, ikuti langkah-langkah berikut:

- Tinjau kemampuan aplikasi. Gunakan AWS CLI perintah berikut untuk meninjau kemampuan aplikasi:

```
aws serverlessrepo get-application \  
--application-id application-arn
```

Properti respons [requiredCapabilities](#) berisi daftar kemampuan aplikasi yang perlu Anda akui sebelum Anda dapat menyebarkan aplikasi. Anda juga dapat menggunakan [GetApplication API](#) di AWS SDK untuk mendapatkan data ini.

2. Buat `changeset`. Anda harus menyediakan serangkaian [kemampuan](#) yang diperlukan saat Anda membuat AWS CloudFormation set perubahan. Misalnya, gunakan AWS CLI perintah berikut untuk menerapkan aplikasi dengan mengakui kemampuannya:

```
aws serverlessrepo create-cloud-formation-change-set \  
--application-id application-arn \  
--stack-name unique-name-for-cloud-formation-stack \  
--capabilities list-of-capabilities
```

ID `changeset` dikembalikan ketika perintah ini berhasil dijalankan. Anda memerlukan ID `changeset` untuk langkah berikutnya. Anda juga dapat menggunakan [CreateCloudFormationChangeSet API](#) di AWS SDK untuk membuat set perubahan.

Misalnya, AWS CLI perintah berikut mengakui aplikasi yang berisi [AWS::IAM::Role](#) sumber daya dengan nama kustom dan satu atau lebih aplikasi bersarang:

```
aws serverlessrepo create-cloud-formation-change-set \  
--application-id application-arn \  
--stack-name unique-name-for-cloud-formation-stack \  
--capabilities CAPABILITY_NAMED_IAM CAPABILITY_AUTO_EXPAND
```

3. Jalankan `changeset`. Mengeksekusi `changeset` sebenarnya melakukan penyebaran. Berikan ID `changeset` yang dikembalikan saat Anda membuat set perubahan pada langkah sebelumnya.

Contoh AWS CLI perintah berikut mengeksekusi `changeset` aplikasi untuk menyebarkan aplikasi:

```
aws cloudformation execute-change-set \  
--change-set-name changeset-id-arn
```

Anda juga dapat menggunakan [ExecuteChangeSet API](#) di AWS SDK untuk menjalankan set perubahan.

Menghapus Tumpukan Aplikasi

Untuk menghapus aplikasi yang sebelumnya Anda gunakan menggunakan AWS Serverless Application Repository, ikuti prosedur yang sama seperti untuk menghapus AWS CloudFormation tumpukan:

- **AWS Management Console:** Untuk menghapus aplikasi menggunakan AWS Management Console, lihat [Menghapus Stack pada AWS CloudFormation Console di AWS CloudFormation User Guide](#).
- **AWS CLI:** Untuk menghapus aplikasi menggunakan AWS CLI, lihat [Menghapus Stack](#) di AWS CloudFormation User Guide.

Memperbarui Aplikasi

Setelah Anda menerapkan aplikasi dari AWS Serverless Application Repository, Anda mungkin ingin memperbaruinya. Misalnya, Anda mungkin ingin mengubah pengaturan aplikasi, atau Anda mungkin ingin memperbarui aplikasi ke versi terbaru yang telah diublikasikan.

Bagian berikut menjelaskan cara menerapkan versi baru dari aplikasi dengan menggunakan salah satu AWS Management Console atau AWS CLI.

Memperbarui Aplikasi (Konsol)

Untuk memperbarui aplikasi yang sebelumnya Anda gunakan, gunakan prosedur yang sama seperti menerapkan aplikasi baru, dan berikan nama aplikasi yang sama dengan yang Anda gunakan sebelumnya. Secara khusus, AWS Serverless Application Repository prepend `serverlessrepo-` ke nama aplikasi Anda. Namun, untuk men-deploy versi baru aplikasi Anda, Anda memberikan nama aplikasi asli tanpa `serverlessrepo-` prepended.

Misalnya, jika Anda menerapkan aplikasi dengan nama `MyApplication`, nama stack akan `serverlessrepo-MyApplication`. Untuk memperbarui aplikasi itu, Anda akan memberikan nama `MyApplication` lagi-jangan tentukan nama tumpukan lengkap `serverlessrepo-MyApplication`.

Untuk semua pengaturan aplikasi lainnya, Anda dapat menjaga nilainya sama dengan penyebaran sebelumnya, atau memberikan nilai baru.

Memperbarui Aplikasi (AWS CLI)

Untuk memperbarui aplikasi yang sebelumnya Anda gunakan, gunakan prosedur yang sama seperti menerapkan aplikasi baru, dan berikan hal- `-stack-name` yang sama dengan yang Anda gunakan sebelumnya. Secara khusus, AWS Serverless Application Repository prepend `serverlessrepo-` ke nama stack Anda. Namun, untuk men-deploy versi baru aplikasi Anda, Anda memberikan nama tumpukan asli tanpa `serverlessrepo-` prepended.

Misalnya, jika Anda menerapkan aplikasi dengan nama `stackMyApplication`, nama `stack` yang dibuat akan `serverlessrepo-MyApplication`. Untuk memperbarui aplikasi itu, Anda akan memberikan nama `MyApplication` lagi-jangan tentukan nama tumpukan lengkap `serverlessrepo-MyApplication`.

Keamanan di AWS Serverless Application Repository

Keamanan cloud di AWS merupakan prioritas tertinggi. Sebagai pelanggan AWS, Anda akan 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 AWS dan Anda. [Model tanggung jawab bersama](#) menggambarkan ini sebagai keamanan dari cloud dan keamanan di dalam cloud:

- Keamanan cloud – AWS bertanggung jawab untuk melindungi infrastruktur yang menjalankan layanan AWS di dalam AWS Cloud. AWS juga memberi layanan yang dapat Anda gunakan dengan aman. Auditor pihak ketiga menguji dan memverifikasi efektivitas keamanan kami sebagai bagian dari [program kepatuhan AWS](#). Untuk mempelajari program kepatuhan yang berlaku di AWS Serverless Application Repository, lihat [AWS Layanan dalam Lingkup oleh Program Kepatuhan](#).
- Keamanan di cloud – Tanggung jawab Anda ditentukan menurut layanan AWS yang Anda gunakan. Anda juga bertanggung jawab atas faktor lain termasuk sensitivitas data Anda, persyaratan perusahaan Anda, serta hukum dan peraturan yang berlaku.

Dokumentasi ini membantu Anda memahami cara menerapkan model tanggung jawab bersama saat menggunakan AWS Serverless Application Repository. Topik berikut menunjukkan kepada Anda cara mengonfigurasi AWS Serverless Application Repository untuk memenuhi tujuan keamanan dan kepatuhan Anda. Anda juga mempelajari cara menggunakan layanan AWS lain yang membantu Anda memantau dan mengamankan sumber daya AWS Serverless Application Repository Anda.

Topik

- [Perlindungan Data di AWS Serverless Application Repository](#)
- [Identity and Access Management untuk AWS Serverless Application Repository](#)
- [Pencatatan dan Pemantauan di AWS Serverless Application Repository](#)
- [Validasi Kepatuhan untuk AWS Serverless Application Repository](#)
- [Ketahanan di AWS Serverless Application Repository](#)
- [Keamanan Infrastruktur di AWS Serverless Application Repository](#)

Perlindungan Data di AWS Serverless Application Repository

[Model tanggung jawab bersama](#) AWS diterapkan untuk perlindungan data AWS Serverless Application Repository. Sebagaimana dijelaskan dalam model ini, AWS bertanggung jawab untuk melindungi infrastruktur global yang menjalankan semua AWS Cloud. Anda harus bertanggung jawab untuk memelihara kendali terhadap konten yang di-hosting pada infrastruktur ini. Anda juga bertanggung jawab atas konfigurasi keamanan dan tugas manajemen untuk Layanan AWS yang Anda gunakan. Untuk informasi lebih lanjut tentang privasi data, lihat [FAQ tentang Privasi Data](#). Untuk informasi tentang perlindungan data di Eropa, lihat postingan blog [Model Tanggung Jawab Bersama AWS dan GDPR](#) 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 tersebut, setiap pengguna hanya diberi izin yang diperlukan untuk memenuhi tugas pekerjaan mereka. Kami juga merekomendasikan agar Anda mengamankan data Anda dengan cara-cara berikut:

- Gunakan autentikasi multi-faktor (MFA) pada setiap akun.
- Gunakan SSL/TLS untuk melakukan komunikasi dengan sumber daya AWS. Kami membutuhkan TLS 1.2 dan merekomendasikan TLS 1.3.
- Siapkan API dan log aktivitas pengguna dengan AWS CloudTrail.
- Gunakan solusi AWS enkripsi, bersama dengan semua kontrol keamanan default di dalamnya Layanan AWS.
- Gunakan layanan keamanan terkelola tingkat 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 ketika mengakses AWS melalui antarmuka baris perintah atau API, gunakan titik akhir FIPS. Untuk informasi lebih lanjut tentang titik akhir FIPS yang tersedia, lihat [Standar Pemrosesan Informasi Federal \(FIPS\) 140-2](#).

Kami sangat menyarankan agar Anda tidak pernah memasukkan informasi rahasia atau sensitif, seperti alamat email pelanggan Anda, ke dalam tag atau bidang teks bentuk bebas seperti bidang Nama. Ini termasuk saat Anda bekerja dengan AWS Serverless Application Repository atau lainnya Layanan AWS menggunakan konsol, API AWS CLI, atau AWS SDK. Data apa pun yang Anda masukkan ke dalam tag atau bidang teks bentuk bebas yang digunakan untuk nama dapat digunakan untuk penagihan atau log diagnostik. Saat Anda memberikan URL ke server eksternal, sebaiknya

Anda tidak menyertakan informasi kredensial di URL untuk memvalidasi permintaan Anda ke server tersebut.

Enkripsi Saat Data Berpindah

AWS Serverless Application Repository Titik akhir API hanya mendukung koneksi aman melalui HTTPS. Saat Anda mengelola AWS Serverless Application Repository sumber daya dengan AWS Management Console, AWS SDK, atau AWS Serverless Application Repository API, semua komunikasi dienkripsi dengan Transport Layer Security (TLS).

Untuk daftar lengkap titik akhir API, lihat [AWS Wilayah dan Titik Akhir](#) di Referensi Umum AWS

Enkripsi saat Data Tidak Berpindah

AWS Serverless Application Repository Enkripsi file yang Anda unggah ke AWS Serverless Application Repository, termasuk paket penyebaran dan arsip lapisan.

Identity and Access Management untuk AWS Serverless Application Repository

AWS Identity and Access Management (IAM) adalah Layanan 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 AWS Serverless Application Repository. IAM adalah Layanan AWS yang dapat Anda gunakan tanpa biaya tambahan.

Untuk mendapatkan gambaran umum tentang cara kerja IAM, lihat [Memahami Cara Kerja IAM](#) di Panduan Pengguna IAM.

Topik

- [Penonton](#)
- [Mengautentikasi dengan Identitas](#)
- [Mengelola Akses Menggunakan Kebijakan](#)
- [Bagaimana AWS Serverless Application Repository Bekerja dengan IAM](#)
- [Contoh Kebijakan Berbasis Identitas AWS Serverless Application Repository](#)
- [AWS Serverless Application Repository Contoh Kebijakan Aplikasi](#)
- [AWS Serverless Application Repository Izin API: Referensi Tindakan dan Sumber Daya](#)

- [Pemecahan Masalah Identitas dan Akses AWS Serverless Application Repository](#)

Penonton

Cara menggunakan AWS Identity and Access Management (IAM) berbeda, tergantung pada pekerjaan yang Anda lakukan di AWS Serverless Application Repository.

Pengguna layanan – Jika Anda menggunakan layanan AWS Serverless Application Repository untuk melakukan tugas Anda, administrator Anda akan memberikan kredensial dan izin yang Anda butuhkan. Saat Anda menggunakan lebih banyak fitur AWS Serverless Application Repository untuk melakukan pekerjaan, Anda mungkin memerlukan izin tambahan. Memahami cara mengelola akses dapat membantu Anda meminta izin yang tepat dari administrator Anda. Jika Anda tidak dapat mengakses fitur di AWS Serverless Application Repository, lihat [Pemecahan Masalah Identitas dan Akses AWS Serverless Application Repository](#).

Administrator layanan – Jika Anda bertanggung jawab atas sumber daya AWS Serverless Application Repository di perusahaan Anda, Anda mungkin memiliki akses penuh ke AWS Serverless Application Repository. Tugas Anda adalah menentukan AWS Serverless Application Repository fitur dan sumber daya mana yang harus diakses pengguna layanan Anda. Anda kemudian harus mengirimkan permintaan ke administrator IAM Anda untuk mengubah izin pengguna layanan Anda. Tinjau informasi di halaman ini untuk memahami konsep dasar IAM. Untuk mempelajari lebih lanjut tentang cara perusahaan Anda dapat menggunakan IAM dengan AWS Serverless Application Repository, lihat [Bagaimana AWS Serverless Application Repository Bekerja dengan IAM](#).

Administrator IAM – Jika Anda adalah administrator IAM, Anda mungkin ingin belajar dengan lebih detail tentang cara Anda menulis kebijakan untuk mengelola akses ke AWS Serverless Application Repository. Untuk melihat contoh kebijakan berbasis identitas AWS Serverless Application Repository yang dapat Anda gunakan di IAM, lihat [Contoh Kebijakan Berbasis Identitas AWS Serverless Application Repository](#).

Mengautentikasi dengan Identitas

Autentikasi adalah cara Anda untuk masuk ke AWS menggunakan kredensial identitas Anda. Anda harus diautentikasi (masuk ke AWS) sebagai Pengguna root akun AWS, sebagai pengguna IAM, atau dengan mengasumsikan peran IAM.

Anda dapat masuk AWS sebagai identitas federasi dengan menggunakan kredensial yang disediakan melalui sumber identitas. AWS IAM Identity Center Pengguna (IAM Identity Center), autentikasi masuk tunggal perusahaan Anda, dan kredensial Google atau Facebook Anda adalah contoh identitas

federasi. Saat Anda masuk sebagai identitas federasi, administrator Anda sebelumnya menyiapkan federasi identitas menggunakan peran IAM. Ketika Anda mengakses AWS dengan menggunakan federasi, Anda secara tidak langsung mengambil peran.

Bergantung pada jenis pengguna Anda, Anda dapat masuk ke AWS Management Console atau portal AWS akses. Untuk informasi selengkapnya tentang masuk AWS, lihat [Cara masuk ke Panduan AWS Sign-In Pengguna Anda Akun AWS](#).

Jika Anda mengakses AWS secara terprogram, AWS sediakan kit pengembangan perangkat lunak (SDK) dan antarmuka baris perintah (CLI) untuk menandatangani permintaan Anda secara kriptografis dengan menggunakan kredensial Anda. Jika Anda tidak menggunakan AWS alat, Anda harus menandatangani permintaan sendiri. Untuk informasi selengkapnya tentang penggunaan metode yang disarankan untuk menandatangani permintaan sendiri, lihat [Menandatangani permintaan AWS API](#) di Panduan Pengguna IAM.

Terlepas dari metode otentikasi yang Anda gunakan, Anda mungkin diminta untuk memberikan informasi keamanan tambahan. Misalnya, AWS menyarankan supaya Anda menggunakan autentikasi multi-faktor (MFA) untuk meningkatkan keamanan akun Anda. Untuk mempelajari lebih lanjut, lihat [Autentikasi multi-faktor di Panduan AWS IAM Identity Center Pengguna dan Menggunakan otentikasi multi-faktor \(MFA\) AWS](#) di Panduan Pengguna IAM.

Pengguna root Akun AWS

Saat Anda membuat Akun AWS, Anda mulai dengan satu identitas masuk yang memiliki akses lengkap ke semua Layanan AWS dan sumber daya di akun. Identitas ini disebut pengguna root Akun AWS dan diakses dengan cara masuk menggunakan alamat email dan kata sandi yang Anda gunakan saat membuat akun. Kami sangat menyarankan agar Anda tidak menggunakan pengguna root untuk tugas sehari-hari Anda. Lindungi kredensial pengguna root Anda dan gunakan untuk melakukan tugas-tugas yang hanya dapat dilakukan oleh pengguna root. Untuk daftar lengkap tugas yang mengharuskan Anda masuk sebagai pengguna root, lihat [Tugas yang memerlukan kredensi pengguna root di Panduan Pengguna IAM](#).

Pengguna dan Grup IAM

[Pengguna IAM](#) adalah identitas dalam akun Akun AWS Anda yang memiliki izin khusus untuk satu orang atau aplikasi. Jika memungkinkan, sebaiknya mengandalkan kredensi sementara daripada membuat pengguna IAM yang memiliki kredensi jangka panjang seperti kata sandi dan kunci akses. Namun, jika Anda memiliki kasus penggunaan khusus yang memerlukan kredensial jangka panjang dengan pengguna IAM, kami sarankan Anda memutar kunci akses. Untuk informasi selengkapnya,

lihat [Memutar kunci akses secara teratur untuk kasus penggunaan yang memerlukan kredensial jangka panjang](#) di Panduan Pengguna IAM.

[Grup IAM](#) adalah identitas yang menentukan kumpulan dari para pengguna IAM. Anda tidak dapat masuk sebagai kelompok. Anda dapat menggunakan grup untuk menentukan izin untuk beberapa pengguna sekaligus. Grup membuat izin lebih mudah dikelola untuk set besar pengguna. Misalnya, Anda dapat memiliki grup yang diberi nama IAMAdmins dan memberikan izin kepada grup tersebut untuk mengelola sumber daya IAM.

Para pengguna berbeda dari peran. Seorang pengguna secara unik terkait dengan satu orang atau aplikasi, tetapi peran ini dimaksudkan untuk dapat digunakan oleh siapa pun yang membutuhkannya. Pengguna memiliki kredensial jangka panjang permanen, tetapi peran memberikan kredensial sementara. Untuk mempelajari selengkapnya, lihat [Kapan harus membuat pengguna IAM \(bukan peran\)](#) dalam Panduan Pengguna IAM.

IAM Role

[IAM role](#) adalah identitas dalam akun Akun AWS Anda yang memiliki izin khusus. Peran ini mirip dengan pengguna IAM, tetapi tidak terkait dengan orang tertentu. Anda dapat menggunakan IAM role untuk sementara di dalam AWS Management Console dengan cara [berganti peran](#). Anda dapat mengambil peran dengan cara memanggil operasi API AWS CLI atau AWS atau menggunakan URL khusus. Untuk informasi selengkapnya tentang metode penggunaan peran, lihat [Menggunakan IAM roles](#) dalam Panduan Pengguna IAM.

IAM role dengan kredensial sementara berguna dalam situasi berikut:

- Akses pengguna terfederasi — Untuk menetapkan izin ke identitas federasi, Anda membuat peran dan menentukan izin untuk peran tersebut. Ketika identitas federasi mengautentikasi, identitas dikaitkan dengan peran dan diberikan izin yang ditentukan oleh peran. Untuk informasi tentang peran untuk federasi, lihat [Membuat peran untuk Penyedia Identitas pihak ketiga](#) di Panduan Pengguna IAM. Jika Anda menggunakan Pusat Identitas IAM, Anda mengonfigurasi set izin. Untuk mengontrol apa yang dapat diakses identitas Anda setelah diautentikasi, IAM Identity Center mengkorelasikan izin yang disetel ke peran dalam IAM. Untuk informasi tentang set izin, lihat [Set izin](#) di Panduan AWS IAM Identity Center Pengguna.
- Izin pengguna IAM sementara — Pengguna atau peran IAM dapat mengambil peran IAM untuk sementara mengambil izin yang berbeda untuk tugas tertentu.
- Akses lintas akun – Anda dapat menggunakan IAM role agar seseorang (principal tepercaya) di akun lain diizinkan untuk mengakses sumber daya di akun Anda. Peran adalah cara utama untuk

memberikan akses lintas akun. Namun, dengan beberapa Layanan AWS, Anda dapat melampirkan kebijakan secara langsung ke sumber daya (alih-alih menggunakan peran sebagai proxy). Untuk mempelajari perbedaan antara kebijakan peran dan kebijakan berbasis sumber daya untuk akses lintas akun, lihat [Perbedaan antara IAM role dan kebijakan berbasis sumber daya](#) dalam Panduan Pengguna IAM.

- Akses lintas layanan — Beberapa Layanan AWS menggunakan fitur lain Layanan AWS. Sebagai contoh, ketika Anda melakukan panggilan dalam suatu layanan, biasanya layanan tersebut menjalankan aplikasi di Amazon EC2 atau menyimpan objek di Amazon S3. Layanan mungkin melakukan ini menggunakan izin panggilan principal, menggunakan peran layanan, atau peran tertaut layanan.
- Sesi akses teruskan (FAS) — Saat Anda menggunakan pengguna atau peran IAM untuk melakukan tindakan AWS, Anda dianggap sebagai prinsipal. Ketika Anda menggunakan beberapa layanan, Anda mungkin melakukan tindakan yang kemudian memulai tindakan lain di layanan yang berbeda. FAS menggunakan izin dari pemanggilan utama Layanan AWS, dikombinasikan dengan permintaan Layanan AWS untuk membuat permintaan ke layanan hilir. Permintaan FAS hanya dibuat ketika layanan menerima permintaan yang memerlukan interaksi dengan orang lain Layanan AWS atau sumber daya untuk menyelesaikannya. Dalam hal ini, Anda harus memiliki izin untuk melakukan kedua tindakan tersebut. Untuk detail kebijakan saat membuat permintaan FAS, lihat [Meneruskan sesi akses](#).
- Peran layanan – Peran layanan adalah [IAM role](#) yang diambil oleh layanan untuk melakukan tindakan atas nama Anda. Administrator IAM dapat membuat, memodifikasi, dan menghapus peran layanan dari dalam IAM. Untuk informasi selengkapnya, lihat [Membuat peran untuk mendelegasikan izin ke Layanan AWS dalam Panduan Pengguna IAM](#).
- Peran terkait layanan — Peran terkait layanan adalah jenis peran layanan yang ditautkan ke peran layanan. Layanan AWS dapat menggunakan peran untuk melakukan tindakan atas nama Anda. Peran terkait layanan muncul di Anda Akun AWS dan dimiliki oleh layanan. Administrator IAM dapat melihat, tetapi tidak dapat mengedit izin untuk peran terkait layanan.
- Aplikasi yang berjalan di Amazon EC2 – Anda dapat menggunakan IAM role untuk mengelola kredensial sementara untuk aplikasi yang berjalan pada instans EC2, dan membuat permintaan API AWS CLI atau AWS. Menyimpan access key di dalam instans EC2 lebih disarankan. Untuk menugaskan sebuah peran AWS ke instans EC2 dan membuatnya tersedia untuk semua aplikasinya, Anda dapat membuat sebuah profil instans yang dilampirkan ke instans. Profil instans berisi peran dan memungkinkan program yang berjalan di instans EC2 untuk mendapatkan kredensial sementara. Untuk informasi selengkapnya, lihat [Menggunakan IAM role untuk memberikan izin ke aplikasi yang berjalan di instans Amazon EC2](#) dalam Panduan Pengguna IAM.

Untuk mempelajari kapan waktunya menggunakan IAM role atau pengguna IAM, lihat [Kapan harus membuat IAM role \(bukan pengguna\)](#) dalam Panduan Pengguna IAM.

Mengelola Akses Menggunakan Kebijakan

Anda mengontrol akses AWS dengan membuat kebijakan dan melampirkannya ke AWS identitas atau sumber daya. Kebijakan adalah objek AWS yang, ketika dikaitkan dengan identitas atau sumber daya, menentukan izinnya. AWS mengevaluasi kebijakan ini ketika prinsipal (pengguna, pengguna root, atau sesi peran) membuat permintaan. Izin dalam kebijakan dapat menentukan permintaan yang diizinkan atau ditolak. Sebagian besar kebijakan disimpan di AWS sebagai dokumen JSON. Untuk informasi selengkapnya tentang struktur dan isi dokumen kebijakan JSON, lihat [Gambaran Umum kebijakan JSON](#) dalam Panduan Pengguna IAM.

Administrator dapat menggunakan kebijakan JSON AWS untuk menentukan siapa yang memiliki akses ke hal apa. Yaitu, principal mana yang dapat melakukan tindakan pada sumber daya apa, dan dalam syarat apa.

Secara default, pengguna dan peran tidak memiliki izin. Untuk memberikan izin kepada pengguna untuk melakukan tindakan pada sumber daya yang mereka butuhkan, administrator IAM dapat membuat kebijakan IAM. Administrator kemudian dapat menambahkan kebijakan IAM ke peran, dan pengguna dapat mengambil peran.

Kebijakan IAM mendefinisikan izin untuk suatu tindakan terlepas dari metode yang Anda gunakan untuk pengoperasiannya. Misalnya, Anda memiliki kebijakan yang mengizinkan tindakan `iam:GetRole`. Pengguna dengan kebijakan tersebut dapat memperoleh informasi peran dari API AWS Management Console, the AWS CLI, or the AWS.

Kebijakan Berbasis Identitas

Kebijakan berbasis identitas adalah dokumen kebijakan izin JSON yang dapat Anda lampirkan ke identitas, seperti pengguna IAM, grup pengguna, atau peran. Kebijakan ini mengontrol tipe tindakan yang dapat dilakukan oleh pengguna dan peran, di sumber daya mana, dan dalam syarat. Untuk mempelajari cara membuat kebijakan berbasis identitas, lihat [Membuat kebijakan IAM](#) dalam Panduan Pengguna IAM.

Kebijakan berbasis identitas dapat dikategorikan lebih lanjut sebagai kebijakan inline atau kebijakan terkelola. Kebijakan inline disematkan secara langsung ke satu pengguna, grup, atau peran. Kebijakan terkelola adalah kebijakan mandiri yang dapat Anda lampirkan ke beberapa pengguna, grup, dan peran dalam akun AWS Anda. Kebijakan terkelola meliputi kebijakan yang dikelola

AWS dan kebijakan yang dikelola pelanggan. Untuk mempelajari cara memilih antara kebijakan terkelola atau kebijakan inline, lihat [Memilih antara kebijakan terkelola dan kebijakan inline](#) dalam Panduan Pengguna IAM.

Kebijakan Berbasis Sumber Daya

Kebijakan berbasis sumber daya adalah dokumen kebijakan JSON yang Anda lampirkan ke sumber daya. Contoh kebijakan berbasis sumber daya adalah kebijakan tepercaya IAM role dan kebijakan bucket Amazon S3. Dalam layanan yang mendukung kebijakan berbasis sumber daya, administrator layanan dapat menggunakannya untuk mengontrol akses ke sumber daya tertentu. Untuk sumber daya yang dilampiri kebijakan tersebut, kebijakan menetapkan tindakan apa yang dapat dilakukan oleh principal tertentu di sumber daya tersebut dan dalam kondisi apa. Anda harus [menentukan principal](#) dalam kebijakan berbasis sumber daya. Prinsipal dapat mencakup akun, pengguna, peran, pengguna federasi, atau. Layanan AWS

Kebijakan berbasis sumber daya merupakan kebijakan inline yang terletak di layanan tersebut. Anda tidak dapat menggunakan kebijakan yang dikelola AWS dari IAM dalam kebijakan berbasis sumber daya.

Daftar Kontrol Akses (ACL)

Daftar kontrol akses (ACL) mengendalikan principal mana (anggota akun, pengguna, atau peran) yang memiliki izin untuk mengakses sumber daya. ACL serupa dengan kebijakan berbasis sumber daya, meskipun tidak menggunakan format dokumen kebijakan JSON.

Amazon S3, AWS WAF, dan Amazon VPC adalah contoh layanan yang mendukung ACL. Untuk mempelajari ACL selengkapnya, lihat [Gambaran umum daftar kontrol akses \(ACL\)](#) dalam Panduan Developer Amazon Simple Storage Service.

Tipe Kebijakan Lainnya

AWS mendukung tipe kebijakan tambahan, yang kurang umum. Tipe kebijakan ini dapat mengatur izin maksimum yang diberikan kepada Anda oleh tipe kebijakan yang lebih umum.

- Batasan izin – Batasan izin adalah fitur lanjutan tempat Anda mengatur izin maksimum yang dapat diberikan oleh kebijakan berbasis identitas kepada entitas IAM (pengguna atau IAM role). Anda dapat menetapkan batas izin untuk suatu entitas. Izin yang dihasilkan adalah persimpangan kebijakan berbasis identitas entitas dan batas izinnya. Kebijakan berbasis sumber daya yang menentukan pengguna atau peran dalam bidang `Principal` tidak dibatasi oleh batasan izin.

Penolakan eksplisit dalam salah satu kebijakan ini dapat membatalkan izin. Untuk informasi selengkapnya tentang batasan izin, lihat [Batasan izin untuk entitas IAM](#) dalam Panduan Pengguna IAM.

- Kebijakan Kontrol Layanan (SCPs) – SCP adalah kebijakan JSON yang menentukan izin maksimum untuk sebuah organisasi atau unit organisasi (OU) di AWS Organizations. AWS Organizations adalah layanan untuk mengelompokkan dan mengelola beberapa akun AWS secara terpusat yang dimiliki oleh bisnis Anda. Jika Anda mengaktifkan semua fitur di sebuah organisasi, maka Anda dapat menerapkan kebijakan kontrol layanan (SCP) ke salah satu atau ke semua akun Anda. SCP membatasi izin untuk entitas dalam akun anggota, termasuk setiap Pengguna root akun AWS. Untuk informasi selengkapnya tentang Organizations dan SCP, lihat [Cara kerja SCP](#) dalam Panduan Pengguna AWS Organizations.
- Kebijakan sesi – Kebijakan sesi adalah kebijakan lanjutan yang Anda berikan sebagai parameter saat Anda membuat sesi sementara secara terprogram bagi peran atau pengguna gabungan. Izin sesi yang dihasilkan adalah persimpangan kebijakan berbasis identitas pengguna atau peran dan kebijakan sesi. Izin juga dapat berasal dari kebijakan berbasis sumber daya. Penolakan eksplisit dalam salah satu kebijakan ini membatalkan izin. Untuk informasi selengkapnya, lihat [Kebijakan sesi](#) dalam Panduan Pengguna IAM.

Berbagai Tipe Kebijakan

Ketika beberapa tipe kebijakan berlaku untuk sebuah permintaan, izin yang dihasilkan lebih rumit untuk dipahami. Untuk mempelajari cara AWS menentukan untuk mengizinkan permintaan ketika beberapa tipe kebijakan dilibatkan, lihat [Logika evaluasi kebijakan](#) dalam Panduan Pengguna IAM.

Bagaimana AWS Serverless Application Repository Bekerja dengan IAM

Sebelum Anda menggunakan IAM untuk mengelola akses ke AWS Serverless Application Repository, Anda harus memahami fitur IAM apa yang tersedia untuk digunakan dengan AWS Serverless Application Repository.

Untuk mendapatkan gambaran umum tentang cara kerja IAM, lihat [Memahami Cara Kerja IAM](#) di Panduan Pengguna IAM. Untuk mendapatkan tampilan tingkat tinggi tentang cara AWS Serverless Application Repository dan AWS layanan lainnya bekerja dengan IAM, lihat [AWS Layanan Yang Bekerja dengan IAM](#) dalam Panduan Pengguna IAM.

Topik

- [Kebijakan Berbasis Identitas AWS Serverless Application Repository](#)

- [AWS Serverless Application Repository Kebijakan Aplikasi](#)
- [Otorisasi Berdasarkan Tag AWS Serverless Application Repository](#)
- [AWS Serverless Application Repository IAM Role](#)

Kebijakan Berbasis Identitas AWS Serverless Application Repository

Dengan kebijakan berbasis identitas IAM, Anda dapat menentukan tindakan dan sumber daya yang diizinkan atau ditolak, serta kondisi di mana tindakan tersebut diperbolehkan atau ditolak. AWS Serverless Application Repository mendukung tindakan, sumber daya, dan kunci syarat tertentu. Untuk mempelajari semua elemen yang Anda gunakan dalam kebijakan JSON, lihat [Referensi Elemen Kebijakan JSON IAM](#) dalam Panduan Pengguna IAM.

Berikut adalah contoh kebijakan izin.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "CreateApplication",
      "Effect": "Allow",
      "Action": [
        "serverlessrepo:CreateApplication"
      ],
      "Resource": "*"
    },
    {
      "Sid": "CreateApplicationVersion",
      "Effect": "Allow",
      "Action": [
        "serverlessrepo:CreateApplicationVersion"
      ],
      "Resource": "arn:partition:serverlessrepo:region:account-id:applications/application-name"
    }
  ]
}
```

Kebijakan ini memiliki dua pernyataan:

- Pernyataan pertama memberikan izin untuk AWS Serverless Application Repository tindakan `serverlessrepo:CreateApplication` pada semua AWS Serverless Application Repository sumber daya, sebagaimana ditentukan oleh karakter wildcard (*) sebagai Resource nilai.
- Pernyataan kedua memberikan izin untuk AWS Serverless Application Repository tindakan `serverlessrepo:CreateApplicationVersion` pada AWS sumber daya dengan menggunakan Amazon Resource Name (ARN) untuk AWS Serverless Application Repository aplikasi. Aplikasi ditentukan oleh Resource nilai.

Kebijakan tidak menyebutkan Principal elemen karena dalam kebijakan berbasis identitas, Anda tidak menyebutkan penanggung jawab yang mendapatkan izin. Saat Anda melampirkan kebijakan kepada pengguna, pengguna tersebut adalah prinsipal implisit. Saat Anda melampirkan kebijakan izin pada IAM role, prinsipal yang diidentifikasi dalam kebijakan kepercayaan peran tersebut mendapatkan izinnya.

Untuk tabel yang menampilkan semua operasi AWS Serverless Application Repository API dan AWS sumber daya yang diterapkan, lihat [AWS Serverless Application Repository Izin API: Referensi Tindakan dan Sumber Daya](#).

Tindakan

Administrator dapat menggunakan kebijakan JSON AWS untuk menentukan siapa yang memiliki akses ke hal apa. Yaitu, principal mana yang dapat melakukan tindakan pada sumber daya apa, dan dalam syarat apa.

Elemen Action dari kebijakan JSON menjelaskan tindakan-tindakan yang dapat Anda gunakan untuk mengizinkan atau menolak akses dalam sebuah kebijakan. Tindakan kebijakan biasanya memiliki nama yang sama sebagai operasi API AWS terkait. Ada beberapa pengecualian, misalnya tindakan hanya dengan izin yang tidak memiliki operasi API yang cocok. Ada juga beberapa operasi yang memerlukan beberapa tindakan dalam kebijakan. Tindakan tambahan ini disebut tindakan dependen.

Sertakan tindakan dalam kebijakan untuk memberikan izin guna melakukan operasi yang terkait.

Tindakan kebijakan dalam AWS Serverless Application Repository menggunakan prefiks berikut sebelum tindakan: `serverlessrepo:`. Misalnya, untuk memberikan izin kepada seseorang untuk menjalankan AWS Serverless Application Repository instans dengan operasi AWS Serverless Application Repository `SearchApplications` API, Anda

menyertakan `serverlessrepo:SearchApplications` tindakan dalam kebijakan mereka. Pernyataan kebijakan harus memuat elemen `Action` atau `NotAction`. The AWS Serverless Application Repository menentukan serangkaian tindakannya sendiri yang menjelaskan tugas yang dapat Anda lakukan dengan layanan ini.

Untuk menetapkan beberapa tindakan dalam satu pernyataan, pisahkan dengan koma seperti berikut:

```
"Action": [
    "serverlessrepo:action1",
    "serverlessrepo:action2"
]
```

Anda dapat menentukan beberapa tindakan menggunakan wildcard (*). Misalnya, untuk menentukan semua tindakan yang dimulai dengan kata `List`, sertakan tindakan berikut:

```
"Action": "serverlessrepo:List*"
```

Untuk melihat daftar tindakan AWS Serverless Application Repository, lihat [Tindakan Ditentukan oleh AWS Serverless Application Repository](#) di Panduan Pengguna IAM.

Sumber daya

Administrator dapat menggunakan kebijakan JSON AWS untuk menentukan siapa yang memiliki akses ke hal apa. Yaitu, principal mana yang dapat melakukan tindakan pada sumber daya apa, dan dalam syarat apa.

Elemen kebijakan JSON `Resource` menentukan objek atau objek-objek yang menjadi target penerapan tindakan. Pernyataan harus mencakup elemen `Resource` atau `NotResource`. Sebagai praktik terbaik, tentukan sumber daya menggunakan [Amazon Resource Name \(ARN\)](#). Anda dapat melakukan ini untuk tindakan yang mendukung tipe sumber daya tertentu, yang dikenal sebagai izin tingkat sumber daya.

Untuk tindakan yang tidak mendukung izin tingkat sumber daya, misalnya operasi pencantuman, gunakan karakter wildcard (*) untuk menunjukkan bahwa pernyataan tersebut berlaku bagi semua sumber daya.

```
"Resource": "*"

```

Dalam AWS Serverless Application Repository, AWS sumber daya utama adalah AWS Serverless Application Repository aplikasi. AWS Serverless Application Repository aplikasi memiliki Amazon Resource Names (ARN) unik yang terkait dengannya, seperti yang ditunjukkan dalam tabel berikut.

AWS Jenis Sumber Daya	Format Amazon Resource Name (ARN) Amazon
Aplikasi	<i>arn: partisi: serverlessrepo : region: account-id: applications/application-name</i>

Untuk informasi lebih lanjut tentang format ARN, lihat [Amazon Resource Name \(ARN\) dan Namespace Layanan AWS](#).

Berikut adalah contoh kebijakan yang memberikan izin untuk `serverlessrepo:ListApplications` tindakan pada semua AWS sumber daya. Dalam implementasi saat ini, AWS Serverless Application Repository tidak mendukung identifikasi AWS sumber daya tertentu menggunakan ARN AWS sumber daya (juga disebut sebagai izin tingkat sumber daya) untuk beberapa tindakan API. Dalam kondisi ini, Anda harus menentukan karakter wildcard (*).

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ListExistingApplications",
      "Effect": "Allow",
      "Action": [
        "serverlessrepo:ListApplications"
      ],
      "Resource": "*"
    }
  ]
}
```

Untuk tabel yang menampilkan semua Tindakan AWS Serverless Application Repository API dan AWS sumber daya yang diterapkan, lihat [AWS Serverless Application Repository Izin API: Referensi Tindakan dan Sumber Daya](#).

Kunci kondisi

AWS Serverless Application Repository Tidak menyediakan kunci syarat khusus layanan, tetapi mendukung menggunakan beberapa kunci syarat global. Untuk melihat semua kunci syarat global AWS, lihat [Kunci Konteks Syarat Global AWS](#) dalam Panduan Pengguna IAM.

Contoh

Untuk melihat contoh AWS Serverless Application Repository kebijakan berbasis identitas, lihat [Contoh Kebijakan Berbasis Identitas AWS Serverless Application Repository](#).

AWS Serverless Application Repository Kebijakan Aplikasi

Kebijakan aplikasi menentukan tindakan yang pokok tertentu atau PrincipalOrg dapat melakukan padaAWS Serverless Application Repository aplikasi.

Anda dapat menambahkan izin ke kebijakan yang terkait denganAWS Serverless Application Repository aplikasi. Kebijakan izin yang dilampirkan padaAWS Serverless Application Repository aplikasi disebut sebagai kebijakan aplikasi. [Kebijakan aplikasi](#) adalah ekstensi dari [kebijakan berbasis sumber daya IAM](#). Sumber daya utama adalahAWS Serverless Application Repository aplikasi. Anda dapat menggunakan kebijakanAWS Serverless Application Repository aplikasi untuk mengelola izin penyebaran aplikasi.

AWS Serverless Application Repositorykebijakan aplikasi terutama digunakan oleh penerbit untuk memberikan izin kepada konsumen untuk menyebarkan aplikasi mereka, dan operasi terkait seperti untuk mencari dan melihat rincian aplikasi tersebut. Penerbit dapat mengatur izin aplikasi ke tiga kategori berikut:

- Pribadi - Aplikasi yang dibuat dengan akun yang sama, dan belum dibagikan dengan akun lain. Anda memiliki izin untuk menyebarkan aplikasi yang dibuat menggunakanAWS akun Anda.
- Dibagikan secara pribadi — Aplikasi yang telah dibagikan oleh penerbit secara eksplisit dengan sekumpulanAWS akun atauAWS Organizations tertentu. Anda memiliki izin untuk menyebarkan aplikasi yang telah dibagikan denganAWS akun atauAWS Organisasi Anda.
- Dibagikan secara publik - Aplikasi yang telah dibagikan oleh penerbit dengan semua orang. Anda memiliki izin untuk menyebarkan aplikasi yang dibagikan secara publik.

Anda dapat memberikan izin dengan menggunakanAWS CLI,AWS SDK, atauAWS Management Console.

Contoh

Untuk melihat contoh pengelolaan kebijakan AWS Serverless Application Repository aplikasi, lihat [AWS Serverless Application Repository Contoh Kebijakan Aplikasi](#).

Otorisasi Berdasarkan Tag AWS Serverless Application Repository

AWS Serverless Application Repository Tidak mendukung akses pengendalian ke sumber daya atau tindakan berdasarkan tanda.

AWS Serverless Application Repository IAM Role

[IAM role](#) adalah entitas dalam akun AWS Anda yang memiliki izin khusus.

Menggunakan kredensial sementara dengan AWS Serverless Application Repository

Anda dapat menggunakan kredensial sementara untuk masuk dengan federasi, untuk memainkan peran IAM, atau untuk mengambil peran lintas akun. Anda memperoleh kredensi keamanan sementara dengan memanggil operasi AWS STS API seperti [AssumeRole](#) atau [GetFederationToken](#).

AWS Serverless Application Repository Dukungan menggunakan kredensial sementara.

Peran Tertaut Layanan

AWS Serverless Application Repository Tidak mendukung peran terkait layanan.

Peran Layanan

AWS Serverless Application Repository Tidak mendukung peran layanan.

Contoh Kebijakan Berbasis Identitas AWS Serverless Application Repository

Secara default, pengguna dan peran IAM tidak memiliki izin untuk membuat atau memodifikasi sumber daya AWS Serverless Application Repository. Mereka juga tidak dapat melakukan tugas dengan menggunakan API AWS Management Console, AWS CLI, atau AWS. Administrator IAM harus membuat kebijakan IAM yang memberikan izin kepada pengguna dan peran untuk melakukan operasi API tertentu pada sumber daya yang diperlukan. Administrator kemudian harus melampirkan kebijakan tersebut ke pengguna IAM atau grup yang memerlukan izin tersebut.

Untuk mempelajari cara membuat kebijakan berbasis identitas IAM menggunakan contoh dokumen kebijakan JSON ini, lihat [Membuat kebijakan di tab JSON](#) dalam Panduan Pengguna IAM.

Topik

- [Praktik Terbaik Kebijakan](#)
- [Menggunakan Konsol AWS Serverless Application Repository](#)
- [Izinkan Pengguna untuk Melihat Izin Mereka Sendiri](#)
- [Contoh Kebijakan yang Dikelola Pelanggan](#)

Praktik Terbaik Kebijakan

Kebijakan berbasis identitas sangat kuat. Kebijakan ini menentukan apakah seseorang dapat membuat, mengakses, atau menghapus sumber daya AWS Serverless Application Repository di akun Anda. Tindakan ini dapat menimbulkan biaya untuk akun AWS Anda. Ketika Anda membuat atau mengedit kebijakan berbasis identitas, ikuti panduan dan rekomendasi ini:

- Pemberian hak istimewa terendah – Ketika Anda membuat kebijakan kustom, berikan izin yang diperlukan saja untuk melakukan tugas. Mulai dengan satu set izin minimum dan berikan izin tambahan sesuai kebutuhan. Melakukan hal tersebut lebih aman daripada memulai dengan izin yang terlalu fleksibel, lalu mencoba memperketatnya nanti. Untuk informasi lebih lanjut, lihat [Berikan hak istimewa terkecil](#) dalam Panduan Pengguna IAM.
- Aktifkan autentikasi multifaktor (MFA) untuk operasi sensitif – Untuk keamanan ekstra, mintalah pengguna IAM untuk menggunakan autentikasi multifaktor (MFA) guna mengakses sumber daya atau operasi API yang sensitif. Untuk informasi lebih lanjut, lihat [Menggunakan Autentikasi Multifaktor \(MFA\) di AWS](#) dalam Panduan Pengguna IAM.
- Gunakan syarat kebijakan untuk keamanan tambahan – Selama bisa dilakukan, tentukan persyaratan agar kebijakan berbasis identitas Anda mengizinkan akses ke sumber daya. Misalnya, Anda dapat menulis persyaratan untuk menentukan jangkauan alamat IP yang diizinkan untuk mengajukan permintaan. Anda juga dapat menulis persyaratan untuk mengizinkan permintaan hanya dalam rentang tanggal atau waktu tertentu, atau untuk mewajibkan penggunaan SSL atau autentikasi multifaktor (MFA). Untuk informasi lebih lanjut, lihat [Elemen Kebijakan IAM JSON: Syarat](#) dalam Panduan Pengguna IAM.

Menggunakan Konsol AWS Serverless Application Repository

AWS Serverless Application Repository Konsol ini AWS Serverless Application Repository. Konsol menyediakan fitur dan alur kerja yang sering memerlukan izin untuk mengelola AWS Serverless Application Repository aplikasi selain izin khusus API yang didokumentasikan di [AWS Serverless Application Repository Izin API: Referensi Tindakan dan Sumber Daya](#).

Untuk informasi lain tentang izin yang diperlukan untuk menggunakan AWS Serverless Application Repository konsol, lihat [Contoh Kebijakan yang Dikelola Pelanggan](#).

Izinkan Pengguna untuk Melihat Izin Mereka Sendiri

Contoh ini menunjukkan cara Anda dapat membuat kebijakan yang mengizinkan para pengguna IAM untuk melihat kebijakan inline dan terkelola yang dilampirkan ke identitas pengguna mereka. Kebijakan ini mencakup izin untuk menyelesaikan tindakan ini pada konsol tersebut atau secara terprogram menggunakan AWS CLI atau AWS API.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ViewOwnUserInfo",
      "Effect": "Allow",
      "Action": [
        "iam:GetUserPolicy",
        "iam:ListGroupForUser",
        "iam:ListAttachedUserPolicies",
        "iam:ListUserPolicies",
        "iam:GetUser"
      ],
      "Resource": ["arn:aws:iam::*:user/${aws:username}"]
    },
    {
      "Sid": "NavigateInConsole",
      "Effect": "Allow",
      "Action": [
        "iam:GetGroupPolicy",
        "iam:GetPolicyVersion",
        "iam:GetPolicy",
        "iam:ListAttachedGroupPolicies",
        "iam:ListGroupPolicies",
        "iam:ListPolicyVersions",
        "iam:ListPolicies",
        "iam:ListUsers"
      ],
      "Resource": "*"
    }
  ]
}
```

Contoh Kebijakan yang Dikelola Pelanggan

Contoh dalam bagian ini menyediakan sekelompok contoh kebijakan yang dapat Anda lampirkan ke pengguna. Jika Anda baru membuat kebijakan, sebaiknya buat pengguna IAM terlebih dahulu di akun Anda dan melampirkan kebijakan ke pengguna secara berurutan. Anda juga dapat menggunakan contoh ini untuk membuat kebijakan khusus tunggal yang mencakup izin untuk melakukan beberapa tindakan, dan kemudian melampirkannya ke pengguna.

Untuk informasi selengkapnya tentang cara melampirkan kebijakan kepada pengguna, lihat [Menambahkan Izin ke Pengguna](#) di Panduan Pengguna IAM.

Contoh

- [Contoh Penerbit 1: Izinkan Penerbit untuk Mendaftar Aplikasi](#)
- [Contoh Penerbit 2: Memungkinkan Penerbit untuk Melihat Detail Aplikasi atau Versi Aplikasi](#)
- [Contoh Penerbit 3: Izinkan Penerbit Membuat Aplikasi atau Versi Aplikasi](#)
- [Contoh Penerbit 4: Izinkan Penerbit Membuat Kebijakan Aplikasi untuk Berbagi Aplikasi dengan Orang Lain](#)
- [Contoh Konsumen 1: Memungkinkan Konsumen untuk Mencari Aplikasi](#)
- [Contoh Konsumen 2: Memungkinkan Konsumen untuk Melihat Detail Aplikasi](#)
- [Contoh Konsumen 3: Memungkinkan Konsumen untuk Menyebarkan Aplikasi](#)
- [Contoh Konsumen 4: Tolak Akses ke Aset Deployment](#)
- [Contoh Konsumen 5: Mencegah Pencarian Konsumen dan Penerapan Aplikasi Publik](#)

Contoh Penerbit 1: Izinkan Penerbit untuk Mendaftar Aplikasi

Pengguna IAM di akun Anda harus memiliki izin untuk `serverlessrepo:ListApplications` operasi sebelum pengguna dapat melihat apa pun di konsol. Saat Anda memberikan izin ini, konsol dapat menampilkan daftar AWS Serverless Application Repository aplikasi di AWS akun yang dibuat di AWS Wilayah tertentu yang dimiliki pengguna.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ListExistingApplications",
      "Effect": "Allow",
      "Action": [
```

```

        "serverlessrepo:ListApplications"
      ],
      "Resource": "*"
    }
  ]
}

```

Contoh Penerbit 2: Memungkinkan Penerbit untuk Melihat Detail Aplikasi atau Versi Aplikasi

Seorang pengguna dapat memilih AWS Serverless Application Repository aplikasi dan melihat rincian aplikasi. Rincian tersebut termasuk penulis, deskripsi, versi, dan informasi konfigurasi lainnya. Untuk melakukan ini, pengguna membutuhkan izin untuk operasi `serverlessrepo:GetApplication` dan `serverlessrepo:ListApplicationVersions` API untuk AWS Serverless Application Repository.

Dalam contoh berikut, izin ini diberikan untuk aplikasi tertentu yang Amazon Resource Name (ARN) ditetapkan sebagai `Resource` nilainya.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ViewApplication",
      "Effect": "Allow",
      "Action": [
        "serverlessrepo:GetApplication",
        "serverlessrepo:ListApplicationVersions"
      ],
      "Resource": "arn:aws:serverlessrepo:region:account-id:applications/application-name"
    }
  ]
}

```

Contoh Penerbit 3: Izinkan Penerbit Membuat Aplikasi atau Versi Aplikasi

Jika Anda ingin mengizinkan pengguna memiliki izin untuk membuat AWS Serverless Application Repository aplikasi, Anda perlu memberikan izin `serverlessrepo:CreateApplication`

`serverlessrepo:CreateApplicationVersions` operasi, seperti yang ditunjukkan dalam kebijakan berikut.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "CreateApplication",
      "Effect": "Allow",
      "Action": [
        "serverlessrepo:CreateApplication",
        "serverlessrepo:CreateApplicationVersion",
      ],
      "Resource": "*"
    }
  ]
}
```

Contoh Penerbit 4: Izinkan Penerbit Membuat Kebijakan Aplikasi untuk Berbagi Aplikasi dengan Orang Lain

Agar pengguna dapat berbagi aplikasi dengan orang lain, Anda harus memberi mereka izin untuk membuat kebijakan aplikasi, seperti yang ditunjukkan dalam kebijakan berikut.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ShareApplication",
      "Effect": "Allow",
      "Action": [
        "serverlessrepo:PutApplicationPolicy",
        "serverlessrepo:GetApplicationPolicy",
      ],
      "Resource": "*"
    }
  ]
}
```

Contoh Konsumen 1: Memungkinkan Konsumen untuk Mencari Aplikasi

Agar konsumen dapat mencari aplikasi, Anda harus memberi mereka izin berikut.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "SearchApplications",
      "Effect": "Allow",
      "Action": [
        "serverlessrepo:SearchApplications"
      ],
      "Resource": "*"
    }
  ]
}
```

Contoh Konsumen 2: Memungkinkan Konsumen untuk Melihat Detail Aplikasi

Pengguna dapat memilih AWS Serverless Application Repository aplikasi dan melihat detail aplikasi, seperti penulis, deskripsi, versi, dan informasi konfigurasi lainnya. Untuk melakukannya, pengguna harus memiliki izin untuk AWS Serverless Application Repository operasi berikut.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ViewApplication",
      "Effect": "Allow",
      "Action": [
        "serverlessrepo:GetApplication",
        "serverlessrepo:ListApplicationVersions"
      ],
      "Resource": "*"
    }
  ]
}
```

Contoh Konsumen 3: Memungkinkan Konsumen untuk Menyebarkan Aplikasi

Agar pelanggan dapat menerapkan aplikasi, Anda harus memberi mereka izin untuk melakukan sejumlah operasi. kebijakan yang diperlukan yang diperlukan yang diperlukan.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "DeployApplication",
      "Effect": "Allow",
      "Action": [
        "serverlessrepo:CreateCloudFormationChangeSet",
        "cloudformation:CreateChangeSet",
        "cloudformation:ExecuteChangeSet",
        "cloudformation:DescribeStacks"
      ],
      "Resource": "*"
    }
  ]
}
```

Note

Menerapkan aplikasi mungkin memerlukan izin untuk menggunakan AWS sumber daya tambahan. Karena AWS Serverless Application Repository menggunakan mekanisme penyebaran dasar yang sama seperti AWS CloudFormation, lihat [Mengontrol Akses dengan AWS Identity and Access Management](#) untuk informasi selengkapnya. Untuk bantuan terkait masalah penyebaran yang terkait dengan izin, lihat [Pemecahan Masalah: Izin IAM Tidak Cukup](#).

Contoh Konsumen 4: Tolak Akses ke Aset Deployment

Ketika aplikasi dibagikan secara pribadi dengan AWS akun, secara default, semua pengguna di akun tersebut dapat mengakses aset penyebaran semua pengguna lain di akun yang sama. Kebijakan berikut mencegah pengguna dalam akun mengakses aset penerapan, yang disimpan dalam bucket Amazon S3 untuk AWS Serverless Application Repository.

```
{
```

```

"Version": "2012-10-17",
"Statement": [
  {
    "Sid": "DenyDeploymentAssetAccess",
    "Effect": "Deny",
    "Action": [
      "s3:GetObject"
    ],
    "Resource": [
      "arn:aws:s3:::awsserverlessrepo-changesets*/*"
    ]
  }
]
}

```

Contoh Konsumen 5: Mencegah Pencarian Konsumen dan Penerapan Aplikasi Publik

Anda dapat mencegah pengguna melakukan tindakan tertentu pada aplikasi.

Kebijakan berikut berlaku untuk aplikasi publik dengan `serverlessrepo:applicationType` menjadi `public`. Ini mencegah pengguna melakukan sejumlah tindakan dengan menentukan `Effect` menjadi `Deny`. Untuk informasi lain tentang kunci ketentuan yang tersedia, lihat [Tindakan](#), lihat [Tindakan](#), lihat [Tindakan](#) [AWS Serverless Application Repository](#), lihat [Tindakan](#), lihat [Tindakan](#), lihat [Tindakan](#), lihat [Tindakan](#), lihat [Tindakan](#), lihat [Tindakan](#), lihat [Tindakan](#) [AWS Serverless Application Repository](#)

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Condition": {
        "StringEquals": {
          "serverlessrepo:applicationType": "public"
        }
      },
      "Action": [
        "serverlessrepo:SearchApplications",
        "serverlessrepo:GetApplication",
        "serverlessrepo>CreateCloudFormationTemplate",
        "serverlessrepo>CreateCloudFormationChangeSet",
        "serverlessrepo:ListApplicationVersions",
        "serverlessrepo:ListApplicationDependencies"
      ]
    }
  ]
}

```

```
    ],  
    "Resource": "*",  
    "Effect": "Deny"  
  }  
]  
}
```

Note

Pernyataan kebijakan ini juga dapat digunakan sebagai Kebijakan Kontrol Layanan dan diterapkan keAWS organisasi. Untuk informasi selengkapnya tentang Kebijakan Kontrol Layanan, lihat [Kebijakan Kontrol Layanan](#) di PanduanAWS Organizations Pengguna.

AWS Serverless Application RepositoryContoh Kebijakan Aplikasi

Kebijakan izin yang dilampirkan padaAWS Serverless Application Repository aplikasi disebut sebagai kebijakan aplikasi. Kebijakan aplikasi menentukan tindakan yang pokok tertentu atau PrincipalOrg dapat melakukan padaAWS Serverless Application Repository aplikasi.

SebuahAWS Serverless Application Repository aplikasi adalahAWS sumber daya utama dalamAWS Serverless Application Repository. AWS Serverless Application Repositorykebijakan aplikasi terutama digunakan oleh penerbit untuk memberikan izin kepada konsumen untuk menyebarkan aplikasi mereka, dan operasi terkait seperti untuk mencari dan melihat rincian aplikasi tersebut.

Penerbit dapat mengatur izin aplikasi ke tiga kategori berikut:

- **Pribadi** - Aplikasi yang dibuat dengan akun yang sama, dan belum dibagikan dengan akun lain. Hanya konsumen yang berbagiAWS akun Anda yang memiliki izin untuk menerapkan aplikasi pribadi.
- **Dibagikan secara pribadi** — Aplikasi yang telah dibagikan oleh penerbit secara eksplisit dengan sekumpulanAWS akun tertentu, atau denganAWS akun diAWS organisasi. Konsumen memiliki izin untuk menyebarkan aplikasi yang telah dibagikan denganAWS akun atauAWS organisasi mereka. Untuk informasi selengkapnya tentangAWS organisasi, lihat [PanduanAWS Organizations Pengguna](#).
- **Dibagikan secara publik** - Aplikasi yang telah dibagikan oleh penerbit dengan semua orang. Semua konsumen memiliki izin untuk menyebarkan aplikasi yang dibagikan secara publik.

Note

Untuk aplikasi yang dibagikan secara pribadi, AWS Serverless Application Repository satu-satunya yang mendukung AWS Akun sebagai prinsipal. Penerbit dapat memberikan atau menolak semua pengguna dalam AWS Akun sebagai grup tunggal untuk AWS Serverless Application Repository aplikasi. Penerbit tidak dapat memberikan atau menolak pengguna individu dalam suatu AWS Akun ke AWS Serverless Application Repository aplikasi.

Untuk petunjuk tentang pengaturan izin aplikasi menggunakan AWS Management Console, lihat [Berbagi Aplikasi](#).

Untuk petunjuk tentang pengaturan izin aplikasi menggunakan AWS CLI dan contoh, lihat bagian berikut.

Izin Aplikasi (AWS CLI dan AWS SDK)

Saat menggunakan AWS SDK AWS CLI atau untuk menetapkan izin AWS Serverless Application Repository aplikasi, Anda dapat menentukan tindakan berikut:

Action	Deskripsi
GetApplication	Memberikan izin untuk melihat informasi tentang aplikasi.
CreateCloudFormationChangeSet	Memberikan izin untuk aplikasi yang akan digunakan. Catatan: Tindakan ini tidak memberikan izin lain selain untuk menyebarkan.
CreateCloudFormationTemplate	Memberikan izin untuk membuat AWS CloudFormation templat untuk aplikasi.
ListApplicationVersions	Memberikan izin untuk mencantumkan versi aplikasi.
ListApplicationDependencies	Memberikan izin untuk mencantumkan daftar aplikasi yang bersarang di aplikasi yang berisi.
SearchApplications	Memberikan izin untuk aplikasi yang akan dicari.

Action	Deskripsi
Deploy	Tindakan ini memungkinkan semua tindakan yang tercantum sebelumnya dalam tabel. Artinya, ia memberikan izin untuk aplikasi yang akan dilihat, untuk itu untuk digunakan, untuk versi yang akan terdaftar, dan untuk itu yang akan dicari.

Contoh Kebijakan Aplikasi

Contoh berikut menunjukkan cara memberikan izin dengan menggunakan AWS CLI. Untuk informasi tentang cara memberikan izin menggunakan AWS Management Console, lihat [Berbagi Aplikasi](#).

Semua contoh di bagian ini menggunakan AWS CLI perintah ini untuk mengelola kebijakan izin yang terkait dengan AWS Serverless Application Repository aplikasi:

- [put-application-policy](#)
- [get-application-policy](#)

Topik

- [Contoh 1: Berbagi Aplikasi dengan Akun Lain](#)
- [Contoh 2: Membagikan Aplikasi Secara Publik](#)
- [Contoh 3: Membuat Aplikasi Pribadi](#)
- [Contoh 4: Menentukan Beberapa Akun dan Izin](#)
- [Contoh 5: Berbagi Aplikasi dengan Semua Akun di AWS Organisasi](#)
- [Contoh 6: Berbagi Aplikasi dengan Beberapa Akun di AWS Organisasi](#)
- [Contoh 7: Mengambil Kebijakan Aplikasi](#)
- [Contoh 8: Izinkan Aplikasi Bersarang oleh Akun Tertentu](#)

Contoh 1: Berbagi Aplikasi dengan Akun Lain

Untuk berbagi aplikasi dengan akun tertentu lainnya, tetapi menjaganya agar tidak dibagikan dengan orang lain, Anda menentukan ID AWS akun yang ingin Anda bagikan sebagai prinsipal. Ini juga dikenal sebagai pengaturan aplikasi untuk dibagikan secara pribadi. Untuk melakukannya, gunakan AWS CLI perintah berikut.

```
aws serverlessrepo put-application-policy \  
--region region \  
--application-id application-arn \  
--statements Principals=account-id,Actions=Deploy
```

Note

Aplikasi yang dibagikan secara pribadi hanya dapat digunakan di AWS Wilayah yang sama di mana aplikasi dibuat.

Contoh 2: Membagikan Aplikasi Secara Publik

Untuk membuat aplikasi publik, Anda membagikannya dengan semua orang dengan menentukan “*” sebagai kepala sekolah, seperti pada contoh berikut. Aplikasi yang dibagikan untuk umum tersedia di semua Wilayah.

```
aws serverlessrepo put-application-policy \  
--region region \  
--application-id application-arn \  
--statements Principals=*,Actions=Deploy
```

Note

Dalam rangka untuk berbagi aplikasi publik, itu harus memiliki kedua `SemanticVersion` dan `LicenseUrl` properti ditetapkan.

Contoh 3: Membuat Aplikasi Pribadi

Anda dapat membuat aplikasi pribadi, sehingga tidak dibagikan dengan siapa pun dan hanya dapat digunakan oleh AWS akun yang memilikinya. Untuk melakukannya, Anda menghapus prinsip dan tindakan dari kebijakan, yang juga menghapus izin dari akun lain dalam AWS organisasi Anda agar tidak menerapkan aplikasi Anda.

```
aws serverlessrepo put-application-policy \  
--region region \  
--application-id application-arn \  
--statements Principals=*,Actions=Deploy
```

```
--statements '[]'
```

Note

Aplikasi pribadi hanya dapat digunakan diAWS Wilayah yang sama di mana aplikasi dibuat.

Contoh 4: Menentukan Beberapa Akun dan Izin

Anda dapat memberikan beberapa izin, dan Anda dapat memberikannya ke lebih dari satuAWS akun sekaligus. Untuk melakukannya, Anda menentukan daftar sebagai pokok dan tindakan, seperti yang ditunjukkan dalam contoh berikut.

```
aws serverlessrepo put-application-policy \  
--region region \  
--application-id application-arn \  
--statements Principals=account-id-1,account-  
id-2,Actions=GetApplication,CreateCloudFormationChangeSet
```

Contoh 5: Berbagi Aplikasi dengan Semua Akun diAWS Organisasi

Izin dapat diberikan kepada semua pengguna dalam suatuAWS organisasi. Anda melakukannya dengan menentukan ID organisasi Anda, seperti dalam contoh berikut.

```
aws serverlessrepo put-application-policy \  
--region region \  
--application-id application-arn \  
--statements Principals=*,PrincipalOrgIDs=org-id,Actions=Deploy,UnshareApplication
```

Untuk informasi selengkapnya tentangAWS organisasi, lihat [PanduanAWS Organizations Pengguna](#).

Note

Anda hanya dapat menentukanAWS organisasi yang menjadi anggotaAWS akun Anda. Jika Anda mencoba menentukanAWS organisasi tempat Anda bukan anggota, kesalahan akan terjadi.

Untuk berbagi aplikasi Anda denganAWS organisasi Anda, Anda harus menyertakan izin untukUnshareApplication tindakan tersebut, jika berbagi perlu dicabut di masa future.

Contoh 6: Berbagi Aplikasi dengan Beberapa Akun diAWS Organisasi

Izin dapat diberikan ke akun tertentu dalam suatuAWS organisasi. Anda melakukannya dengan menentukan daftarAWS akun sebagai prinsipal, dan ID organisasi Anda, seperti pada contoh berikut.

```
aws serverlessrepo put-application-policy \  
--region region \  
--application-id application-arn \  
--statements Principals=account-id-1,account-id-2,PrincipalOrgIDs=org-  
id,Actions=Deploy,UnshareApplication
```

Note

Anda hanya dapat menentukanAWS organisasi yang menjadi anggotaAWS akun Anda. Jika Anda mencoba menentukanAWS organisasi tempat Anda bukan anggota, kesalahan akan terjadi.

Untuk berbagi aplikasi Anda denganAWS organisasi Anda, Anda harus menyertakan izin untukUnshareApplication tindakan tersebut, jika berbagi perlu dicabut di masa future.

Contoh 7: Mengambil Kebijakan Aplikasi

Untuk melihat kebijakan aplikasi saat ini, misalnya untuk melihat apakah saat ini sedang dibagikan, Anda menggunakan*get-application-policy* perintah, seperti pada contoh berikut.

```
aws serverlessrepo get-application-policy \  
--region region \  
--application-id application-arn
```

Contoh 8: Izinkan Aplikasi Bersarang oleh Akun Tertentu

Aplikasi publik diizinkan untuk disarangkan oleh siapa pun. Jika Anda hanya ingin mengizinkan aplikasi Anda bersarang oleh akun tertentu, Anda harus menetapkan izin minimal berikut, seperti yang ditunjukkan pada contoh berikut.

```
aws serverlessrepo put-application-policy \  
--region region \  
--application-id application-arn \  
--statements Principals=account-id-1,account-  
id-2,Actions=GetApplication,CreateCloudFormationTemplate
```

AWS Serverless Application Repository Izin API: Referensi Tindakan dan Sumber Daya

Ketika Anda mengatur [kontrol akses](#) dan menulis kebijakan izin yang dapat Anda lampirkan ke identitas IAM (kebijakan berbasis identitas), Anda dapat menggunakan tabel berikut sebagai referensi. Parameter setiap AWS Serverless Application Repository Operasi API, tindakan terkait yang dapat Anda berikan izin untuk melakukan tindakan tersebut, dan AWS sumber daya yang dapat Anda berikan izin. Anda menentukan tindakan di kolom `Action` kebijakan, dan Anda menentukan nilai sumber daya di kolom `Resource` kebijakan.

Untuk menentukan tindakan, gunakan awalan `serverlessrepo:` diikuti dengan nama operasi API (misalnya, `serverlessrepo:ListApplications`).

Operasi	URI	Metode	AWS Sumber Daya (ARN)
Operasi: ListApplications Izin yang Diperlukan: <code>serverlessrepo:ListApplications</code>	/aplikasi	DAPATKAN	*
Operasi: CreateApplication Izin yang Diperlukan: <code>serverlessrepo:CreateApplication</code>	/aplikasi	POST	*
Operasi: GetApplication Izin yang Diperlukan: <code>serverlessrepo:GetApplication</code>	/aplikasi/ <i>ID aplikasi</i>	DAPATKAN	arn:aws:serverlessrepo: <i>daerah</i> : <i>account-id</i> :aplikasi/ <i>nama aplikasi</i>
Operasi: DeleteApplication	/aplikasi/ <i>ID aplikasi</i>	HAPUS	arn:aws:serverlessrepo: <i>daerah</i> : <i>account-</i>

Operasi	URI	Metode	AWSSumber Daya (ARN)
Izin yang Diperlukan:serverlessrepo:DeleteApplication			<i>id</i> :aplikasi/ <i>nama aplikasi</i>
Operasi: UpdateApplication Izin yang Diperlukan:serverlessrepo:UpdateApplication	/aplikasi/ <i>ID aplikasi</i>	TAMBALAN	arn:aws:serverlessrepo: <i>daerah</i> : <i>account-id</i> :aplikasi/ <i>nama aplikasi</i>
Operasi: CreateCloudFormationChangeSET Izin yang Diperlukan:serverlessrepo:CreateCloudFormationChangeSET	/aplikasi/ <i>ID aplikasi</i> /changesets	POST	arn:aws:serverlessrepo: <i>daerah</i> : <i>account-id</i> :aplikasi/ <i>nama aplikasi</i>
Operasi: GetApplicationKebijakan Izin yang Diperlukan:serverlessrepo:GetApplicationKebijakan	/aplikasi/ <i>ID aplikasi</i> /kebijakan	DAPATKAN	arn:aws:serverlessrepo: <i>daerah</i> : <i>account-id</i> :aplikasi/ <i>nama aplikasi</i>
Operasi: PutApplicationKebijakan Izin yang Diperlukan:serverlessrepo:PutApplicationKebijakan	/aplikasi/ <i>ID aplikasi</i> /kebijakan	PUT	arn:aws:serverlessrepo: <i>daerah</i> : <i>account-id</i> :aplikasi/ <i>nama aplikasi</i>

Operasi	URI	Metode	AWSSumber Daya (ARN)
Operasi: ListApplicationVersi Izin yang Diperlukan:serverlessrepo:ListApplicationVersi	/aplikasi/ <i>ID</i> <i>aplikasi</i> /versi	DAPATKAN	arn:aws:serverlessrepo: <i>daerah</i> : <i>account-id</i> :aplikasi/ <i>nama</i> <i>aplikasi</i>
Operasi: CreateApplicationVersi Izin yang Diperlukan:serverlessrepo:CreateApplicationVersi	/aplikasi/ <i>ID</i> <i>aplikasi</i> / versi/ <i>semantik-</i> <i>versi</i>	PUT	arn:aws:serverlessrepo: <i>daerah</i> : <i>account-id</i> :aplikasi/ <i>nama</i> <i>aplikasi</i>
Operasi: ListApplicationDependensi Izin yang Diperlukan:serverlessrepo:ListApplicationDependensi	/aplikasi/ <i>ID</i> <i>aplikasi</i> /Dependensi	DAPATKAN	arn:aws:serverlessrepo: <i>daerah</i> : <i>account-id</i> :aplikasi/ <i>nama</i> <i>aplikasi</i>
Operasi: SearchApplications Izin yang Diperlukan:serverlessrepo:SearchApplications	tidak berlaku	tidak ada	*

Pemecahan Masalah Identitas dan Akses AWS Serverless Application Repository

Gunakan informasi berikut untuk membantu Anda mendiagnosis dan memperbaiki masalah umum yang mungkin Anda temukan saat bekerja dengan AWS Serverless Application Repository dan IAM.

Topik

- [Saya tidak Berwenang untuk Melakukan TindakanAWS Serverless Application Repository](#)
- [Saya Tidak Diizinkan untuk Melakukan iam:PassRole](#)
- [Saya adalah administrator dan ingin mengizinkan orang lain untuk mengaksesAWS Serverless Application Repository](#)
- [Saya ingin mengizinkan orang di LuarAWS AkunAWS Serverless Application Repository](#)

Saya tidak Berwenang untuk Melakukan TindakanAWS Serverless Application Repository

Jika AWS Management Console memberi tahu bahwa Anda tidak diotorisasi untuk melakukan tindakan, Anda harus menghubungi administrator untuk mendapatkan bantuan. Administrator adalah orang yang memberikan nama pengguna dan kata sandi Anda untuk Anda.

Contoh kesalahan berikut terjadi ketika pengguna `mateojackson` IAM pengguna mencoba menggunakan konsol untuk melihat detail tentang aplikasi, tetapi tidak memiliki `serverlessrepo:GetApplication` izin.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
serverlessrepo:GetApplication on resource: my-example-application
```

Dalam hal ini, Mateo meminta administratornya untuk memperbarui kebijakannya agar dia dapat mengakses `my-example-application` sumber daya dengan menggunakan `serverlessrepo:GetApplication` operasi.

Saya Tidak Diizinkan untuk Melakukan iam:PassRole

Jika Anda menerima kesalahan bahwa Anda tidak terotorisasi untuk melakukan `iam:PassRole` tindakan, maka Anda dapat memberikan kesalahan bahwa Anda tidak terotorisasi untuk melakukan tindakan tersebut `AWS Serverless Application Repository`.

Beberapa Layanan AWS memungkinkan Anda untuk memberikan peran yang sudah ada ke layanan tersebut alih-alih membuat peran layanan baru atau peran tertaut layanan. Untuk melakukannya, Anda harus memiliki izin untuk meneruskan peran ke layanan.

Contoh kesalahan berikut terjadi ketika pengguna IAM bernama `marymajor` mencoba menggunakan konsol tersebut untuk melakukan tindakan di `AWS Serverless Application Repository`. Namun,

tindakan tersebut mengharuskan layanan untuk memiliki izin yang diberikan oleh peran layanan. Mary tidak memiliki izin untuk meneruskan peran tersebut ke layanan.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
iam:PassRole
```

Dalam hal ini, kebijakan Mary harus diperbarui agar dia dapat melakukan `iam:PassRole` tindakan.

Jika Anda memerlukan bantuan, maka Anda harus menghubungi AWS administrator. Administrator Anda adalah orang yang memberikan kredensial masuk.

Saya adalah administrator dan ingin mengizinkan orang lain untuk mengakses AWS Serverless Application Repository

Untuk mengizinkan orang lain mengakses AWS Serverless Application Repository, Anda harus membuat entitas IAM (pengguna atau peran) untuk orang atau aplikasi yang memerlukan akses. Mereka akan menggunakan kredensial untuk entitas tersebut untuk mengakses AWS. Anda kemudian harus melampirkan kebijakan yang memberi mereka izin yang tepat di AWS Serverless Application Repository.

Untuk segera memulai, lihat [Membuat pengguna dan grup IAM pertama Anda yang delegasikan](#) dalam Panduan Pengguna IAM.

Saya ingin mengizinkan orang di Luar AWS Akun AWS Serverless Application Repository

Anda dapat membuat peran yang dapat digunakan pengguna di akun atau orang lain di luar organisasi Anda untuk mengakses sumber daya Anda. Anda dapat menentukan siapa yang dipercaya untuk mengambil peran tersebut. Untuk layanan yang mendukung kebijakan berbasis sumber daya atau daftar kontrol akses (ACL), Anda dapat menggunakan kebijakan tersebut untuk memberi akses pada orang ke sumber daya Anda.

Untuk mempelajari selengkapnya, lihat hal berikut:

- Untuk mempelajari apakah AWS Serverless Application Repository mendukung fitur-fitur ini, lihat [Bagaimana AWS Serverless Application Repository Bekerja dengan IAM](#).
- Untuk mempelajari cara memberikan akses ke sumber daya di seluruh akun AWS yang Anda miliki, lihat [Memberikan akses ke pengguna IAM di akun AWS lain yang Anda miliki](#) dalam Panduan Pengguna IAM.

- Untuk mempelajari cara memberikan akses ke sumber daya Anda ke Akun AWS pihak ketiga, lihat [Menyediakan akses ke akun AWS yang dimiliki oleh pihak ketiga](#) dalam Panduan Pengguna IAM.
- Untuk mempelajari cara memberikan akses melalui federasi identitas, lihat [Menyediakan akses ke pengguna terautentikasi eksternal \(gabungan identitas\)](#) dalam Panduan Pengguna IAM .
- Untuk mempelajari perbedaan antara penggunaan peran dan kebijakan berbasis sumber daya untuk akses lintas akun, lihat [Perbedaan IAM role dan kebijakan berbasis sumber daya](#) di Panduan Pengguna IAM.

Pencatatan dan Pemantauan diAWS Serverless Application Repository

Pemantauan adalah bagian penting dalam menjaga keandalan, ketersediaan, dan performa solusi AWS Anda. Anda harus mengumpulkan data pemantauan dari semua bagian AndaAWSsolusi sehingga Anda dapat dengan lebih mudah melakukan debug kegagalan multipoin jika ada yang terjadi.AWSmenyediakan beberapa alat untuk memantauAWS Serverless Application Repositorysumber daya dan merespons potensi insiden, seperti berikut:

AWS CloudTrail Log

ParameterAWS Serverless Application Repositoryterintegrasi denganAWS CloudTrail, layanan yang menyediakan catatan tindakan yang diambil oleh pengguna, peran, atauAWSlayanan diAWS Serverless Application Repository. CloudTrailmerekam semua panggilan API untukAWS Serverless Application Repositorysebagai peristiwa.

Topik

- [Mencatat AWS Serverless Application Repository panggilan API dengan AWS CloudTrail](#)

Mencatat AWS Serverless Application Repository panggilan API dengan AWS CloudTrail

AWS Serverless Application Repositoryterintegrasi denganAWS CloudTrail, yang merupakan layanan yang menyediakan catatan tindakan yang diambil oleh pengguna, peran, atauAWSlayanan diAWS Serverless Application Repository. CloudTrailmerekam semua panggilan API untukAWS Serverless Application Repositorysebagai peristiwa. Panggilan yang direkam mencakup panggilan dari AWS

Serverless Application Repository konsol dan panggilan kode ke operasi API AWS Serverless Application Repository ini.

Jika Anda membuat jejak, Anda dapat mengaktifkan pengiriman berkelanjutan CloudTrail peristiwa ke bucket Amazon S3, termasuk peristiwa untuk AWS Serverless Application Repository. Jika Anda tidak membuat konfigurasi jejak, Anda masih dapat melihat kejadian terbaru dalam konsol CloudTrail di Riwayat peristiwa.

Menggunakan informasi yang dikumpulkan oleh CloudTrail, Anda dapat menentukan permintaan yang dibuat ke AWS Serverless Application Repository. Anda juga dapat menentukan alamat IP tempat permintaan dibuat, siapa yang membuat permintaan, kapan permintaan dibuat, dan detail tambahan.

Untuk mempelajari selengkapnya tentang CloudTrail, lihat [AWS CloudTrail Panduan Pengguna](#).

AWS Serverless Application Repository Informasi di CloudTrail

CloudTrail diaktifkan pada akun AWS Anda saat Anda membuat akun tersebut. Ketika aktivitas terjadi di AWS Serverless Application Repository, aktivitas yang dicatat dalam CloudTrail acara, bersama dengan lainnya AWS event layanan di Riwayat peristiwa. Anda dapat melihat, mencari, dan mengunduh peristiwa terbaru di akun AWS Anda. Untuk informasi lain, lihat [Melihat Peristiwa dengan Riwayat Peristiwa CloudTrail](#).

Untuk catatan berkelanjutan tentang peristiwa di AWS akun, termasuk acara untuk AWS Serverless Application Repository, membuat jejak. Jejak memungkinkan CloudTrail untuk mengirim berkas log ke bucket Amazon S3. Secara default, saat Anda membuat jejak di dalam konsol tersebut, jejak diterapkan ke semua Wilayah AWS. Jejak mencatat peristiwa dari semua Wilayah AWS di partisi AWS dan mengirimkan berkas log ke bucket Amazon S3 yang Anda tentukan. Selain itu, Anda dapat membuat konfigurasi layanan AWS lainnya untuk menganalisis lebih lanjut dan bertindak berdasarkan data peristiwa yang dikumpulkan di log CloudTrail. Untuk informasi selengkapnya, lihat yang berikut:

- [Ikhtisar untuk Membuat Jejak](#)
- [CloudTrail Layanan dan Integrasi yang Didukung](#)
- [Mengonfigurasi Notifikasi Amazon SNS untuk CloudTrail](#)
- [Menerima File Log CloudTrail dari Beberapa Wilayah](#) dan [Menerima File Log CloudTrail dari Beberapa Akun](#)

Semua AWS Serverless Application Repository tindakan dicatat oleh CloudTrail dan didokumentasikan pada [AWS Serverless Application Repository Sumber Daya](#) halaman. Misalnya, panggilan

keCreateApplication, UpdateApplications, dan ListApplications operasi menghasilkan entri dalam CloudTrail file log.

Setiap entri peristiwa atau log berisi informasi tentang siapa yang membuat permintaan tersebut. Informasi identitas membantu Anda menentukan hal berikut:

- Bahwa permintaan dibuat dengan kredensial pengguna root atau pengguna AWS Identity and Access Management (IAM).
- Bahwa permintaan tersebut dibuat dengan kredensial keamanan sementara untuk peran atau pengguna gabungan.
- Bahwa permintaan dibuat oleh layanan AWS lain.

Untuk informasi lain, lihat [Elemen userIdentity CloudTrail](#).

Memahami Entri Berkas Log AWS Serverless Application Repository

Jejak adalah konfigurasi yang memungkinkan pengiriman peristiwa sebagai berkas log ke bucket Amazon S3 yang telah Anda tentukan. Berkas log CloudTrail berisi satu atau beberapa entri log. Peristiwa mewakili satu permintaan dari sumber apa pun dan mencakup informasi tentang tindakan yang diminta, tanggal dan waktu tindakan, parameter permintaan, dan sebagainya. Berkas log CloudTrail bukan jejak tumpukan terurut dari panggilan API publik, sehingga berkas tersebut tidak muncul dalam urutan tertentu.

Contoh berikut menunjukkan CloudTrail catatan log yang menunjukkan CreateApplication tindakan.

```
{
  "eventVersion": "1.05",
  "userIdentity": {
    "type": "Root",
    "principalId": "999999999999",
    "arn": "arn:aws:iam::999999999999:root",
    "accountId": "999999999999",
    "accessKeyId": "ASIAUVPLBDH76HEXAMPLE",
    "sessionContext": {
      "attributes": {
        "mfaAuthenticated": "false",
        "creationDate": "2018-07-30T16:40:42Z"
      }
    }
  },
}
```

```
    "invokedBy": "signin.amazonaws.com"
  },
  "eventTime": "2018-07-30T17:37:37Z",
  "eventSource": "serverlessrepo.amazonaws.com",
  "eventName": "CreateApplication",
  "awsRegion": "us-east-1",
  "sourceIpAddress": "72.21.217.161",
  "userAgent": "signin.amazonaws.com",
  "requestParameters": {
    "licenseBody": "<content of license>",
    "sourceCodeUrl": "<sample url>",
    "spdxLicenseId": "<sample license id>",
    "readmeBody": "<content of readme>",
    "author": "<author name>",
    "templateBody": "<content of SAM template>",
    "name": "<application name>",
    "semanticVersion": "<version>",
    "description": "<content of description>",
    "homePageUrl": "<sample url>",
    "labels": [
      "<label1>",
      "<label2>"
    ]
  },
  "responseElements": {
    "licenseUrl": "<url to access content of license>",
    "readmeUrl": "<url to access content of readme>",
    "spdxLicenseId": "<sample license id>",
    "creationTime": "2018-07-30T17:37:37.045Z",
    "author": "<author name>",
    "name": "<application name>",
    "description": "<content of description>",
    "applicationId": "arn:aws:serverlessrepo:us-
east-1:999999999999:applications/<application name>",
    "homePageUrl": "<sample url>",
    "version": {
      "applicationId": "arn:aws:serverlessrepo:us-
east-1:999999999999:applications/<application name>",
      "semanticVersion": "<version>",
      "sourceCodeUrl": "<sample url>",
      "templateUrl": "<url to access content of SAM template>",
      "creationTime": "2018-07-30T17:37:37.027Z",
      "parameterDefinitions": [
        {
```

```
        "name": "<parameter name>",
        "description": "<parameter description>",
        "type": "<parameter type>"
    }
  ],
  },
  "labels": [
    "<label1>",
    "<label2>"
  ]
},
"requestID": "3f50d899-941f-11e8-ab18-01063f863be5",
"eventID": "a66a6490-d388-4a4f-8c7b-9d6ec61ab262",
"readOnly": false,
"eventType": "AwsApiCall",
"recipientAccountId": "999999999999"
}
```

Validasi Kepatuhan untuk AWS Serverless Application Repository

Auditor pihak ketiga menilai keamanan dan kepatuhan AWS Serverless Application Repository sebagai bagian dari beberapa AWS program kepatuhan. Ini mencakup SOC, PCI, FedRAMP, dan lainnya.

Untuk daftar AWS layanan yang berada di lingkup program kepatuhan spesifik, lihat [AWS Layanan dalam Lingkup oleh Program Kepatuhan](#). Untuk informasi umum, lihat [Program Kepatuhan AWS](#).

Anda bisa mengunduh laporan audit pihak ketiga dengan menggunakan AWS Artifact. Untuk informasi selengkapnya, lihat [Mengunduh Laporan di AWS Artifact](#).

Tanggung jawab kepatuhan Anda saat menggunakan AWS Serverless Application Repository ditentukan oleh sensitivitas data Anda, tujuan kepatuhan perusahaan Anda, serta undang-undang dan peraturan yang berlaku. AWS menyediakan sumber daya berikut untuk membantu kepatuhan:

- [Panduan Quick Start Keamanan dan Kepatuhan](#)— Panduan deployment ini membahas pertimbangan arsitektur dan memberikan langkah untuk men-deploy lingkungan dasar yang berfokus pada keamanan dan fokus pada AWS.
- [AWS Sumber Daya Kepatuhan](#) – Kumpulan buku kerja dan panduan ini mungkin berlaku untuk industri dan lokasi Anda.

- [AWS Config](#) – Layanan AWS ini menilai seberapa baik konfigurasi sumber daya Anda mematuhi praktik internal, pedoman industri, dan peraturan.
- [AWS Security Hub](#) – Layanan AWS ini akan menyediakan tampilan komprehensif dari status keamanan Anda dalam AWS yang akan membantu Anda memeriksa kepatuhan Anda terhadap standar dan praktik terbaik industri.

Ketahanan diAWS Serverless Application Repository

Infrastruktur global AWS dibangun di sekitar Wilayah dan Availability Zone AWS. AWS Wilayah menyediakan beberapa Availability Zone yang terpisah dan terisolasi secara fisik, yang terhubung dengan jaringan yang memiliki latensi rendah, throughput tinggi, dan sangat berlebihan. Dengan Availability Zone, Anda dapat merancang dan mengoperasikan aplikasi dan basis data yang secara otomatis melakukan fail over di antara Availability Zone tanpa gangguan. Availability Zone memiliki ketersediaan yang lebih baik, menoleransi kegagalan, dan dapat diskalakan dibandingkan satu atau beberapa infrastruktur pusat data tradisional.

Untuk informasi selengkapnya tentang Wilayah AWS dan Availability Zone, lihat [Infrastruktur Global AWS](#).

Keamanan Infrastruktur diAWS Serverless Application Repository

Sebagai layanan terkelola, AWS Serverless Application Repository dilindungi oleh AWS keamanan jaringan global. Untuk informasi tentang AWS layanan keamanan dan bagaimana AWS melindungi infrastruktur, lihat [AWS Keamanan Cloud](#). Untuk mendesain AWS lingkungan menggunakan praktik terbaik untuk keamanan infrastruktur, lihat [Perlindungan Infrastruktur](#) di Pilar Keamanan AWS Kerangka Kerja yang Diarsiteksikan dengan Baik.

Anda menggunakan panggilan API AWS yang dipublikasikan untuk mengakses AWS Serverless Application Repository melalui jaringan. Klien harus mendukung hal berikut:

- Transport Layer Security (TLS). Kami membutuhkan TLS 1.2 dan merekomendasikan TLS 1.3.
- Suite cipher dengan kerahasiaan maju sempurna (PFS) seperti DHE (Ephemeral Diffie-Hellman) atau ECDHE (Elliptic Curve Ephemeral Diffie-Hellman). Sebagian besar sistem modern seperti Java 7 dan sistem yang lebih baru mendukung mode ini.

Selain itu, permintaan harus ditandatangani menggunakan access key ID dan secret access key yang terkait dengan principal IAM. Atau Anda bisa menggunakan [AWS Security Token Service](#) (AWS STS) untuk membuat kredensial keamanan sementara guna menandatangani permintaan.

Kuota AWS Serverless Application Repository

Parameter AWS Serverless Application Repository memiliki kuota untuk jumlah aplikasi publik yang AWS Akun dapat memiliki di setiap AWS Wilayah. Kuota ini berlaku per Wilayah dan dapat ditingkatkan. Untuk meminta kenaikan, gunakan [konsol Pusat Dukungan](#).

Sumber Daya	Kuota Default
Aplikasi publik (per AWS Akun per AWS Wilayah)	100

Kuota berikut berlaku untuk penyimpanan yang tersedia untuk paket kode dan kebijakan aplikasi. Anda tidak dapat mengubah kuota ini.

Resource	Kuota
Penyimpanan Amazon S3 gratis untuk paket kode (per AWS Akun per AWS Wilayah)	5 GB
Panjang kebijakan aplikasi	6.144 karakter

Memecahkan masalah AWS Serverless Application Repository

Saat Anda menggunakan AWS Serverless Application Repository, Anda mungkin mengalami masalah saat membuat, memperbarui, atau menghapus aplikasi. Gunakan bagian ini untuk membantu memecahkan masalah umum yang mungkin Anda temui. Anda juga dapat mencari jawaban dan memposting pertanyaan di [forum AWS Serverless Application Repository](#).

Note

Aplikasi di AWS Serverless Application Repository dikerahkan dengan menggunakan AWS CloudFormation. Untuk informasi tentang pemecahan masalah AWS CloudFormation masalah, lihat [AWS CloudFormation Panduan pemecahan masalah](#).

Topik

- [Anda Tidak Bisa Membuat Aplikasi Publik](#)
- [Kuota Terlampaui](#)
- [File Readme yang Diperbarui Tidak Muncul Segera](#)
- [Anda Tidak Dapat Menyebarkan Aplikasi Karena Izin IAM Tidak Cukup](#)
- [Anda Tidak Dapat Menyebarkan Aplikasi yang Sama Dua Kali](#)
- [Mengapa Aplikasi Saya Tidak Tersedia untuk Umum](#)
- [Menghubungi Support](#)

Anda Tidak Bisa Membuat Aplikasi Publik

Jika Anda tidak dapat membuat aplikasi Anda publik, Anda mungkin kehilangan file lisensi untuk aplikasi Anda yang disetujui oleh Open Source Initiative (OSI).

Untuk membuat aplikasi Anda publik, Anda memerlukan file lisensi yang disetujui OSI, dan juga versi aplikasi yang berhasil dipublikasikan dengan URL kode sumber untuk versi tersebut. Anda tidak dapat memperbarui lisensi aplikasi setelah aplikasi dibuat.

Jika Anda tidak dapat membuat aplikasi Anda publik karena Anda kehilangan file lisensi, menghapus aplikasi dan membuat yang baru dengan nama yang sama. Pastikan Anda memberikannya satu atau lebih lisensi open-source yang disetujui oleh organisasi Open Source Initiative (OSI).

Kuota Terlampaui

Jika Anda menerima pesan galat yang menunjukkan bahwa kuota terlampaui, periksa apakah Anda mencapai kuota sumber daya. Untuk AWS Serverless Application Repository kuota, lihat [Kuota AWS Serverless Application Repository](#).

File Readme yang Diperbarui Tidak Muncul Segera

Ketika Anda membuat aplikasi Anda publik, isi aplikasi Anda dapat memakan waktu hingga 24 jam untuk diperbarui. Jika Anda mengalami penundaan lebih dari 24 jam, coba hubungi AWS Support untuk bantuan. Untuk detailnya, lihat berikut.

Anda Tidak Dapat Menyebarkan Aplikasi Karena Izin IAM Tidak Cukup

Untuk menggunakan AWS Serverless Application Repository aplikasi, Anda memerlukan izin untuk AWS Serverless Application Repository sumber daya dan AWS CloudFormation tumpukan. Anda mungkin juga memerlukan izin untuk menggunakan layanan dasar yang dijelaskan dalam aplikasi. Misalnya, jika Anda membuat bucket Amazon S3 atau tabel Amazon DynamoDB, Anda perlu izin untuk Amazon S3 atau DynamoDB.

Jika Anda mengalami masalah jenis ini, tinjau AWS Identity and Access Management (IAM) kebijakan dan verifikasi bahwa Anda memiliki izin yang diperlukan. Untuk informasi lebih lanjut, lihat [Mengendalikan Akses dengan Identity and Access Management AWS](#).

Anda Tidak Dapat Menyebarkan Aplikasi yang Sama Dua Kali

Nama aplikasi yang Anda berikan digunakan sebagai nama AWS CloudFormation tumpukan. Jika Anda memiliki masalah saat menerapkan aplikasi, pastikan Anda belum memiliki AWS CloudFormation tumpukan dengan nama yang sama. Jika Anda melakukannya, berikan nama aplikasi yang berbeda atau hapus tumpukan yang ada untuk menyebarkan aplikasi dengan nama yang sama.

Mengapa Aplikasi Saya Tidak Tersedia untuk Umum

Aplikasi bersifat privat secara default. Untuk membuat aplikasi Anda publik, ikuti langkah-langkah [kemari](#).

Menghubungi Support

Dalam beberapa kasus, Anda mungkin tidak dapat menemukan solusi pemecahan masalah di bagian ini atau melalui [AWS Serverless Application Repository forum](#). Jika Anda AWS Support Premium, Anda dapat membuat kasus dukungan teknis di [AWS Dukungan](#).

Sebelum Anda menghubungi AWS Support, pastikan untuk mendapatkan Amazon Resource Name (ARN) untuk aplikasi yang Anda miliki. Anda dapat menemukan aplikasi ARN di [AWS Serverless Application Repository konsol](#).

Operasi

API REST AWS Serverless Application Repository mencakup operasi berikut.

- [CreateApplication](#)

Membuat aplikasi, opsional termasuk AWS File SAM untuk membuat versi aplikasi pertama dalam panggilan yang sama.

- [CreateApplicationVersion](#)

Membuat versi aplikasi.

- [CreateCloudFormationChangeSet](#)

Membuat AWS CloudFormation perubahan set untuk aplikasi yang diberikan.

- [CreateCloudFormationTemplate](#)

Membuat AWS CloudFormation templat.

- [DeleteApplication](#)

Menghapus aplikasi yang ditentukan.

- [GetApplication](#)

Mendapat aplikasi yang ditentukan.

- [GetApplicationPolicy](#)

Mengambil kebijakan untuk aplikasi.

- [GetCloudFormationTemplate](#)

Mendapat ditentukan AWS CloudFormation templat.

- [ListApplicationDependencies](#)

Mengambil daftar aplikasi bersarang di aplikasi yang mengandung.

- [ListApplications](#)

Daftar aplikasi yang dimiliki oleh pemohon.

- [ListApplicationVersions](#)

Daftar versi untuk aplikasi yang ditentukan.

- [PutApplicationPolicy](#)

Menetapkan kebijakan izin untuk aplikasi. Untuk daftar tindakan yang didukung untuk operasi ini, lihat [Izin Aplikasi](#).

- [UnshareApplication](#)

Unshares aplikasi dari AWS Organisasi.

Operasi ini hanya dapat dipanggil dari akun manajemen organisasi.

- [UpdateApplication](#)

Memperbarui aplikasi yang ditentukan.

Sumber daya

AWS Serverless Application Repository REST API mencakup sumber daya berikut.

Topik

- [Applications](#)
- [ApplicationId Aplikasi](#)
- [Aplikasi ApplicationId Changesets](#)
- [Applications applicationId Dependencies](#)
- [Kebijakan ApplicationID Aplikasi](#)
- [Applications applicationId Templates](#)
- [Applications applicationId Templates templateId](#)
- [Applications applicationId Unshare](#)
- [Versi ApplicationId Aplikasi](#)
- [Aplikasi ApplicationID Versi SemanticVersion](#)

Applications

URI

/applications

Metode HTTP

GET

ID Operasi: ListApplications

Daftar aplikasi yang dimiliki oleh pemohon.

Parameter kueri

Nama	Tipe	Diperlukan	Deskripsi
MaxItems	String	False	Jumlah total item yang akan dikembalikan.
nextToken	String	False	Token untuk menentukan di mana harus memulai paginating.

Respons

Kode status	Model respons	Deskripsi
200	ApplicationPage	Berhasil
400	BadRequestException	Salah satu parameter dalam permintaan tidak valid.
403	ForbiddenException	Klien tidak diautentikasi.
404	NotFoundException	Sumber daya (misalnya, pernyataan kebijakan akses) yang ditentukan dalam permintaan tidak ada.
500	InternalServerErrorException	Layanan AWS Serverless Application Repository mengalami kesalahan internal.

POST

ID Operasi: CreateApplication

Membuat aplikasi, secara opsional termasuk file AWS SAM untuk membuat versi aplikasi pertama dalam panggilan yang sama.

Respons

Kode status	Model respons	Deskripsi
201	Application	Berhasil
400	BadRequestException	Salah satu parameter dalam permintaan tidak valid.
403	ForbiddenException	Klien tidak diautentikasi.
409	ConflictException	Sumber daya sudah ada.
429	TooManyRequestsException	Klien mengirim lebih dari jumlah permintaan yang diizinkan per unit waktu.
500	InternalServerErrorException	Layanan AWS Serverless Application Repository mengalami kesalahan internal.

OPTIONS

Respons

Kode status	Model respons	Deskripsi
200	None	200 respon

Skema

Permintaan badan

POSTskema

```
{
  "name": "string",
  "description": "string",
  "author": "string",
  "spdxLicenseId": "string",
```

```
"licenseBody": "string",
"licenseUrl": "string",
"readmeBody": "string",
"readmeUrl": "string",
"labels": [
  "string"
],
"homePageUrl": "string",
"semanticVersion": "string",
"templateBody": "string",
"templateUrl": "string",
"sourceCodeUrl": "string",
"sourceCodeArchiveUrl": "string"
}
```

Badan respons

ApplicationPageskema

```
{
  "applications": [
    {
      "applicationId": "string",
      "name": "string",
      "description": "string",
      "author": "string",
      "spdxLicenseId": "string",
      "labels": [
        "string"
      ],
      "creationTime": "string",
      "homePageUrl": "string"
    }
  ],
  "nextToken": "string"
}
```

Applicationskema

```
{
  "applicationId": "string",
  "name": "string",
}
```

```
"description": "string",
"author": "string",
"isVerifiedAuthor": boolean,
"verifiedAuthorUrl": "string",
"spdxLicenseId": "string",
"licenseUrl": "string",
"readmeUrl": "string",
"labels": [
  "string"
],
"creationTime": "string",
"homePageUrl": "string",
"version": {
  "applicationId": "string",
  "semanticVersion": "string",
  "sourceCodeUrl": "string",
  "sourceCodeArchiveUrl": "string",
  "templateUrl": "string",
  "creationTime": "string",
  "parameterDefinitions": [
    {
      "name": "string",
      "defaultValue": "string",
      "description": "string",
      "type": "string",
      "noEcho": boolean,
      "allowedPattern": "string",
      "constraintDescription": "string",
      "minValue": integer,
      "maxValue": integer,
      "minLength": integer,
      "maxLength": integer,
      "allowedValues": [
        "string"
      ],
    }
  ],
  "referencedByResources": [
    "string"
  ]
}
],
"requiredCapabilities": [
  enum
],
"resourcesSupported": boolean
```

```
}  
}
```

BadRequestExceptionskema

```
{  
  "message": "string",  
  "errorCode": "string"  
}
```

ForbiddenExceptionskema

```
{  
  "message": "string",  
  "errorCode": "string"  
}
```

NotFoundExceptionskema

```
{  
  "message": "string",  
  "errorCode": "string"  
}
```

ConflictExceptionskema

```
{  
  "message": "string",  
  "errorCode": "string"  
}
```

TooManyRequestsExceptionskema

```
{  
  "message": "string",  
  "errorCode": "string"  
}
```

InternalServerErrorExceptionskema

```
{  
  "message": "string",  
  "errorCode": "string"  
}
```

Properti

Application

Detail tentang aplikasi.

applicationId

Aplikasi Amazon Resource Name (ARN).

Jenis: string

Diperlukan: Benar

name

Nama aplikasi.

Panjang minimum=1. Panjang maksimum = 140

Pola: "[A-Za-Z0-9\ -] +";

Jenis: string

Diperlukan: Benar

description

Deskripsi aplikasi.

Panjang minimum=1. Panjang maksimum = 256

Jenis: string

Diperlukan: Benar

author

Nama penulis yang menerbitkan aplikasi.

Panjang minimum=1. Panjang maksimum=127.

Pola “^ [a-z0-9] (([a-z0-9] | - (?! -)) * [a-z0-9])? \$”;

Jenis: string

Diperlukan: Benar

isVerifiedAuthor

Menentukan apakah penulis aplikasi ini telah diverifikasi. Ini berarti bahwa AWS telah melakukan tinjauan itikad baik, sebagai penyedia layanan yang masuk akal dan bijaksana, atas informasi yang diberikan oleh pemohon dan telah mengkonfirmasi bahwa identitas pemohon adalah seperti yang diklaim.

Jenis: boolean

Diperlukan: Salah

verifiedAuthorUrl

URL ke profil publik penulis terverifikasi. URL ini dikirimkan oleh penulis.

Jenis: string

Diperlukan: Salah

spdxLicenseId

Pengidentifikasi yang valid dari <https://spdx.org/licenses/>.

Jenis: string

Diperlukan: Salah

licenseUrl

Tautan ke file lisensi aplikasi yang cocok dengan nilai SPDXLicenseId aplikasi Anda.

Ukuran maksimal 5 MB

Jenis: string

Diperlukan: Salah

readmeUrl

Tautan ke file readme dalam bahasa Markdown yang berisi deskripsi aplikasi yang lebih rinci dan cara kerjanya.

Ukuran maksimal 5 MB

Jenis: string

Diperlukan: Salah

labels

Label untuk meningkatkan penemuan aplikasi di hasil penelusuran.

Panjang minimum=1. Panjang maksimum=127. Jumlah label maksimum: 10

Pola: “`^[A-Za-Z0-9+\\ -_:\\/@] +$`”;

Jenis: Array tipe string

Diperlukan: Salah

creationTime

Tanggal dan waktu sumber daya ini dibuat.

Jenis: string

Diperlukan: Salah

homePageUrl

URL dengan informasi lebih lanjut tentang aplikasi, misalnya lokasi GitHub repositori Anda untuk aplikasi.

Jenis: string

Diperlukan: Salah

version

Informasi versi tentang aplikasi.

Jenis: [Versi](#)

Diperlukan: Salah

ApplicationPage

Daftar detail aplikasi.

applications

Sebuah array ringkasan aplikasi.

Jenis: Array tipe [ApplicationSummary](#)

Diperlukan: Benar

nextToken

Token untuk meminta halaman hasil berikutnya.

Jenis: string

Diperlukan: Salah

ApplicationSummary

Ringkasan detail tentang aplikasi.

applicationId

Aplikasi Amazon Resource Name (ARN).

Jenis: string

Diperlukan: Benar

name

Nama aplikasi.

Panjang minimum=1. Panjang maksimum = 140

Pola: “[A-Za-Z0-9\ -]+”;

Jenis: string

Diperlukan: Benar

description

Deskripsi aplikasi.

Panjang minimum=1. Panjang maksimum = 256

Jenis: string

Diperlukan: Benar

author

Nama penulis yang menerbitkan aplikasi.

Panjang minimum=1. Panjang maksimum=127.

Pola “^ [a-z0-9] ([a-z0-9] | - (?! -)) * [a-z0-9]? \$”;

Jenis: string

Diperlukan: Benar

spdxLicenseId

Pengidentifikasi yang valid dari <https://spdx.org/licenses/>.

Jenis: string

Diperlukan: Salah

labels

Label untuk meningkatkan penemuan aplikasi di hasil penelusuran.

Panjang minimum=1. Panjang maksimum=127. Jumlah label maksimum: 10

Pola: “^ [A-Za-Z0-9+\ -_:\ /@] +\$”;

Jenis: Array tipe string

Diperlukan: Salah

creationTime

Tanggal dan waktu sumber daya ini dibuat.

Jenis: string

Diperlukan: Salah

homePageUrl

URL dengan informasi lebih lanjut tentang aplikasi, misalnya lokasi GitHub repositori Anda untuk aplikasi.

Jenis: string

Diperlukan: Salah

BadRequestException

Salah satu parameter dalam permintaan tidak valid.

message

Salah satu parameter dalam permintaan tidak valid.

Jenis: string

Diperlukan: Salah

errorCode

400

Jenis: string

Diperlukan: Salah

Capability

Nilai yang harus ditentukan untuk menyebarkan beberapa aplikasi.

CAPABILITY_IAM
CAPABILITY_NAMED_IAM
CAPABILITY_AUTO_EXPAND
CAPABILITY_RESOURCE_POLICY

ConflictException

Sumber daya sudah ada.

message

Sumber daya sudah ada.

Jenis: string

Diperlukan: Salah

errorCode

409

Jenis: string

Diperlukan: Salah

CreateApplicationInput

Buat permintaan aplikasi.

name

Nama aplikasi yang ingin Anda terbitkan.

Panjang minimum=1. Panjang maksimum = 140

Pola: “[A-Za-Z0-9\ -]+”;

Jenis: string

Diperlukan: Benar

description

Deskripsi aplikasi.

Panjang minimum=1. Panjang maksimum = 256

Jenis: string

Diperlukan: Benar

author

Nama penulis yang menerbitkan aplikasi.

Panjang minimum=1. Panjang maksimum=127.

Pola “^ [a-z0-9] (([a-z0-9] |- (?! -)) * [a-z0-9])? \$”;

Jenis: string

Diperlukan: Benar

spdxLicenseId

Pengidentifikasi yang valid dari <https://spdx.org/licenses/>.

Jenis: string

Diperlukan: Salah

licenseBody

File teks lokal yang berisi lisensi aplikasi yang cocok dengan nilai SPDXLicenseId aplikasi Anda. File tersebut memiliki format `file://<path>/<filename>`.

Ukuran maksimal 5 MB

Anda dapat menentukan hanya satu dari `licenseBody` dan `licenseUrl`; jika tidak, hasil kesalahan.

Jenis: string

Diperlukan: Salah

licenseUrl

Tautan ke objek S3 yang berisi lisensi aplikasi yang cocok dengan nilai `SPDXLicenseId` aplikasi Anda.

Ukuran maksimal 5 MB

Anda dapat menentukan hanya satu dari `licenseBody` dan `licenseUrl`; jika tidak, hasil kesalahan.

Jenis: string

Diperlukan: Salah

readmeBody

File readme teks lokal dalam bahasa Markdown yang berisi deskripsi aplikasi yang lebih rinci dan cara kerjanya. File tersebut memiliki format `file://<path>/<filename>`.

Ukuran maksimal 5 MB

Anda dapat menentukan hanya satu dari `readmeBody` dan `readmeUrl`; jika tidak, hasil kesalahan.

Jenis: string

Diperlukan: Salah

readmeUrl

Tautan ke objek S3 dalam bahasa Markdown yang berisi deskripsi aplikasi yang lebih rinci dan cara kerjanya.

Ukuran maksimal 5 MB

Anda dapat menentukan hanya satu dari `readmeBody` dan `readmeUrl`; jika tidak, hasil kesalahan.

Jenis: string

Diperlukan: Salah

labels

Label untuk meningkatkan penemuan aplikasi di hasil penelusuran.

Panjang minimum=1. Panjang maksimum=127. Jumlah label maksimum: 10

Pola: “`^[A-Za-Z0-9+\\ -_:\\ /@] +$`”;

Jenis: Array tipe string

Diperlukan: Salah

homePageUrl

URL dengan informasi lebih lanjut tentang aplikasi, misalnya lokasi GitHub repositori Anda untuk aplikasi.

Jenis: string

Diperlukan: Salah

semanticVersion

Versi semantik aplikasi:

<https://semver.org/>

Jenis: string

Diperlukan: Salah

templateBody

File AWS SAM template paket mentah lokal dari aplikasi Anda. File tersebut memiliki format `file://<path>/<filename>`.

Anda hanya dapat menentukan salah satu dari `templateBody` dan `templateUrl`; jika tidak, hasil kesalahan.

Jenis: string

Diperlukan: Salah

templateUrl

Sebuah link ke objek S3 yang berisi AWS SAM template paket aplikasi Anda.

Anda hanya dapat menentukan salah satu dari `templateBody` dan `templateUrl`; jika tidak, hasil kesalahan.

Jenis: string

Diperlukan: Salah

`sourceCodeUrl`

Tautan ke repositori publik untuk kode sumber aplikasi Anda, misalnya URL dari komit tertentu GitHub .

Jenis: string

Diperlukan: Salah

`sourceCodeArchiveUrl`

Tautan ke objek S3 yang berisi arsip ZIP dari kode sumber untuk versi aplikasi Anda ini.

Ukuran maksimal 50 MB

Jenis: string

Diperlukan: Salah

`ForbiddenException`

Klien tidak diautentikasi.

`message`

Klien tidak diautentikasi.

Jenis: string

Diperlukan: Salah

`errorCode`

403

Jenis: string

Diperlukan: Salah

InternalServerErrorException

Layanan AWS Serverless Application Repository mengalami kesalahan internal.

message

Layanan AWS Serverless Application Repository mengalami kesalahan internal.

Jenis: string

Diperlukan: Salah

errorCode

500

Jenis: string

Diperlukan: Salah

NotFoundException

Sumber daya (misalnya, pernyataan kebijakan akses) yang ditentukan dalam permintaan tidak ada.

message

Sumber daya (misalnya, pernyataan kebijakan akses) yang ditentukan dalam permintaan tidak ada.

Jenis: string

Diperlukan: Salah

errorCode

404

Jenis: string

Diperlukan: Salah

ParameterDefinition

Parameter yang didukung oleh aplikasi.

name

Nama parameter.

Jenis: string

Diperlukan: Benar

defaultValue

Nilai tipe yang sesuai untuk templat untuk digunakan jika tidak ada nilai yang ditentukan ketika tumpukan dibuat. Jika Anda menentukan kendala untuk parameter, Anda harus menentukan nilai yang mematuhi kendala tersebut.

Jenis: string

Diperlukan: Salah

description

Sebuah string hingga 4.000 karakter yang menggambarkan parameter.

Jenis: string

Diperlukan: Salah

type

Jenis parameter.

Nilai yang valid: `String` | `Number` | `List<Number>` | `CommaDelimitedList`

`String`: Sebuah string literal.

Misalnya, pengguna dapat menentukan `"MyUserName"`.

`Number`: Sebuah integer atau float. AWS CloudFormation memvalidasi nilai parameter sebagai angka. Namun, ketika Anda menggunakan parameter di tempat lain di template Anda (misalnya, dengan menggunakan fungsi Ref intrinsik), nilai parameter menjadi string.

Misalnya, pengguna mungkin menentukan `"8888"`.

`List<Number>`: Sebuah array bilangan bulat atau float yang dipisahkan oleh koma. AWS CloudFormation memvalidasi nilai parameter sebagai angka. Namun, ketika Anda menggunakan

parameter di tempat lain di template Anda (misalnya, dengan menggunakan fungsi Ref intrinsik), nilai parameter menjadi daftar string.

Misalnya, pengguna mungkin menentukan "80,20", dan kemudian Ref menghasilkan. ["80", "20"]

CommaDelimitedList: Sebuah array string literal yang dipisahkan oleh koma. Jumlah total string harus satu lebih dari jumlah koma. Juga, setiap string anggota dipangkas spasi.

Misalnya, pengguna mungkin menentukan "test, dev, prod", dan kemudian Ref menghasilkan. ["test", "dev", "prod"]

Jenis: string

Diperlukan: Salah

noEcho

Apakah akan menutupi nilai parameter setiap kali ada yang membuat panggilan yang menjelaskan tumpukan. Jika Anda menetapkan nilai ke true, nilai parameter ditutupi dengan tanda bintang (*****).

Jenis: boolean

Diperlukan: Salah

allowedPattern

Suatu ekspresi reguler yang mewakili pola untuk memungkinkan String tipe.

Jenis: string

Diperlukan: Salah

constraintDescription

Suatu string yang menjelaskan kendala ketika ia dilanggar. Sebagai contoh, tanpa deskripsi kendala, parameter yang memiliki pola yang diizinkan [A-Za-z0-9]+ menampilkan pesan kesalahan berikut ketika pengguna menentukan nilai yang tidak valid:

```
Malformed input-Parameter MyParameter must match pattern [A-Za-z0-9]+
```

Dengan menambahkan deskripsi kendala, seperti "harus berisi hanya huruf besar dan kecil dan angka," Anda dapat menampilkan pesan kesalahan yang disesuaikan berikut:

Malformed input-Parameter MyParameter must contain only uppercase and lowercase letters and numbers.

Jenis: string

Diperlukan: Salah

minValue

Nilai numerik yang menentukan nilai numerik terkecil yang ingin Anda izinkan untuk Number tipe.

Jenis: bilangan bulat

Diperlukan: Salah

maxValue

Nilai numerik yang menentukan nilai numerik terbesar yang ingin Anda izinkan untuk Number tipe.

Jenis: bilangan bulat

Diperlukan: Salah

minLength

Nilai integer yang menentukan jumlah karakter terkecil yang ingin Anda izinkan untuk String tipe.

Jenis: bilangan bulat

Diperlukan: Salah

maxLength

Nilai integer yang menentukan jumlah karakter terbesar yang ingin Anda izinkan untuk String tipe.

Jenis: bilangan bulat

Diperlukan: Salah

allowedValues

Array yang berisi daftar nilai yang diizinkan untuk parameter.

Jenis: Array tipe string

Diperlukan: Salah

referencedByResources

Daftar AWS SAM sumber daya yang menggunakan parameter ini.

Jenis: Array tipe string

Diperlukan: Benar

TooManyRequestsException

Klien mengirim lebih dari jumlah permintaan yang diizinkan per unit waktu.

message

Klien mengirim lebih dari jumlah permintaan yang diizinkan per unit waktu.

Jenis: string

Diperlukan: Salah

errorCode

429

Jenis: string

Diperlukan: Salah

Version

Detail versi aplikasi.

applicationId

Aplikasi Amazon Resource Name (ARN).

Jenis: string

Diperlukan: Benar

semanticVersion

Versi semantik aplikasi:

<https://semver.org/>

Jenis: string

Diperlukan: Benar

sourceCodeUrl

Tautan ke repositori publik untuk kode sumber aplikasi Anda, misalnya URL dari komit tertentu GitHub .

Jenis: string

Diperlukan: Salah

sourceCodeArchiveUrl

Tautan ke objek S3 yang berisi arsip ZIP dari kode sumber untuk versi aplikasi Anda ini.

Ukuran maksimal 50 MB

Jenis: string

Diperlukan: Salah

templateUrl

Sebuah link ke AWS SAM template paket aplikasi Anda.

Jenis: string

Diperlukan: Benar

creationTime

Tanggal dan waktu sumber daya ini dibuat.

Jenis: string

Diperlukan: Benar

parameterDefinitions

Array tipe parameter yang didukung oleh aplikasi.

Jenis: Array tipe [ParameterDefinition](#)

Diperlukan: Benar

requiredCapabilities

Daftar nilai yang harus Anda tentukan sebelum Anda dapat menyebarkan aplikasi tertentu. Beberapa aplikasi mungkin menyertakan sumber daya yang dapat memengaruhi izin di AWS akun Anda, misalnya, dengan membuat pengguna new AWS Identity and Access Management (IAM). Untuk aplikasi tersebut, Anda harus secara eksplisit mengakui kemampuannya dengan menentukan parameter ini.

Satu-satunya nilai yang valid

adalah `CAPABILITY_IAM`, `CAPABILITY_NAMED_IAM`, `CAPABILITY_RESOURCE_POLICY`, dan `CAPABILITY_AUTO_EXPAND`.

Sumber daya berikut mengharuskan Anda untuk menentukan `CAPABILITY_IAM` atau `CAPABILITY_NAMED_IAM`: [AWS::IAM::Group](#), [AWS::IAM::InstanceProfile](#), [AWS::IAM::Policy](#), dan [AWS::IAM::Role](#). Jika aplikasi berisi sumber daya IAM, Anda dapat menentukan salah satu `CAPABILITY_IAM` atau `CAPABILITY_NAMED_IAM`. Jika aplikasi berisi sumber daya IAM dengan nama khusus, Anda harus menentukan `CAPABILITY_NAMED_IAM`.

Sumber daya berikut mengharuskan Anda untuk menentukan `CAPABILITY_RESOURCE_POLICY`: [AWS::Lambda::Permission](#), [AWS::IAM::Policy](#), [AWS::ApplicationAutoScaling::ScalingPolicy](#), [AWS::S3::BucketPolicy](#) dan [AWS::SQS::QueuePolicy](#) [AWS::SNS::TopicPolicy](#)

Aplikasi yang berisi satu atau lebih aplikasi bersarang mengharuskan Anda untuk menentukan `CAPABILITY_AUTO_EXPAND`.

Jika template aplikasi Anda berisi salah satu sumber daya di atas, kami sarankan Anda meninjau semua izin yang terkait dengan aplikasi sebelum menerapkan. Jika Anda tidak menentukan parameter ini untuk aplikasi yang membutuhkan kemampuan, panggilan akan gagal.

Jenis: Array tipe [Capability](#)

Diperlukan: Benar

resourcesSupported

Apakah semua sumber AWS daya yang terkandung dalam aplikasi ini didukung di wilayah di mana ia sedang diambil.

Jenis: boolean

Diperlukan: Benar

Lihat juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK dan referensi khusus bahasa, lihat berikut ini:

ListApplications

- [Antarmuka Baris Perintah AWS](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK untuk V3 JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK untuk Python](#)
- [AWS SDK for Ruby V3](#)

CreateApplication

- [Antarmuka Baris Perintah AWS](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK untuk V3 JavaScript](#)
- [AWS SDK for PHP V3](#)

- [AWS SDK untuk Python](#)
- [AWS SDK for Ruby V3](#)

ApplicationId Aplikasi

URI

/applications/*applicationId*

Metode HTTP

GET

ID Operasi: GetApplication

Mendapat aplikasi yang ditentukan.

Parameter jalur

Nama	Tipe	Diperlukan	Deskripsi
<i>applicationId</i>	String	True	Amazon Resource Name (ARN) aplikasi.

Parameter kueri

Nama	Tipe	Diperlukan	Deskripsi
SemanticVersion	String	False	Versi semantik aplikasi untuk mendapatkan.

Respons

Kode status	Model respons	Deskripsi
200	Application	Berhasil

Kode status	Model respons	Deskripsi
400	<u>BadRequestException</u>	Salah satu parameter dalam permintaan tidak valid.
403	<u>ForbiddenException</u>	Klien tidak diautentikasi.
404	<u>NotFoundException</u>	Sumber daya (misalnya, pernyataan kebijakan akses) yang ditentukan dalam permintaan tidak ada.
429	<u>TooManyRequestsException</u>	Klien mengirim lebih dari jumlah permintaan yang diizinkan per unit waktu.
500	<u>InternalServerErrorException</u>	Layanan AWS Serverless Application Repository mengalami kesalahan internal.

DELETE

ID Operasi: DeleteApplication

Menghapus aplikasi yang ditentukan.

Parameter jalur

Nama	Tipe	Diperlukan	Deskripsi
<i>applicationId</i>	String	True	Amazon Resource Name (ARN) aplikasi.

Respons

Kode status	Model respons	Deskripsi
204	None	Berhasil

Kode status	Model respons	Deskripsi
400	<u>BadRequestException</u>	Salah satu parameter dalam permintaan tidak valid.
403	<u>ForbiddenException</u>	Klien tidak diautentikasi.
404	<u>NotFoundException</u>	Sumber daya (misalnya, pernyataan kebijakan akses) yang ditentukan dalam permintaan tidak ada.
409	<u>ConflictException</u>	Sumber daya sudah ada.
429	<u>TooManyRequestsException</u>	Klien mengirim lebih dari jumlah permintaan yang diizinkan per unit waktu.
500	<u>InternalServerErrorException</u>	Layanan AWS Serverless Application Repository mengalami kesalahan internal.

OPTIONS

Parameter jalur

Nama	Tipe	Diperlukan	Deskripsi
<i>applicationId</i>	String	True	Amazon Resource Name (ARN) aplikasi.

Respons

Kode status	Model respons	Deskripsi
200	None	200 respon

PATCH

ID Operasi: UpdateApplication

Memperbarui aplikasi yang ditentukan.

Parameter jalur

Nama	Tipe	Diperlukan	Deskripsi
<i>applicationId</i>	String	True	Amazon Resource Name (ARN) aplikasi.

Respons

Kode status	Model respons	Deskripsi
200	Application	Berhasil
400	BadRequestException	Salah satu parameter dalam permintaan tidak valid.
403	ForbiddenException	Klien tidak diautentikasi.
404	NotFoundException	Sumber daya (misalnya, pernyataan kebijakan akses) yang ditentukan dalam permintaan tidak ada.
409	ConflictException	Sumber daya sudah ada.
429	TooManyRequestsException	Klien mengirim lebih dari jumlah permintaan yang diizinkan per unit waktu.
500	InternalServerErrorException	Layanan AWS Serverless Application Repository mengalami kesalahan internal.

Skema

Permintaan badan

PATCHskema

```
{
  "description": "string",
  "author": "string",
  "readmeBody": "string",
  "readmeUrl": "string",
  "labels": [
    "string"
  ],
  "homePageUrl": "string"
}
```

Badan respons

Applicationskema

```
{
  "applicationId": "string",
  "name": "string",
  "description": "string",
  "author": "string",
  "isVerifiedAuthor": boolean,
  "verifiedAuthorUrl": "string",
  "spdxLicenseId": "string",
  "licenseUrl": "string",
  "readmeUrl": "string",
  "labels": [
    "string"
  ],
  "creationTime": "string",
  "homePageUrl": "string",
  "version": {
    "applicationId": "string",
    "semanticVersion": "string",
    "sourceCodeUrl": "string",
    "sourceCodeArchiveUrl": "string",
    "templateUrl": "string",
  }
}
```

```
"creationTime": "string",
"parameterDefinitions": [
  {
    "name": "string",
    "defaultValue": "string",
    "description": "string",
    "type": "string",
    "noEcho": boolean,
    "allowedPattern": "string",
    "constraintDescription": "string",
    "minValue": integer,
    "maxValue": integer,
    "minLength": integer,
    "maxLength": integer,
    "allowedValues": [
      "string"
    ],
    "referencedByResources": [
      "string"
    ]
  }
],
"requiredCapabilities": [
  enum
],
"resourcesSupported": boolean
}
```

BadRequestExceptionkema

```
{
  "message": "string",
  "errorCode": "string"
}
```

ForbiddenExceptionkema

```
{
  "message": "string",
  "errorCode": "string"
}
```

```
}
```

NotFoundExceptionskema

```
{  
  "message": "string",  
  "errorCode": "string"  
}
```

ConflictExceptionskema

```
{  
  "message": "string",  
  "errorCode": "string"  
}
```

TooManyRequestsExceptionskema

```
{  
  "message": "string",  
  "errorCode": "string"  
}
```

InternalServerErrorExceptionskema

```
{  
  "message": "string",  
  "errorCode": "string"  
}
```

Properti

Application

Detail tentang aplikasi.

applicationId

Aplikasi Amazon Resource Name (ARN).

Jenis: string

Diperlukan: Benar

name

Nama aplikasi.

Panjang minimum=1. Panjang maksimum = 140

Pola: “[A-Za-Z0-9\ -]+”;

Jenis: string

Diperlukan: Benar

description

Deskripsi aplikasi.

Panjang minimum=1. Panjang maksimum = 256

Jenis: string

Diperlukan: Benar

author

Nama penulis yang menerbitkan aplikasi.

Panjang minimum=1. Panjang maksimum=127.

Pola “^ [a-z0-9] (([a-z0-9] | - (?! -)) * [a-z0-9])? \$”;

Jenis: string

Diperlukan: Benar

isVerifiedAuthor

Menentukan apakah penulis aplikasi ini telah diverifikasi. Ini berarti bahwa AWS telah melakukan tinjauan itikad baik, sebagai penyedia layanan yang masuk akal dan bijaksana, atas informasi yang diberikan oleh pemohon dan telah mengkonfirmasi bahwa identitas pemohon adalah seperti yang diklaim.

Jenis: boolean

Diperlukan: Salah

verifiedAuthorUrl

URL ke profil publik penulis terverifikasi. URL ini dikirimkan oleh penulis.

Jenis: string

Diperlukan: Salah

spdxLicenseId

Pengidentifikasi yang valid dari <https://spdx.org/licenses/>.

Jenis: string

Diperlukan: Salah

licenseUrl

Tautan ke file lisensi aplikasi yang cocok dengan nilai SPDXLicenseId aplikasi Anda.

Ukuran maksimal 5 MB

Jenis: string

Diperlukan: Salah

readmeUrl

Tautan ke file readme dalam bahasa Markdown yang berisi deskripsi aplikasi yang lebih rinci dan cara kerjanya.

Ukuran maksimal 5 MB

Jenis: string

Diperlukan: Salah

labels

Label untuk meningkatkan penemuan aplikasi di hasil penelusuran.

Panjang minimum=1. Panjang maksimum=127. Jumlah label maksimum: 10

Pola: “^ [A-Za-Z0-9+\\ -_:\\/@] +\$”;

Jenis: Array tipe string

Diperlukan: Salah

creationTime

Tanggal dan waktu sumber daya ini dibuat.

Jenis: string

Diperlukan: Salah

homePageUrl

URL dengan informasi lebih lanjut tentang aplikasi, misalnya lokasi GitHub repositori Anda untuk aplikasi.

Jenis: string

Diperlukan: Salah

version

Informasi versi tentang aplikasi.

Jenis: [Versi](#)

Diperlukan: Salah

BadRequestException

Salah satu parameter dalam permintaan tidak valid.

message

Salah satu parameter dalam permintaan tidak valid.

Jenis: string

Diperlukan: Salah

errorCode

400

Jenis: string

Diperlukan: Salah

Capability

Nilai yang harus ditentukan untuk menyebarkan beberapa aplikasi.

CAPABILITY_IAM

CAPABILITY_NAMED_IAM

CAPABILITY_AUTO_EXPAND

CAPABILITY_RESOURCE_POLICY

ConflictException

Sumber daya sudah ada.

message

Sumber daya sudah ada.

Jenis: string

Diperlukan: Salah

errorCode

409

Jenis: string

Diperlukan: Salah

ForbiddenException

Klien tidak diautentikasi.

message

Klien tidak diautentikasi.

Jenis: string

Diperlukan: Salah

errorCode

403

Jenis: string

Diperlukan: Salah

InternalServerErrorException

Layanan AWS Serverless Application Repository mengalami kesalahan internal.

message

Layanan AWS Serverless Application Repository mengalami kesalahan internal.

Jenis: string

Diperlukan: Salah

errorCode

500

Jenis: string

Diperlukan: Salah

NotFoundException

Sumber daya (misalnya, pernyataan kebijakan akses) yang ditentukan dalam permintaan tidak ada.

message

Sumber daya (misalnya, pernyataan kebijakan akses) yang ditentukan dalam permintaan tidak ada.

Jenis: string

Diperlukan: Salah

errorCode

404

Jenis: string

Diperlukan: Salah

ParameterDefinition

Parameter yang didukung oleh aplikasi.

name

Nama parameter.

Jenis: string

Diperlukan: Benar

defaultValue

Nilai tipe yang sesuai untuk templat untuk digunakan jika tidak ada nilai yang ditentukan ketika tumpukan dibuat. Jika Anda menentukan kendala untuk parameter, Anda harus menentukan nilai yang mematuhi kendala tersebut.

Jenis: string

Diperlukan: Salah

description

Sebuah string hingga 4.000 karakter yang menggambarkan parameter.

Jenis: string

Diperlukan: Salah

type

Jenis parameter.

Nilai yang valid: `String` | `Number` | `List<Number>` | `CommaDelimitedList`

`String`: Sebuah string literal.

Misalnya, pengguna dapat menentukan `"MyUserName"`.

`Number`: Sebuah integer atau float. AWS CloudFormation memvalidasi nilai parameter sebagai angka. Namun, ketika Anda menggunakan parameter di tempat lain di template Anda (misalnya, dengan menggunakan fungsi Ref intrinsik), nilai parameter menjadi string.

Misalnya, pengguna mungkin menentukan `"8888"`.

`List<Number>`: Sebuah array bilangan bulat atau float yang dipisahkan oleh koma. AWS CloudFormation memvalidasi nilai parameter sebagai angka. Namun, ketika Anda menggunakan parameter di tempat lain di template Anda (misalnya, dengan menggunakan fungsi Ref intrinsik), nilai parameter menjadi daftar string.

Misalnya, pengguna mungkin menentukan `"80,20"`, dan kemudian Ref menghasilkan `["80", "20"]`

`CommaDelimitedList`: Sebuah array string literal yang dipisahkan oleh koma. Jumlah total string harus satu lebih dari jumlah koma. Juga, setiap string anggota dipangkas spasi.

Misalnya, pengguna mungkin menentukan `"test, dev, prod"`, dan kemudian Ref menghasilkan `["test", "dev", "prod"]`

Jenis: string

Diperlukan: Salah

`noEcho`

Apakah akan menutupi nilai parameter setiap kali ada yang membuat panggilan yang menjelaskan tumpukan. Jika Anda menetapkan nilai ke true, nilai parameter ditutupi dengan tanda bintang (`*****`).

Jenis: boolean

Diperlukan: Salah

`allowedPattern`

Suatu ekspresi reguler yang mewakili pola untuk memungkinkan `String` tipe.

Jenis: string

Diperlukan: Salah

constraintDescription

Suatu string yang menjelaskan kendala ketika ia dilanggar. Sebagai contoh, tanpa deskripsi kendala, parameter yang memiliki pola yang diizinkan `[A-Za-z0-9]+` menampilkan pesan kesalahan berikut ketika pengguna menentukan nilai yang tidak valid:

```
Malformed input-Parameter MyParameter must match pattern [A-Za-z0-9]+
```

Dengan menambahkan deskripsi kendala, seperti “harus berisi hanya huruf besar dan kecil dan angka,” Anda dapat menampilkan pesan kesalahan yang disesuaikan berikut:

```
Malformed input-Parameter MyParameter must contain only uppercase and lowercase letters and numbers.
```

Jenis: string

Diperlukan: Salah

minValue

Nilai numerik yang menentukan nilai numerik terkecil yang ingin Anda izinkan untuk `Number` tipe.

Jenis: bilangan bulat

Diperlukan: Salah

maxValue

Nilai numerik yang menentukan nilai numerik terbesar yang ingin Anda izinkan untuk `Number` tipe.

Jenis: bilangan bulat

Diperlukan: Salah

minLength

Nilai integer yang menentukan jumlah karakter terkecil yang ingin Anda izinkan untuk `String` tipe.

Jenis: bilangan bulat

Diperlukan: Salah

maxLength

Nilai integer yang menentukan jumlah karakter terbesar yang ingin Anda izinkan untuk `String` tipe.

Jenis: bilangan bulat

Diperlukan: Salah

allowedValues

Array yang berisi daftar nilai yang diizinkan untuk parameter.

Jenis: Array tipe string

Diperlukan: Salah

referencedByResources

Daftar AWS SAM sumber daya yang menggunakan parameter ini.

Jenis: Array tipe string

Diperlukan: Benar

TooManyRequestsException

Klien mengirim lebih dari jumlah permintaan yang diizinkan per unit waktu.

message

Klien mengirim lebih dari jumlah permintaan yang diizinkan per unit waktu.

Jenis: string

Diperlukan: Salah

errorCode

429

Jenis: string

Diperlukan: Salah

UpdateApplicationInput

Perbarui permintaan aplikasi.

description

Deskripsi aplikasi.

Panjang minimum=1. Panjang maksimum = 256

Jenis: string

Diperlukan: Salah

author

Nama penulis yang menerbitkan aplikasi.

Panjang minimum=1. Panjang maksimum=127.

Pola “`^ [a-z0-9] (([a-z0-9] | - (?! -)) * [a-z0-9])? $`”;

Jenis: string

Diperlukan: Salah

readmeBody

File readme teks dalam bahasa Markdown yang berisi deskripsi aplikasi yang lebih rinci dan cara kerjanya.

Ukuran maksimal 5 MB

Jenis: string

Diperlukan: Salah

readmeUrl

Tautan ke file readme dalam bahasa Markdown yang berisi deskripsi aplikasi yang lebih rinci dan cara kerjanya.

Ukuran maksimal 5 MB

Jenis: string

Diperlukan: Salah

labels

Label untuk meningkatkan penemuan aplikasi di hasil penelusuran.

Panjang minimum=1. Panjang maksimum=127. Jumlah label maksimum: 10

Pola: “`^[A-Za-Z0-9+\\ -_:\\ /@] +$`”;

Jenis: Array tipe string

Diperlukan: Salah

homePageUrl

URL dengan informasi lebih lanjut tentang aplikasi, misalnya lokasi GitHub repositori Anda untuk aplikasi.

Jenis: string

Diperlukan: Salah

Version

Detail versi aplikasi.

applicationId

Aplikasi Amazon Resource Name (ARN).

Jenis: string

Diperlukan: Benar

semanticVersion

Versi semantik aplikasi:

<https://semver.org/>

Jenis: string

Diperlukan: Benar

sourceCodeUrl

Tautan ke repositori publik untuk kode sumber aplikasi Anda, misalnya URL dari komit tertentu GitHub .

Jenis: string

Diperlukan: Salah

sourceCodeArchiveUrl

Tautan ke objek S3 yang berisi arsip ZIP dari kode sumber untuk versi aplikasi Anda ini.

Ukuran maksimal 50 MB

Jenis: string

Diperlukan: Salah

templateUrl

Sebuah link ke AWS SAM template paket aplikasi Anda.

Jenis: string

Diperlukan: Benar

creationTime

Tanggal dan waktu sumber daya ini dibuat.

Jenis: string

Diperlukan: Benar

parameterDefinitions

Array tipe parameter yang didukung oleh aplikasi.

Jenis: Array tipe [ParameterDefinition](#)

Diperlukan: Benar

requiredCapabilities

Daftar nilai yang harus Anda tentukan sebelum Anda dapat menyebarkan aplikasi tertentu. Beberapa aplikasi mungkin menyertakan sumber daya yang dapat memengaruhi izin di AWS akun Anda, misalnya, dengan membuat pengguna new AWS Identity and Access Management (IAM). Untuk aplikasi tersebut, Anda harus secara eksplisit mengakui kemampuannya dengan menentukan parameter ini.

Satu-satunya nilai yang valid

adalah `CAPABILITY_IAM`, `CAPABILITY_NAMED_IAM`, `CAPABILITY_RESOURCE_POLICY`, dan `CAPABILITY_AUTO_EXPAND`.

Sumber daya berikut mengharuskan Anda untuk menentukan `CAPABILITY_IAM` atau `CAPABILITY_NAMED_IAM`: [AWS::IAM::Group](#), [AWS::IAM::InstanceProfile](#), [AWS::IAM::Policy](#), dan [AWS::IAM::Role](#). Jika aplikasi berisi sumber daya IAM, Anda dapat menentukan salah satu `CAPABILITY_IAM` atau `CAPABILITY_NAMED_IAM`. Jika aplikasi berisi sumber daya IAM dengan nama khusus, Anda harus menentukan `CAPABILITY_NAMED_IAM`.

Sumber daya berikut mengharuskan Anda untuk menentukan `CAPABILITY_RESOURCE_POLICY`: [AWS::Lambda::Permission](#), [AWS::IAM::Policy](#), [AWS::ApplicationAutoScaling::ScalingPolicy](#), [AWS::S3::BucketPolicy](#) dan [AWS::SQS::QueuePolicy](#) [AWS::SNS::TopicPolicy](#)

Aplikasi yang berisi satu atau lebih aplikasi bersarang mengharuskan Anda untuk menentukan `CAPABILITY_AUTO_EXPAND`.

Jika template aplikasi Anda berisi salah satu sumber daya di atas, kami sarankan Anda meninjau semua izin yang terkait dengan aplikasi sebelum menerapkan. Jika Anda tidak menentukan parameter ini untuk aplikasi yang membutuhkan kemampuan, panggilan akan gagal.

Jenis: Array tipe [Capability](#)

Diperlukan: Benar

resourcesSupported

Apakah semua sumber AWS daya yang terkandung dalam aplikasi ini didukung di wilayah di mana ia sedang diambil.

Jenis: boolean

Diperlukan: Benar

Lihat juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK dan referensi khusus bahasa, lihat berikut ini:

GetApplication

- [Antarmuka Baris Perintah AWS](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK untuk V3 JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK untuk Python](#)
- [AWS SDK for Ruby V3](#)

DeleteApplication

- [Antarmuka Baris Perintah AWS](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK untuk V3 JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK untuk Python](#)
- [AWS SDK for Ruby V3](#)

UpdateApplication

- [Antarmuka Baris Perintah AWS](#)

- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK untuk V3 JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK untuk Python](#)
- [AWS SDK for Ruby V3](#)

Aplikasi ApplicationId Changesets

URI

/applications/*applicationId*/changesets

Metode HTTP

POST

ID Operasi: CreateCloudFormationChangeSet

Membuat set AWS CloudFormation perubahan untuk aplikasi yang diberikan.

Parameter jalur

Nama	Tipe	Diperlukan	Deskripsi
<i>applicationId</i>	String	True	Amazon Resource Name (ARN) aplikasi.

Respons

Kode status	Model respons	Deskripsi
201	ChangeSetDetails	Berhasil

Kode status	Model respons	Deskripsi
400	BadRequestException	Salah satu parameter dalam permintaan tidak valid.
403	ForbiddenException	Klien tidak diautentikasi.
429	TooManyRequestsException	Klien mengirim lebih dari jumlah permintaan yang diizinkan per unit waktu.
500	InternalServerErrorException	Layanan AWS Serverless Application Repository mengalami kesalahan internal.

OPTIONS

Parameter jalur

Nama	Tipe	Diperlukan	Deskripsi
<i>applicationId</i>	String	True	Amazon Resource Name (ARN) aplikasi.

Respons

Kode status	Model respons	Deskripsi
200	None	200 respon

Skema

Permintaan badan

POSTskema

```
{
  "stackName": "string",
```

```
"semanticVersion": "string",
"templateId": "string",
"parameterOverrides": [
  {
    "name": "string",
    "value": "string"
  }
],
"capabilities": [
  "string"
],
"changeSetName": "string",
"clientToken": "string",
"description": "string",
"notificationArns": [
  "string"
],
"resourceTypes": [
  "string"
],
"rollbackConfiguration": {
  "rollbackTriggers": [
    {
      "arn": "string",
      "type": "string"
    }
  ],
  "monitoringTimeInMinutes": integer
},
"tags": [
  {
    "key": "string",
    "value": "string"
  }
]
}
```

Badan respons

ChangeSetDetailsskema

```
{
  "applicationId": "string",
```



```
"semanticVersion": "string",  
"changeSetId": "string",  
"stackId": "string"  
}
```

BadRequestExceptionkema

```
{  
  "message": "string",  
  "errorCode": "string"  
}
```

ForbiddenExceptionkema

```
{  
  "message": "string",  
  "errorCode": "string"  
}
```

TooManyRequestsExceptionkema

```
{  
  "message": "string",  
  "errorCode": "string"  
}
```

InternalServerErrorExceptionkema

```
{  
  "message": "string",  
  "errorCode": "string"  
}
```

Properti

BadRequestException

Salah satu parameter dalam permintaan tidak valid.

message

Salah satu parameter dalam permintaan tidak valid.

Jenis: string

Diperlukan: Salah

errorCode

400

Jenis: string

Diperlukan: Salah

ChangeSetDetails

Detail dari set perubahan.

applicationId

Aplikasi Amazon Resource Name (ARN).

Jenis: string

Diperlukan: Benar

semanticVersion

Versi semantik aplikasi:

<https://semver.org/>

Jenis: string

Diperlukan: Benar

changeSetId

Nama Sumber Daya Amazon (ARN) dari set perubahan.

Panjang batasan: Panjang minimum 1.

Pola: ARN: [-A-ZA-Z0-9:/] *

Jenis: string

Diperlukan: Benar

stackId

ID unik dari tumpukan.

Jenis: string

Diperlukan: Benar

CreateCloudFormationChangeSetInput

Buat permintaan set perubahan aplikasi.

stackName

Properti ini sesuai dengan parameter dengan nama yang sama untuk AWS CloudFormation [CreateChangeSet](#) API.

Jenis: string

Diperlukan: Benar

semanticVersion

Versi semantik aplikasi:

<https://semver.org/>

Jenis: string

Diperlukan: Salah

templateId

UUID kembali. CreateCloudFormationTemplate

Pola: [0-9A-FA-F] {8}\ - [0-9A-FA-F] {4}\ - [0-9A-FA-F] {4}\ - [0-9a-fa-F] {4}\ - [0-9A-FA-F] {12}

Jenis: string

Diperlukan: Salah

parameterOverrides

Daftar nilai parameter untuk parameter aplikasi.

Jenis: Array tipe [ParameterValue](#)

Diperlukan: Salah

capabilities

Daftar nilai yang harus Anda tentukan sebelum Anda dapat menyebarkan aplikasi tertentu. Beberapa aplikasi mungkin menyertakan sumber daya yang dapat memengaruhi izin di AWS akun Anda, misalnya, dengan membuat pengguna new AWS Identity and Access Management (IAM). Untuk aplikasi tersebut, Anda harus secara eksplisit mengakui kemampuannya dengan menentukan parameter ini.

Satu-satunya nilai yang valid

adalah `CAPABILITY_IAM`, `CAPABILITY_NAMED_IAM`, `CAPABILITY_RESOURCE_POLICY`, dan `CAPABILITY_AUTO_EXPAND`.

Sumber daya berikut mengharuskan Anda untuk menentukan `CAPABILITY_IAM` atau `CAPABILITY_NAMED_IAM`: [AWS::IAM::Group](#), [AWS::IAM::InstanceProfile](#), [AWS::IAM::Policy](#), dan [AWS::IAM::Role](#). Jika aplikasi berisi sumber daya IAM, Anda dapat menentukan salah satu `CAPABILITY_IAM` atau `CAPABILITY_NAMED_IAM`. Jika aplikasi berisi sumber daya IAM dengan nama khusus, Anda harus menentukan `CAPABILITY_NAMED_IAM`.

Sumber daya berikut mengharuskan Anda untuk menentukan `CAPABILITY_RESOURCE_POLICY`: [AWS::Lambda::Permission](#), [AWS::IAM::policy](#),,, [AWS::ApplicationAutoScaling::ScalingPolicy](#), [AWS::S3::BucketPolicy](#), [AWS::SQS::QueuePolicy](#) dan [AWS::SNS::TopicPolicy](#)

Aplikasi yang berisi satu atau lebih aplikasi bersarang mengharuskan Anda untuk menentukan `CAPABILITY_AUTO_EXPAND`.

Jika template aplikasi Anda berisi salah satu sumber daya di atas, kami sarankan Anda meninjau semua izin yang terkait dengan aplikasi sebelum menerapkan. Jika Anda tidak menentukan parameter ini untuk aplikasi yang membutuhkan kemampuan, panggilan akan gagal.

Jenis: Array tipe string

Diperlukan: Salah

changeSetName

Properti ini sesuai dengan parameter dengan nama yang sama untuk AWS CloudFormation [CreateChangeSetAPI](#).

Jenis: string

Diperlukan: Salah

clientToken

Properti ini sesuai dengan parameter dengan nama yang sama untuk AWS CloudFormation [CreateChangeSetAPI](#).

Jenis: string

Diperlukan: Salah

description

Properti ini sesuai dengan parameter dengan nama yang sama untuk AWS CloudFormation [CreateChangeSetAPI](#).

Jenis: string

Diperlukan: Salah

notificationArns

Properti ini sesuai dengan parameter dengan nama yang sama untuk AWS CloudFormation [CreateChangeSetAPI](#).

Jenis: Array tipe string

Diperlukan: Salah

resourceTypes

Properti ini sesuai dengan parameter dengan nama yang sama untuk AWS CloudFormation [CreateChangeSetAPI](#).

Jenis: Array tipe string

Diperlukan: Salah

rollbackConfiguration

Properti ini sesuai dengan parameter dengan nama yang sama untuk AWS CloudFormation [CreateChangeSetAPI](#).

Jenis: [RollbackConfiguration](#)

Diperlukan: Salah

tags

Properti ini sesuai dengan parameter dengan nama yang sama untuk AWS CloudFormation [CreateChangeSetAPI](#).

Jenis: Array tipe [Tag](#)

Diperlukan: Salah

ForbiddenException

Klien tidak diautentikasi.

message

Klien tidak diautentikasi.

Jenis: string

Diperlukan: Salah

errorCode

403

Jenis: string

Diperlukan: Salah

InternalServerErrorException

Layanan AWS Serverless Application Repository mengalami kesalahan internal.

message

Layanan AWS Serverless Application Repository mengalami kesalahan internal.

Jenis: string

Diperlukan: Salah

errorCode

500

Jenis: string

Diperlukan: Salah

ParameterValue

Nilai parameter aplikasi.

name

Kunci yang terkait dengan parameter. Jika Anda tidak menetapkan kunci dan nilai untuk parameter tertentu, AWS CloudFormation menggunakan nilai default yang ditentukan dalam templat Anda.

Jenis: string

Diperlukan: Benar

value

Nilai input yang terkait dengan parameter.

Jenis: string

Diperlukan: Benar

RollbackConfiguration

Properti ini sesuai dengan Tipe AWS CloudFormation [RollbackConfiguration](#)Data.

rollbackTriggers

Properti ini sesuai dengan konten dengan nama yang sama untuk Tipe AWS CloudFormation [RollbackConfigurationData](#).

Jenis: Array tipe [RollbackTrigger](#)

Diperlukan: Salah

monitoringTimeInMinutes

Properti ini sesuai dengan konten dengan nama yang sama untuk Tipe AWS CloudFormation [RollbackConfigurationData](#).

Jenis: bilangan bulat

Diperlukan: Salah

RollbackTrigger

Properti ini sesuai dengan Tipe AWS CloudFormation [RollbackTriggerData](#).

arn

Properti ini sesuai dengan konten dengan nama yang sama untuk Tipe AWS CloudFormation [RollbackTriggerData](#).

Jenis: string

Diperlukan: Benar

type

Properti ini sesuai dengan konten dengan nama yang sama untuk Tipe AWS CloudFormation [RollbackTriggerData](#).

Jenis: string

Diperlukan: Benar

Tag

Properti ini sesuai dengan Tipe Data AWS CloudFormation [Tag](#).

key

Properti ini sesuai dengan konten dengan nama yang sama untuk Tipe Data AWS CloudFormation [Tag](#).

Jenis: string

Diperlukan: Benar

value

Properti ini sesuai dengan konten dengan nama yang sama untuk Tipe Data AWS CloudFormation [Tag](#).

Jenis: string

Diperlukan: Benar

TooManyRequestsException

Klien mengirim lebih dari jumlah permintaan yang diizinkan per unit waktu.

message

Klien mengirim lebih dari jumlah permintaan yang diizinkan per unit waktu.

Jenis: string

Diperlukan: Salah

errorCode

429

Jenis: string

Diperlukan: Salah

Lihat juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK dan referensi khusus bahasa, lihat berikut ini:

CreateCloudFormationChangeSet

- [Antarmuka Baris Perintah AWS](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK untuk V3 JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK untuk Python](#)
- [AWS SDK for Ruby V3](#)

Applications applicationId Dependencies

URI

/applications/*applicationId*/dependencies

Metode HTTP

GET

ID Operasi: ListApplicationDependencies

Mengambil daftar aplikasi bersarang di aplikasi yang berisi.

Parameter jalur

Nama	Tipe	Diperlukan	Deskripsi
<i>applicationId</i>	String	True	Amazon Resource Name (ARN) aplikasi.

Parameter kueri

Nama	Tipe	Diperlukan	Deskripsi
nextToken	String	False	Token untuk menentukan di mana harus memulai paginating.
MaxItems	String	False	Jumlah total item yang akan dikembalikan.
SemanticVersion	String	False	Versi semantik aplikasi untuk mendapatkan.

Respons

Kode status	Model respons	Deskripsi
200	<u>ApplicationDependencyPage</u>	Berhasil
400	<u>BadRequestException</u>	Salah satu parameter dalam permintaan tidak valid.
403	<u>ForbiddenException</u>	Klien tidak diautentikasi.
404	<u>NotFoundException</u>	Sumber daya (misalnya, pernyataan kebijakan akses) yang ditentukan dalam permintaan tidak ada.
429	<u>TooManyRequestsException</u>	Klien mengirim lebih dari jumlah permintaan yang diizinkan per unit waktu.

Kode status	Model respons	Deskripsi
500	InternalServerErrorException	Layanan AWS Serverless Application Repository mengalami kesalahan internal.

OPTIONS

Parameter jalur

Nama	Tipe	Diperlukan	Deskripsi
<i>applicationId</i>	String	True	Amazon Resource Name (ARN) aplikasi.

Respons

Kode status	Model respons	Deskripsi
200	None	200 respon

Skema

Badan respons

ApplicationDependencyPageskema

```
{
  "dependencies": [
    {
      "applicationId": "string",
      "semanticVersion": "string"
    }
  ],
  "nextToken": "string"
}
```

BadRequestExceptionkema

```
{  
  "message": "string",  
  "errorCode": "string"  
}
```

ForbiddenExceptionkema

```
{  
  "message": "string",  
  "errorCode": "string"  
}
```

NotFoundExceptionkema

```
{  
  "message": "string",  
  "errorCode": "string"  
}
```

TooManyRequestsExceptionkema

```
{  
  "message": "string",  
  "errorCode": "string"  
}
```

InternalServerErrorExceptionkema

```
{  
  "message": "string",  
  "errorCode": "string"  
}
```

Properti

ApplicationDependencyPage

Daftar ringkasan aplikasi yang bersarang di aplikasi.

dependencies

Sebuah array ringkasan aplikasi bersarang dalam aplikasi.

Jenis: Array tipe [ApplicationDependencySummary](#)

Diperlukan: Benar

nextToken

Token untuk meminta halaman hasil berikutnya.

Jenis: string

Diperlukan: Salah

ApplicationDependencySummary

Ringkasan aplikasi bersarang.

applicationId

Nama Sumber Daya Amazon (ARN) dari aplikasi bersarang.

Jenis: string

Diperlukan: Benar

semanticVersion

Versi semantik dari aplikasi bersarang.

Jenis: string

Diperlukan: Benar

BadRequestException

Salah satu parameter dalam permintaan tidak valid.

message

Salah satu parameter dalam permintaan tidak valid.

Jenis: string

Diperlukan: Salah

errorCode

400

Jenis: string

Diperlukan: Salah

ForbiddenException

Klien tidak diautentikasi.

message

Klien tidak diautentikasi.

Jenis: string

Diperlukan: Salah

errorCode

403

Jenis: string

Diperlukan: Salah

InternalServerErrorException

Layanan AWS Serverless Application Repository mengalami kesalahan internal.

message

Layanan AWS Serverless Application Repository mengalami kesalahan internal.

Jenis: string

Diperlukan: Salah

errorCode

500

Jenis: string

Diperlukan: Salah

NotFoundException

Sumber daya (misalnya, pernyataan kebijakan akses) yang ditentukan dalam permintaan tidak ada.

message

Sumber daya (misalnya, pernyataan kebijakan akses) yang ditentukan dalam permintaan tidak ada.

Jenis: string

Diperlukan: Salah

errorCode

404

Jenis: string

Diperlukan: Salah

TooManyRequestsException

Klien mengirim lebih dari jumlah permintaan yang diizinkan per unit waktu.

message

Klien mengirim lebih dari jumlah permintaan yang diizinkan per unit waktu.

Jenis: string

Diperlukan: Salah

errorCode

429

Jenis: string

Diperlukan: Salah

Lihat juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK dan referensi khusus bahasa, lihat berikut ini:

ListApplicationDependencies

- [Antarmuka Baris Perintah AWS](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK untuk V3 JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK untuk Python](#)
- [AWS SDK for Ruby V3](#)

Kebijakan ApplicationID Aplikasi

URI

/applications/*applicationId*/policy

Metode HTTP

GET

ID Operasi: `GetApplicationPolicy`

Mengambil kebijakan untuk aplikasi.

Parameter jalur

Nama	Tipe	Diperlukan	Deskripsi
<i><code>applicationId</code></i>	String	True	Amazon Resource Name (ARN) aplikasi.

Respons

Kode status	Model respons	Deskripsi
200	ApplicationPolicy	Berhasil
400	BadRequestException	Salah satu parameter dalam permintaan tidak valid.
403	ForbiddenException	Klien tidak diautentikasi.
404	NotFoundException	Sumber daya (misalnya, pernyataan kebijakan akses) yang ditentukan dalam permintaan tidak ada.
429	TooManyRequestsException	Klien mengirim lebih dari jumlah permintaan yang diizinkan per unit waktu.
500	InternalServerErrorException	Layanan AWS Serverless Application Repository mengalami kesalahan internal.

PUT

ID Operasi: PutApplicationPolicy

Menetapkan kebijakan izin untuk aplikasi. Untuk daftar tindakan yang didukung untuk operasi ini, lihat [Izin Aplikasi](#).

Parameter jalur

Nama	Tipe	Diperlukan	Deskripsi
<i>applicationId</i>	String	True	Amazon Resource Name (ARN) aplikasi.

Respons

Kode status	Model respons	Deskripsi
200	ApplicationPolicy	Berhasil
400	BadRequestException	Salah satu parameter dalam permintaan tidak valid.
403	ForbiddenException	Klien tidak diautentikasi.
404	NotFoundException	Sumber daya (misalnya, pernyataan kebijakan akses) yang ditentukan dalam permintaan tidak ada.
429	TooManyRequestsException	Klien mengirim lebih dari jumlah permintaan yang diizinkan per unit waktu.
500	InternalServerErrorException	Layanan AWS Serverless Application Repository mengalami kesalahan internal.

OPTIONS

Parameter jalur

Nama	Tipe	Diperlukan	Deskripsi
<i>applicationId</i>	String	True	Amazon Resource Name (ARN) aplikasi.

Respons

Kode status	Model respons	Deskripsi
200	None	200 respon

Skema

Permintaan badan

PUTskema

```
{
  "statements": [
    {
      "statementId": "string",
      "principals": [
        "string"
      ],
      "actions": [
        "string"
      ],
      "principalOrgIDs": [
        "string"
      ]
    }
  ]
}
```

Badan respons

ApplicationPolicyskema

```
{
  "statements": [
    {
      "statementId": "string",
      "principals": [
        "string"
      ],
      "actions": [
        "string"
      ],
      "principalOrgIDs": [
        "string"
      ]
    }
  ]
}
```

BadRequestExceptionskema

```
{
  "message": "string",
  "errorCode": "string"
}
```

ForbiddenExceptionskema

```
{
  "message": "string",
  "errorCode": "string"
}
```

NotFoundExceptionskema

```
{
  "message": "string",
  "errorCode": "string"
}
```

TooManyRequestsExceptionsKema

```
{  
  "message": "string",  
  "errorCode": "string"  
}
```

InternalServerErrorExceptionsKema

```
{  
  "message": "string",  
  "errorCode": "string"  
}
```

Properti

ApplicationPolicy

Pernyataan kebijakan diterapkan pada aplikasi.

statements

Array pernyataan kebijakan yang diterapkan pada aplikasi.

Jenis: Array tipe [ApplicationPolicyStatement](#)

Diperlukan: Benar

ApplicationPolicyStatement

Pernyataan kebijakan diterapkan pada aplikasi.

statementId

ID unik untuk pernyataan tersebut.

Jenis: string

Diperlukan: Salah

principals

Array ID AWS akun untuk berbagi aplikasi dengan, atau* untuk membuat aplikasi publik.

Jenis: Array tipe string

Diperlukan: Benar

actions

Untuk daftar tindakan yang didukung untuk operasi ini, lihat [Izin Aplikasi](#).

Jenis: Array tipe string

Diperlukan: Benar

principalOrgIDs

AWS OrganizationsID untuk berbagi aplikasi dengan.

Jenis: Array tipe string

Diperlukan: Salah

BadRequestException

Salah satu parameter dalam permintaan tidak valid.

message

Salah satu parameter dalam permintaan tidak valid.

Jenis: string

Diperlukan: Salah

errorCode

400

Jenis: string

Diperlukan: Salah

ForbiddenException

Klien tidak diautentikasi.

message

Klien tidak diautentikasi.

Jenis: string

Diperlukan: Salah

errorCode

403

Jenis: string

Diperlukan: Salah

InternalServerErrorException

Layanan AWS Serverless Application Repository mengalami kesalahan internal.

message

Layanan AWS Serverless Application Repository mengalami kesalahan internal.

Jenis: string

Diperlukan: Salah

errorCode

500

Jenis: string

Diperlukan: Salah

NotFoundException

Sumber daya (misalnya, pernyataan kebijakan akses) yang ditentukan dalam permintaan tidak ada.

message

Sumber daya (misalnya, pernyataan kebijakan akses) yang ditentukan dalam permintaan tidak ada.

Jenis: string

Diperlukan: Salah

errorCode

404

Jenis: string

Diperlukan: Salah

TooManyRequestsException

Klien mengirim lebih dari jumlah permintaan yang diizinkan per unit waktu.

message

Klien mengirim lebih dari jumlah permintaan yang diizinkan per unit waktu.

Jenis: string

Diperlukan: Salah

errorCode

429

Jenis: string

Diperlukan: Salah

Lihat juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK dan referensi khusus bahasa, lihat berikut ini:

GetApplicationPolicy

- [Antarmuka Baris Perintah AWS](#)
- [AWS SDK for .NET](#)

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK untuk V3 JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK untuk Python](#)
- [AWS SDK for Ruby V3](#)

PutApplicationPolicy

- [Antarmuka Baris Perintah AWS](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK untuk V3 JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK untuk Python](#)
- [AWS SDK for Ruby V3](#)

Applications applicationId Templates

URI

/applications/*applicationId*/templates

Metode HTTP

POST

ID Operasi: CreateCloudFormationTemplate

Membuat AWS CloudFormation template.

Parameter jalur

Nama	Tipe	Diperlukan	Deskripsi
<i>applicationId</i>	String	True	Amazon Resource Name (ARN) aplikasi.

Respons

Kode status	Model respons	Deskripsi
201	TemplateDetails	Berhasil
400	BadRequestException	Salah satu parameter dalam permintaan tidak valid.
403	ForbiddenException	Klien tidak diautentikasi.
404	NotFoundException	Sumber daya (misalnya, pernyataan kebijakan akses) yang ditentukan dalam permintaan tidak ada.
429	TooManyRequestsException	Klien mengirim lebih dari jumlah permintaan yang diizinkan per unit waktu.
500	InternalServerErrorException	Layanan AWS Serverless Application Repository mengalami kesalahan internal.

OPTIONS

Parameter jalur

Nama	Tipe	Diperlukan	Deskripsi
<i>applicationId</i>	String	True	Amazon Resource Name (ARN) aplikasi.

Respons

Kode status	Model respons	Deskripsi
200	None	200 respon

Skema

Permintaan badan

POSTskema

```
{  
  "semanticVersion": "string"  
}
```

Badan respons

TemplateDetailsskema

```
{  
  "templateId": "string",  
  "templateUrl": "string",  
  "applicationId": "string",  
  "semanticVersion": "string",  
  "status": enum,  
  "creationTime": "string",  
  "expirationTime": "string"  
}
```

BadRequestExceptionskema

```
{  
  "message": "string",  
  "errorCode": "string"  
}
```

ForbiddenExceptionskema

```
{
```

```
"message": "string",  
"errorCode": "string"  
}
```

NotFoundExceptionskema

```
{  
  "message": "string",  
  "errorCode": "string"  
}
```

TooManyRequestsExceptionskema

```
{  
  "message": "string",  
  "errorCode": "string"  
}
```

InternalServerErrorExceptionskema

```
{  
  "message": "string",  
  "errorCode": "string"  
}
```

Properti

BadRequestException

Salah satu parameter dalam permintaan tidak valid.

message

Salah satu parameter dalam permintaan tidak valid.

Jenis: string

Diperlukan: Salah

errorCode

400

Jenis: string

Diperlukan: Salah

CreateCloudFormationTemplateInput

Buat permintaan template.

semanticVersion

Versi semantik aplikasi:

<https://semver.org/>

Jenis: string

Diperlukan: Salah

ForbiddenException

Klien tidak diautentikasi.

message

Klien tidak diautentikasi.

Jenis: string

Diperlukan: Salah

errorCode

403

Jenis: string

Diperlukan: Salah

InternalServerErrorException

Layanan AWS Serverless Application Repository mengalami kesalahan internal.

message

Layanan AWS Serverless Application Repository mengalami kesalahan internal.

Jenis: string

Diperlukan: Salah

errorCode

500

Jenis: string

Diperlukan: Salah

NotFoundException

Sumber daya (misalnya, pernyataan kebijakan akses) yang ditentukan dalam permintaan tidak ada.

message

Sumber daya (misalnya, pernyataan kebijakan akses) yang ditentukan dalam permintaan tidak ada.

Jenis: string

Diperlukan: Salah

errorCode

404

Jenis: string

Diperlukan: Salah

TemplateDetails

Detail template.

templateId

UUID kembali. CreateCloudFormationTemplate

Pola: [0-9A-FA-F] {8}\ - [0-9A-FA-F] {4}\ - [0-9A-FA-F] {4}\ - [0-9a-fa-F] {4}\ - [0-9A-FA-F] {12}

Jenis: string

Diperlukan: Benar

templateUrl

Tautan ke template yang dapat digunakan untuk menyebarkan aplikasi menggunakan AWS CloudFormation.

Jenis: string

Diperlukan: Benar

applicationId

Aplikasi Amazon Resource Name (ARN).

Jenis: string

Diperlukan: Benar

semanticVersion

Versi semantik aplikasi:

<https://semver.org/>

Jenis: string

Diperlukan: Benar

status

Status alur kerja pembuatan template.

Nilai yang mungkin: PREPARING | ACTIVE | EXPIRED

Jenis: string

Diperlukan: Benar

Nilai: PREPARING | ACTIVE | EXPIRED

creationTime

Tanggal dan waktu sumber daya ini dibuat.

Jenis: string

Diperlukan: Benar

expirationTime

Tanggal dan waktu template ini kedaluwarsa. Template kedaluwarsa 1 jam setelah pembuatan.

Jenis: string

Diperlukan: Benar

TooManyRequestsException

Klien mengirim lebih dari jumlah permintaan yang diizinkan per unit waktu.

message

Klien mengirim lebih dari jumlah permintaan yang diizinkan per unit waktu.

Jenis: string

Diperlukan: Salah

errorCode

429

Jenis: string

Diperlukan: Salah

Lihat juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK dan referensi khusus bahasa, lihat berikut ini:

CreateCloudFormationTemplate

- [Antarmuka Baris Perintah AWS](#)

- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK untuk V3 JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK untuk Python](#)
- [AWS SDK for Ruby V3](#)

Applications applicationId Templates templateId

URI

/applications/*applicationId*/templates/*templateId*

Metode HTTP

GET

ID Operasi: GetCloudFormationTemplate

Mendapat AWS CloudFormation template yang ditentukan.

Parameter jalur

Nama	Tipe	Diperlukan	Deskripsi
<i>applicationId</i>	String	True	Amazon Resource Name (ARN) aplikasi.
<i>TemplateID</i>	String	True	UUID kembali. CreateCloudFormationTemplate Pola: [0-9A-FA-F]{8}\ - [0-9A-FA-F]{4}\ - [0-9A-FA-F]{4}\

Nama	Tipe	Diperlukan	Deskripsi
			- [0-9a-fa-F] {4}\ - [0-9A-FA-F] {12}
Respons			
Kode status	Model respons		Deskripsi
200	TemplateDetails		Berhasil
400	BadRequestException		Salah satu parameter dalam permintaan tidak valid.
403	ForbiddenException		Klien tidak diautentikasi.
404	NotFoundException		Sumber daya (misalnya, pernyataan kebijakan akses) yang ditentukan dalam permintaan tidak ada.
429	TooManyRequestsException		Klien mengirim lebih dari jumlah permintaan yang diizinkan per unit waktu.
500	InternalServerErrorException		Layanan AWS Serverless Application Repository mengalami kesalahan internal.

OPTIONS

Parameter jalur

Nama	Tipe	Diperlukan	Deskripsi
<i>applicationId</i>	String	True	Amazon Resource Name (ARN) aplikasi.

Nama	Tipe	Diperlukan	Deskripsi
<i>TemplateID</i>	String	True	UUID kembali. CreateCloudFormati onTemplate Pola: [0-9A-FA-F] {8}\ - [0-9A-FA-F] {4}\ - [0-9A-FA-F] {4}\ - [0-9a-fa-F] {4}\ - [0-9A-FA-F] {12}

Respons

Kode status	Model respons	Deskripsi
200	None	200 respon

Skema

Badan respons

TemplateDetailsskema

```
{
  "templateId": "string",
  "templateUrl": "string",
  "applicationId": "string",
  "semanticVersion": "string",
  "status": enum,
  "creationTime": "string",
  "expirationTime": "string"
}
```

BadRequestExceptionskema

```
{
  "message": "string",
}
```

```
"errorCode": "string"  
}
```

ForbiddenExceptionsKema

```
{  
  "message": "string",  
  "errorCode": "string"  
}
```

NotFoundExceptionsKema

```
{  
  "message": "string",  
  "errorCode": "string"  
}
```

TooManyRequestsExceptionsKema

```
{  
  "message": "string",  
  "errorCode": "string"  
}
```

InternalServerErrorExceptionsKema

```
{  
  "message": "string",  
  "errorCode": "string"  
}
```

Properti

BadRequestException

Salah satu parameter dalam permintaan tidak valid.

message

Salah satu parameter dalam permintaan tidak valid.

Jenis: string

Diperlukan: Salah

errorCode

400

Jenis: string

Diperlukan: Salah

ForbiddenException

Klien tidak diautentikasi.

message

Klien tidak diautentikasi.

Jenis: string

Diperlukan: Salah

errorCode

403

Jenis: string

Diperlukan: Salah

InternalServerErrorException

Layanan AWS Serverless Application Repository mengalami kesalahan internal.

message

Layanan AWS Serverless Application Repository mengalami kesalahan internal.

Jenis: string

Diperlukan: Salah

errorCode

500

Jenis: string

Diperlukan: Salah

NotFoundException

Sumber daya (misalnya, pernyataan kebijakan akses) yang ditentukan dalam permintaan tidak ada.

message

Sumber daya (misalnya, pernyataan kebijakan akses) yang ditentukan dalam permintaan tidak ada.

Jenis: string

Diperlukan: Salah

errorCode

404

Jenis: string

Diperlukan: Salah

TemplateDetails

Detail template.

templateId

UUID kembali. CreateCloudFormationTemplate

Pola: [0-9A-FA-F] {8}\ - [0-9A-FA-F] {4}\ - [0-9A-FA-F] {4}\ - [0-9a-fa-F] {4}\ - [0-9A-FA-F] {12}

Jenis: string

Diperlukan: Benar

templateUrl

Tautan ke template yang dapat digunakan untuk menyebarkan aplikasi menggunakan AWS CloudFormation.

Jenis: string

Diperlukan: Benar

applicationId

Aplikasi Amazon Resource Name (ARN).

Jenis: string

Diperlukan: Benar

semanticVersion

Versi semantik aplikasi:

<https://semver.org/>

Jenis: string

Diperlukan: Benar

status

Status alur kerja pembuatan template.

Nilai yang mungkin: PREPARING | ACTIVE | EXPIRED

Jenis: string

Diperlukan: Benar

Nilai: PREPARING | ACTIVE | EXPIRED

creationTime

Tanggal dan waktu sumber daya ini dibuat.

Jenis: string

Diperlukan: Benar

expirationTime

Tanggal dan waktu template ini kedaluwarsa. Template kedaluwarsa 1 jam setelah pembuatan.

Jenis: string

Diperlukan: Benar

TooManyRequestsException

Klien mengirim lebih dari jumlah permintaan yang diizinkan per unit waktu.

message

Klien mengirim lebih dari jumlah permintaan yang diizinkan per unit waktu.

Jenis: string

Diperlukan: Salah

errorCode

429

Jenis: string

Diperlukan: Salah

Lihat juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK dan referensi khusus bahasa, lihat berikut ini:

GetCloudFormationTemplate

- [Antarmuka Baris Perintah AWS](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)

- [AWS SDK untuk V3 JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK untuk Python](#)
- [AWS SDK for Ruby V3](#)

Applications applicationId Unshare

URI

/applications/*applicationId*/unshare

Metode HTTP

POST

ID Operasi: UnshareApplication

Membatalkan aplikasi dari AWS Organisasi.

Operasi ini hanya dapat dipanggil dari akun manajemen organisasi.

Parameter jalur

Nama	Tipe	Diperlukan	Deskripsi
<i>applicationId</i>	String	True	Amazon Resource Name (ARN) aplikasi.

Respons

Kode status	Model respons	Deskripsi
204	None	Berhasil
400	BadRequestException	Salah satu parameter dalam permintaan tidak valid.
403	ForbiddenException	Klien tidak diautentikasi.

Kode status	Model respons	Deskripsi
404	NotFoundException	Sumber daya (misalnya, pernyataan kebijakan akses) yang ditentukan dalam permintaan tidak ada.
429	TooManyRequestsException	Klien mengirim lebih dari jumlah permintaan yang diizinkan per unit waktu.
500	InternalServerErrorException	Layanan AWS Serverless Application Repository mengalami kesalahan internal.

OPTIONS

Parameter jalur

Nama	Tipe	Diperlukan	Deskripsi
<i>applicationId</i>	String	True	Amazon Resource Name (ARN) aplikasi.

Respons

Kode status	Model respons	Deskripsi
200	None	200 respon

Skema

Permintaan badan

POSTskema

```
{
```

```
"organizationId": "string"  
}
```

Badan respons

BadRequestExceptionkema

```
{  
  "message": "string",  
  "errorCode": "string"  
}
```

ForbiddenExceptionkema

```
{  
  "message": "string",  
  "errorCode": "string"  
}
```

NotFoundExceptionkema

```
{  
  "message": "string",  
  "errorCode": "string"  
}
```

TooManyRequestsExceptionkema

```
{  
  "message": "string",  
  "errorCode": "string"  
}
```

InternalServerErrorExceptionkema

```
{  
  "message": "string",
```

```
"errorCode": "string"  
}
```

Properti

BadRequestException

Salah satu parameter dalam permintaan tidak valid.

message

Salah satu parameter dalam permintaan tidak valid.

Jenis: string

Diperlukan: Salah

errorCode

400

Jenis: string

Diperlukan: Salah

ForbiddenException

Klien tidak diautentikasi.

message

Klien tidak diautentikasi.

Jenis: string

Diperlukan: Salah

errorCode

403

Jenis: string

Diperlukan: Salah

InternalServerErrorException

Layanan AWS Serverless Application Repository mengalami kesalahan internal.

message

Layanan AWS Serverless Application Repository mengalami kesalahan internal.

Jenis: string

Diperlukan: Salah

errorCode

500

Jenis: string

Diperlukan: Salah

NotFoundException

Sumber daya (misalnya, pernyataan kebijakan akses) yang ditentukan dalam permintaan tidak ada.

message

Sumber daya (misalnya, pernyataan kebijakan akses) yang ditentukan dalam permintaan tidak ada.

Jenis: string

Diperlukan: Salah

errorCode

404

Jenis: string

Diperlukan: Salah

TooManyRequestsException

Klien mengirim lebih dari jumlah permintaan yang diizinkan per unit waktu.

message

Klien mengirim lebih dari jumlah permintaan yang diizinkan per unit waktu.

Jenis: string

Diperlukan: Salah

errorCode

429

Jenis: string

Diperlukan: Salah

UnshareApplicationInput

Unshare permintaan aplikasi.

organizationId

AWS OrganizationsID untuk membatalkan berbagi aplikasi dari.

Jenis: string

Diperlukan: Benar

Lihat juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK dan referensi khusus bahasa, lihat berikut ini:

UnshareApplication

- [Antarmuka Baris Perintah AWS](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK untuk V3 JavaScript](#)

- [AWS SDK for PHP V3](#)
- [AWS SDK untuk Python](#)
- [AWS SDK for Ruby V3](#)

Versi ApplicationId Aplikasi

URI

/applications/*applicationId*/versions

Metode HTTP

GET

ID Operasi: ListApplicationVersions

Daftar versi untuk aplikasi yang ditentukan.

Parameter jalur

Nama	Tipe	Diperlukan	Deskripsi
<i>applicationId</i>	String	True	Amazon Resource Name (ARN) aplikasi.

Parameter kueri

Nama	Tipe	Diperlukan	Deskripsi
MaxItems	String	False	Jumlah total item yang akan dikembalikan.
nextToken	String	False	Token untuk menentukan di mana harus memulai paginating.

Respons

Kode status	Model respons	Deskripsi
200	ApplicationVersionPage	Berhasil
400	BadRequestException	Salah satu parameter dalam permintaan tidak valid.
403	ForbiddenException	Klien tidak diautentikasi.
404	NotFoundException	Sumber daya (misalnya, pernyataan kebijakan akses) yang ditentukan dalam permintaan tidak ada.
429	TooManyRequestsException	Klien mengirim lebih dari jumlah permintaan yang diizinkan per unit waktu.
500	InternalServerErrorException	Layanan AWS Serverless Application Repository mengalami kesalahan internal.

OPTIONS

Parameter jalur

Nama	Tipe	Diperlukan	Deskripsi
<i>applicationId</i>	String	True	Amazon Resource Name (ARN) aplikasi.

Respons

Kode status	Model respons	Deskripsi
200	None	200 respon

Skema

Badan respons

ApplicationVersionPageskema

```
{
  "versions": [
    {
      "applicationId": "string",
      "semanticVersion": "string",
      "sourceCodeUrl": "string",
      "creationTime": "string"
    }
  ],
  "nextToken": "string"
}
```

BadRequestExceptionskema

```
{
  "message": "string",
  "errorCode": "string"
}
```

ForbiddenExceptionskema

```
{
  "message": "string",
  "errorCode": "string"
}
```

NotFoundExceptionskema

```
{
  "message": "string",
  "errorCode": "string"
}
```

TooManyRequestsExceptionkema

```
{
  "message": "string",
  "errorCode": "string"
}
```

InternalServerErrorExceptionkema

```
{
  "message": "string",
  "errorCode": "string"
}
```

Properti

ApplicationVersionPage

Daftar ringkasan versi untuk aplikasi.

versions

Sebuah array ringkasan versi untuk aplikasi.

Jenis: Array tipe [VersionSummary](#)

Diperlukan: Benar

nextToken

Token untuk meminta halaman hasil berikutnya.

Jenis: string

Diperlukan: Salah

BadRequestException

Salah satu parameter dalam permintaan tidak valid.

message

Salah satu parameter dalam permintaan tidak valid.

Jenis: string

Diperlukan: Salah

errorCode

400

Jenis: string

Diperlukan: Salah

ForbiddenException

Klien tidak diautentikasi.

message

Klien tidak diautentikasi.

Jenis: string

Diperlukan: Salah

errorCode

403

Jenis: string

Diperlukan: Salah

InternalServerErrorException

Layanan AWS Serverless Application Repository mengalami kesalahan internal.

message

Layanan AWS Serverless Application Repository mengalami kesalahan internal.

Jenis: string

Diperlukan: Salah

errorCode

500

Jenis: string

Diperlukan: Salah

NotFoundException

Sumber daya (misalnya, pernyataan kebijakan akses) yang ditentukan dalam permintaan tidak ada.

message

Sumber daya (misalnya, pernyataan kebijakan akses) yang ditentukan dalam permintaan tidak ada.

Jenis: string

Diperlukan: Salah

errorCode

404

Jenis: string

Diperlukan: Salah

TooManyRequestsException

Klien mengirim lebih dari jumlah permintaan yang diizinkan per unit waktu.

message

Klien mengirim lebih dari jumlah permintaan yang diizinkan per unit waktu.

Jenis: string

Diperlukan: Salah

errorCode

429

Jenis: string

Diperlukan: Salah

VersionSummary

Ringkasan versi aplikasi.

applicationId

Aplikasi Amazon Resource Name (ARN).

Jenis: string

Diperlukan: Benar

semanticVersion

Versi semantik aplikasi:

<https://semver.org/>

Jenis: string

Diperlukan: Benar

sourceCodeUrl

Tautan ke repositori publik untuk kode sumber aplikasi Anda, misalnya URL dari komit tertentu GitHub .

Jenis: string

Diperlukan: Salah

creationTime

Tanggal dan waktu sumber daya ini dibuat.

Jenis: string

Diperlukan: Benar

Lihat juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK dan referensi khusus bahasa, lihat berikut ini:

ListApplicationVersions

- [Antarmuka Baris Perintah AWS](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK untuk V3 JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK untuk Python](#)
- [AWS SDK for Ruby V3](#)

Aplikasi ApplicationID Versi SemanticVersion

URI

/applications/*applicationId*/versions/*semanticVersion*

Metode HTTP

PUT

ID Operasi: CreateApplicationVersion

Membuat versi aplikasi.

Parameter jalur

Nama	Tipe	Diperlukan	Deskripsi
<i>applicationId</i>	String	True	Amazon Resource Name (ARN) aplikasi.
<i>SemanticVersion</i>	String	True	Versi semantik dari versi baru.

Respons

Kode status	Model respons	Deskripsi
201	Version	Berhasil
400	BadRequestException	Salah satu parameter dalam permintaan tidak valid.
403	ForbiddenException	Klien tidak diautentikasi.
409	ConflictException	Sumber daya sudah ada.
429	TooManyRequestsException	Klien mengirim lebih dari jumlah permintaan yang diizinkan per unit waktu.
500	InternalServerErrorException	Layanan AWS Serverless Application Repository mengalami kesalahan internal.

OPTIONS

Parameter jalur

Nama	Tipe	Diperlukan	Deskripsi
<i>applicationId</i>	String	True	Amazon Resource Name (ARN) aplikasi.

Nama	Tipe	Diperlukan	Deskripsi
<i>SemanticVersion</i>	String	True	Versi semantik dari versi baru.

Respons

Kode status	Model respons	Deskripsi
200	None	200 respon

Skema

Permintaan badan

PUTskema

```
{
  "templateBody": "string",
  "templateUrl": "string",
  "sourceCodeUrl": "string",
  "sourceCodeArchiveUrl": "string"
}
```

Badan respons

Versionskema

```
{
  "applicationId": "string",
  "semanticVersion": "string",
  "sourceCodeUrl": "string",
  "sourceCodeArchiveUrl": "string",
  "templateUrl": "string",
  "creationTime": "string",
  "parameterDefinitions": [
    {
      "name": "string",
      "defaultValue": "string",
      "description": "string",
    }
  ]
}
```

```
"type": "string",
"noEcho": boolean,
"allowedPattern": "string",
"constraintDescription": "string",
"minValue": integer,
"maxValue": integer,
"minLength": integer,
"maxLength": integer,
"allowedValues": [
  "string"
],
"referencedByResources": [
  "string"
]
}
],
"requiredCapabilities": [
  enum
],
"resourcesSupported": boolean
}
```

BadRequestExceptionkema

```
{
  "message": "string",
  "errorCode": "string"
}
```

ForbiddenExceptionkema

```
{
  "message": "string",
  "errorCode": "string"
}
```

ConflictExceptionkema

```
{
  "message": "string",
  "errorCode": "string"
}
```

```
}
```

TooManyRequestsExceptionsKema

```
{  
  "message": "string",  
  "errorCode": "string"  
}
```

InternalServerErrorExceptionsKema

```
{  
  "message": "string",  
  "errorCode": "string"  
}
```

Properti

BadRequestException

Salah satu parameter dalam permintaan tidak valid.

message

Salah satu parameter dalam permintaan tidak valid.

Jenis: string

Diperlukan: Salah

errorCode

400

Jenis: string

Diperlukan: Salah

Capability

Nilai yang harus ditentukan untuk menyebarkan beberapa aplikasi.

CAPABILITY_IAM
CAPABILITY_NAMED_IAM
CAPABILITY_AUTO_EXPAND
CAPABILITY_RESOURCE_POLICY

ConflictException

Sumber daya sudah ada.

message

Sumber daya sudah ada.

Jenis: string
Diperlukan: Salah

errorCode

409

Jenis: string
Diperlukan: Salah

CreateApplicationVersionInput

Buat permintaan versi.

templateBody

AWS SAMTemplate paket mentah dari aplikasi Anda.

Jenis: string
Diperlukan: Salah

templateUrl

Sebuah link ke AWS SAM template paket aplikasi Anda.

Jenis: string
Diperlukan: Salah

sourceCodeUrl

Tautan ke repositori publik untuk kode sumber aplikasi Anda, misalnya URL dari komit tertentu GitHub .

Jenis: string

Diperlukan: Salah

sourceCodeArchiveUrl

Tautan ke objek S3 yang berisi arsip ZIP dari kode sumber untuk versi aplikasi Anda ini.

Ukuran maksimal 50 MB

Jenis: string

Diperlukan: Salah

ForbiddenException

Klien tidak diautentikasi.

message

Klien tidak diautentikasi.

Jenis: string

Diperlukan: Salah

errorCode

403

Jenis: string

Diperlukan: Salah

InternalServerErrorException

Layanan AWS Serverless Application Repository mengalami kesalahan internal.

message

Layanan AWS Serverless Application Repository mengalami kesalahan internal.

Jenis: string

Diperlukan: Salah

errorCode

500

Jenis: string

Diperlukan: Salah

ParameterDefinition

Parameter yang didukung oleh aplikasi.

name

Nama parameter.

Jenis: string

Diperlukan: Benar

defaultValue

Nilai tipe yang sesuai untuk templat untuk digunakan jika tidak ada nilai yang ditentukan ketika tumpukan dibuat. Jika Anda menentukan kendala untuk parameter, Anda harus menentukan nilai yang mematuhi kendala tersebut.

Jenis: string

Diperlukan: Salah

description

Sebuah string hingga 4.000 karakter yang menggambarkan parameter.

Jenis: string

Diperlukan: Salah

type

Jenis parameter.

Nilai yang valid: `String` | `Number` | `List<Number>` | `CommaDelimitedList`

`String`: Sebuah string literal.

Misalnya, pengguna dapat menentukan `"MyUserName"`.

`Number`: Sebuah integer atau float. AWS CloudFormation memvalidasi nilai parameter sebagai angka. Namun, ketika Anda menggunakan parameter di tempat lain di template Anda (misalnya, dengan menggunakan fungsi Ref intrinsik), nilai parameter menjadi string.

Misalnya, pengguna mungkin menentukan `"8888"`.

`List<Number>`: Sebuah array bilangan bulat atau float yang dipisahkan oleh koma. AWS CloudFormation memvalidasi nilai parameter sebagai angka. Namun, ketika Anda menggunakan parameter di tempat lain di template Anda (misalnya, dengan menggunakan fungsi Ref intrinsik), nilai parameter menjadi daftar string.

Misalnya, pengguna mungkin menentukan `"80,20"`, dan kemudian Ref menghasilkan `["80", "20"]`

`CommaDelimitedList`: Sebuah array string literal yang dipisahkan oleh koma. Jumlah total string harus satu lebih dari jumlah koma. Juga, setiap string anggota dipangkas spasi.

Misalnya, pengguna mungkin menentukan `"test, dev, prod"`, dan kemudian Ref menghasilkan `["test", "dev", "prod"]`

Jenis: string

Diperlukan: Salah

noEcho

Apakah akan menutupi nilai parameter setiap kali ada yang membuat panggilan yang menjelaskan tumpukan. Jika Anda menetapkan nilai ke true, nilai parameter ditutupi dengan tanda bintang (`*****`).

Jenis: boolean

Diperlukan: Salah

`allowedPattern`

Suatu ekspresi reguler yang mewakili pola untuk memungkinkan `String` tipe.

Jenis: `string`

Diperlukan: Salah

`constraintDescription`

Suatu string yang menjelaskan kendala ketika ia dilanggar. Sebagai contoh, tanpa deskripsi kendala, parameter yang memiliki pola yang diizinkan `[A-Za-z0-9]+` menampilkan pesan kesalahan berikut ketika pengguna menentukan nilai yang tidak valid:

```
Malformed input-Parameter MyParameter must match pattern [A-Za-z0-9]+
```

Dengan menambahkan deskripsi kendala, seperti “harus berisi hanya huruf besar dan kecil dan angka,” Anda dapat menampilkan pesan kesalahan yang disesuaikan berikut:

```
Malformed input-Parameter MyParameter must contain only uppercase and lowercase letters and numbers.
```

Jenis: `string`

Diperlukan: Salah

`minValue`

Nilai numerik yang menentukan nilai numerik terkecil yang ingin Anda izinkan untuk `Number` tipe.

Jenis: `bilangan bulat`

Diperlukan: Salah

`maxValue`

Nilai numerik yang menentukan nilai numerik terbesar yang ingin Anda izinkan untuk `Number` tipe.

Jenis: `bilangan bulat`

Diperlukan: Salah

minLength

Nilai integer yang menentukan jumlah karakter terkecil yang ingin Anda izinkan untuk `String` tipe.

Jenis: bilangan bulat

Diperlukan: Salah

maxLength

Nilai integer yang menentukan jumlah karakter terbesar yang ingin Anda izinkan untuk `String` tipe.

Jenis: bilangan bulat

Diperlukan: Salah

allowedValues

Array yang berisi daftar nilai yang diizinkan untuk parameter.

Jenis: Array tipe string

Diperlukan: Salah

referencedByResources

Daftar AWS SAM sumber daya yang menggunakan parameter ini.

Jenis: Array tipe string

Diperlukan: Benar

TooManyRequestsException

Klien mengirim lebih dari jumlah permintaan yang diizinkan per unit waktu.

message

Klien mengirim lebih dari jumlah permintaan yang diizinkan per unit waktu.

Jenis: string

Diperlukan: Salah

errorCode

429

Jenis: string

Diperlukan: Salah

Version

Detail versi aplikasi.

applicationId

Aplikasi Amazon Resource Name (ARN).

Jenis: string

Diperlukan: Benar

semanticVersion

Versi semantik aplikasi:

<https://semver.org/>

Jenis: string

Diperlukan: Benar

sourceCodeUrl

Tautan ke repositori publik untuk kode sumber aplikasi Anda, misalnya URL dari komit tertentu GitHub .

Jenis: string

Diperlukan: Salah

sourceCodeArchiveUrl

Tautan ke objek S3 yang berisi arsip ZIP dari kode sumber untuk versi aplikasi Anda ini.

Ukuran maksimal 50 MB

Jenis: string

Diperlukan: Salah

templateUrl

Sebuah link ke AWS SAM template paket aplikasi Anda.

Jenis: string

Diperlukan: Benar

creationTime

Tanggal dan waktu sumber daya ini dibuat.

Jenis: string

Diperlukan: Benar

parameterDefinitions

Array tipe parameter yang didukung oleh aplikasi.

Jenis: Array tipe [ParameterDefinition](#)

Diperlukan: Benar

requiredCapabilities

Daftar nilai yang harus Anda tentukan sebelum Anda dapat menyebarkan aplikasi tertentu. Beberapa aplikasi mungkin menyertakan sumber daya yang dapat memengaruhi izin di AWS akun Anda, misalnya, dengan membuat pengguna new AWS Identity and Access Management (IAM). Untuk aplikasi tersebut, Anda harus secara eksplisit mengakui kemampuannya dengan menentukan parameter ini.

Satu-satunya nilai yang valid

adalah `CAPABILITY_IAM`, `CAPABILITY_NAMED_IAM`, `CAPABILITY_RESOURCE_POLICY`, dan `CAPABILITY_AUTO_EXPAND`.

Sumber daya berikut mengharuskan Anda untuk menentukan `CAPABILITY_IAM`

atau `CAPABILITY_NAMED_IAM`: [AWS::IAM::Group](#), [AWS::IAM::InstanceProfile](#), [AWS::IAM::Policy](#),

dan [AWS::IAM::Role](#). Jika aplikasi berisi sumber daya IAM, Anda dapat menentukan salah satu `CAPABILITY_IAM` atau `CAPABILITY_NAMED_IAM`. Jika aplikasi berisi sumber daya IAM dengan nama khusus, Anda harus menentukan `CAPABILITY_NAMED_IAM`.

Sumber daya berikut mengharuskan Anda untuk menentukan `CAPABILITY_RESOURCE_POLICY`: [AWS::Lambda::Permission](#), [AWS::IAM::Policy](#), [AWS::ApplicationAutoScaling::ScalingPolicy](#), [AWS::S3::BucketPolicy](#) dan [AWS::SQS::QueuePolicy](#) [AWS::SNS::TopicPolicy](#)

Aplikasi yang berisi satu atau lebih aplikasi bersarang mengharuskan Anda untuk menentukan `CAPABILITY_AUTO_EXPAND`.

Jika template aplikasi Anda berisi salah satu sumber daya di atas, kami sarankan Anda meninjau semua izin yang terkait dengan aplikasi sebelum menerapkan. Jika Anda tidak menentukan parameter ini untuk aplikasi yang membutuhkan kemampuan, panggilan akan gagal.

Jenis: Array tipe [Capability](#)

Diperlukan: Benar

`resourcesSupported`

Apakah semua sumber AWS daya yang terkandung dalam aplikasi ini didukung di wilayah di mana ia sedang diambil.

Jenis: boolean

Diperlukan: Benar

Lihat juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK dan referensi khusus bahasa, lihat berikut ini:

CreateApplicationVersion

- [Antarmuka Baris Perintah AWS](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)

- [AWS SDK untuk V3 JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK untuk Python](#)
- [AWS SDK for Ruby V3](#)

Riwayat Dokumen

- Versi API: terbaru
- Pembaruan dokumentasi terbaru: 10 Maret 2020

Tabel berikut menjelaskan perubahan penting dalam setiap perilsan Panduan Developer AWS Serverless Application Repository. Untuk notifikasi tentang pembaruan dokumentasi ini, Anda dapat berlangganan ke umpan RSS.

Perubahan	Deskripsi	Tanggal
Pembaruan untuk berbagi dan membatasi akses ke aplikasi	Menambahkan dukungan untuk berbagi aplikasi ke akun diAWS Organizations, dan membatasi akses ke aplikasi publik untukAWS akun danAWS Organisasi. Untuk contoh selengkapnya tentang berbagi aplikasi kepada pengguna di organisasi, lihat Contoh Kebijakan AWS Serverless Application Repository Aplikasi . Untuk contoh pembatasan akses ke aplikasi publik, lihat Contoh KebijakanAWS Serverless Application Repository Berbasis Identitas .	10 Maret 2020
Sumber daya baru yang didukung	Ditambahkan dukungan untuk sejumlah sumber daya tambahan. Untuk daftar lengkap sumber daya yang didukung, lihat Daftar	17 Januari 2020

Sumber AWS Daya yang Didukung.		
Wilayah China	kini AWS Serverless Application Repository tersedia di wilayah China, Beijing dan Ningxia Untuk informasi selengkapnya tentang AWS Serverless Application Repository wilayah dan titik akhir, lihat Wilayah dan titik akhir dalam versi Referensi Umum AWS.	15 Januari 2020
Bagian keamanan diperbarui untuk konsistensi dengan AWS layanan dengan layanan.	Untuk informasi selengkapnya, lihat Keamanan .	2 Januari 2020
Proses yang disederhanakan untuk penerbitan aplikasi	Perintah <code>publish</code> baru di AWS SAM CLI menyederhanakan proses untuk memublikasikan aplikasi nirserver di AWS Serverless Application Repository. Untuk end-to-end tutorial mengunduh dan menerbitkan contoh aplikasi, lihat Mulai Cepat: Menerbitkan Aplikasi . Untuk petunjuk tentang memublikasikan aplikasi yang telah Anda kembangkan dan uji di AWS Cloud, lihat Menerbitkan Aplikasi melalui AWS SAM CLI .	21 Desember 2018

Aplikasi bersarang dan dukungan Layers	Menambahkan dukungan untuk Aplikasi dan Lapisan Bersarang. Ini termasuk pembaruan untuk AWSSumber Daya yang Didukung dan Mengakui Kemampuan Aplikasi .	29 November 2018
Menerbitkan aplikasi dengan peran IAM kustom dan kebijakan sumber daya	Menambahkan dukungan untuk menerbitkan aplikasi dengan peran IAM khusus dan kebijakan sumber daya. Ini termasuk pembaruan pada alur kerja Aplikasi Konsumsi dan Penerbitan Aplikasi dan pembaruan ke AWSSumber Daya yang Didukung dan Referensi API di PanduanAWS Serverless Application Repository Pengembang.	16 Novbucket 2018
Pembaruan Template Kebijakan	Pembaruan untuk Template Kebijakan yang didukung di PanduanAWS Serverless Application Repository Pengembang.	26 September 2018
Update dokumentasi	Menambahkan topik Otentikasi dan Kontrol Akses ke PanduanAWS Serverless Application Repository Pengembang.	2 Juli 2018

[Rilis publik](#)

Rilis publik AWS Serverless Application Repository, yang sekarang tersedia di 14 AWS Wilayah. Untuk informasi lebih lanjut tentang AWS Wilayah tempat AWS Serverless Application Repository tersedia dan AWS Serverless Application Repository titik akhir, lihat [Wilayah dan Titik Akhir](#) di Referensi Umum AWS.

20 Februari 2018

[Panduan baru](#)

Ini adalah rilis pratinjau pertama dari Panduan AWS Serverless Application Repository Pengembang.

30 November 2017

AWSGlosarium

Untuk AWS terminologi terbaru, lihat [AWSglosarium di Referensi](#). Glosarium AWS

Terjemahan disediakan oleh mesin penerjemah. Jika konten terjemahan yang diberikan bertentangan dengan versi bahasa Inggris aslinya, utamakan versi bahasa Inggris.