Guia do desenvolvedor

AWS Serverless Application Repository

Copyright © 2023 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

AWS Serverless Application Repository: Guia do desenvolvedor

Copyright © 2023 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

As marcas e imagens comerciais da Amazon não podem ser usadas em conexão com nenhum produto ou serviço que não seja da Amazon, de qualquer maneira que possa causar confusão entre os clientes, que menospreze ou desacredite a Amazon. Todas as outras marcas comerciais que não pertencem à Amazon pertencem a seus respectivos proprietários, que podem ou não ser afiliados, conectados ou patrocinados pela Amazon.

Table of Contents

O que é o AWS Serverless Application Repository? ... 1
Próximas etapas ... 1
Quick Start: Publishing Applications ... 3
Visão geral ... 3
Aplicativo Hello World ... 3
Antes de começar ... 4
Etapa 1: Inicializar o aplicativo ... 4
Etapa 2: Testar o aplicativo localmente ... 5
Etapa 3: Empacotar o aplicativo ... 6
Etapa 4: publicar o aplicativo ... 8
Próximas etapas ... 8
Mais informações ... 9
Publicar aplicativos ... 10
Como usar o AWS SAM com a AWS Serverless Application Repository ... 11
CompatívelAWSRecursos noAWS Serverless Application Repository ... 11
Modelos de política ... 12
Lista de compatíveisAWSRecursos ... 12
Como publicar aplicativos ... 19
Publicar um aplicativo (AWS CLI) ... 19
Publicar um novo aplicativo (console) ... 20
Compartilhar um aplicativo ... 25
Cancelar o compartilhamento de um aplicativo ... 28
Excluir um aplicativo ... 30
Publicar novas versões do aplicativo ... 30
Selo de autor verificado ... 32
Solicitar um selo de autor verificado ... 32
Compartilhar camadas do Lambda ... 33
Como funcionam ... 33
Exemplo ... 33
Como implantar aplicativos ... 35
Permissões de implantação do aplicativo ... 35
Recursos do aplicativo ... 36
Encontrar e confirmar recursos do aplicativo (console) ... 37
Visualizar recursos do aplicativo (AWS CLI) ... 37

Como implantar aplicativos ... 38
Implantar um novo aplicativo (console) ... 38
Implantar um novo aplicativo (AWS CLI) ... 39
Exclusão de pilhas do aplicativo ... 41
Atualizar aplicativos ... 41
Segurança ... 43
Proteção de dados ... 44
Criptografia em trânsito ... 45
Criptografia em repouso ... 45
Identity and Access Management (Gerenciamento de identidade e acesso) ... 45
Público ... 46
Autenticação com identidades ... 46
Gerenciamento do acesso usando políticas ... 50
Como aAWS Serverless Application Repository funcionam com o IAM ... 52
Exemplos de políticas baseadas em identidade ... 58
Exemplos de política de usuário ... 67
Referência de permissões da API do AWS Serverless Application Repository ... 73
Solução de problemas ... 76
Registro em log e monitoramento ... 79
Registro em log de chamadas à API do AWS Serverless Application Repository com o AWS CloudTrail ... 79
Validação de conformidade ... 83
Resiliência ... 84
Segurança da infraestrutura ... 84
Cotas ... 86
Solução de problemas ... 87
Não é possível tornar um aplicativo público ... 87
Uma cota foi excedida ... 88
Um arquivo Readme atualizado não aparece imediatamente ... 88
Você não pode implantar um aplicativo devido a permissões do IAM insuficientes ... 88
Não é possível implantar o mesmo aplicativo duas vezes ... 88
Por que meu aplicativo não está publicamente disponível? ... 89
Como entrar em contato com o Support ... 89
Operações ... 90
Recursos ... 92
Applications ... 92

URI ... 92
Métodos HTTP ... 92
Esquemas ... 94
Propriedades ... 98
Consulte também ... 116
Aplicativos ApplicationID ... 117
URI ... 117
Métodos HTTP ... 117
Esquemas ... 121
Propriedades ... 124
Consulte também ... 138
Aplicativos ApplicationID Changesets ... 139
URI ... 139
Métodos HTTP ... 139
Esquemas ... 140
Propriedades ... 142
Consulte também ... 150
Applications applicationId Dependencies ... 151
URI ... 151
Métodos HTTP ... 151
Esquemas ... 153
Propriedades ... 155
Consulte também ... 158
Política de ApplicationID de aplicativos ... 158
URI ... 158
Métodos HTTP ... 159
Esquemas ... 161
Propriedades ... 163
Consulte também ... 167
Applications applicationId Templates ... 168
URI ... 168
Métodos HTTP ... 168
Esquemas ... 169
Propriedades ... 171
Consulte também ... 175
Applications applicationId Templates templateId ... 176

URI ... 176
Métodos HTTP ... 176
Esquemas ... 178
Propriedades ... 179
Consulte também ... 183
Applications applicationId Unshare ... 184
URI ... 184
Métodos HTTP ... 184
Esquemas ... 185
Propriedades ... 187
Consulte também ... 189
Versões do ApplicationID de aplicativos ... 190
URI ... 190
Métodos HTTP ... 190
Esquemas ... 192
Propriedades ... 193
Consulte também ... 197
Aplicativos: ApplicationID Versions SemanticVersion ... 198
URI ... 198
Métodos HTTP ... 198
Esquemas ... 199
Propriedades ... 202
Consulte também ... 211
Histórico do documento ... 212
Glossário da AWS ... 216
... ccxvii

O que é o AWS Serverless Application Repository?

OAWS Serverless Application RepositoryCom o, os desenvolvedores e as empresas podem encontrar, implantar e publicar de forma mais fácil e rápida aplicativos sem servidor noAWSCloud. Para obter mais informações sobre aplicativos sem servidor, consulteComputação e aplicativos sem servidornoAWSWebsite.

É possível publicar aplicativos facilmente, compartilhá-los com a comunidade em geral ou de maneira particular entre membros da sua equipe ou em toda a organização. Para publicar um aplicativo sem servidor (ou aplicativo), é possível usar oAWS Management Console, oAWS SAMInterface da linha de comando do (AWS SAMouAWSSDKs para carregar seu código. Juntamente com seu código, você pode fazer upload de um arquivo manifesto simples, também conhecido como modelo AWS Serverless Application Model (AWS SAM). Para obter mais informações sobre o AWS SAM, consulte o Guia do desenvolvedor do AWS Serverless Application Model.

O AWS Serverless Application Repository está profundamente integrado com o console do AWS Lambda. A integração significa que os desenvolvedores de todos os níveis podem começar a usar a computação sem servidor sem necessidade de aprender coisas novas. Você pode usar palavras-chave de categoria para procurar aplicativos como back-ends da web e de dispositivos móveis, aplicativos de processamento de dados ou chatbots. Você também pode pesquisar aplicativos por nome, editor ou fonte de evento. Para usar um aplicativo, basta selecioná-lo, configurar os campos necessários e implantá-lo com alguns cliques.

Neste guia, mostraremos as duas maneiras de trabalhar com o AWS Serverless Application Repository:

- Publicar aplicativos— Configure e carregue aplicativos para disponibilizá-los a outros desenvolvedores e publique novas versões de aplicativos.
- Como implantar aplicativos— Procure aplicativos e visualize informações sobre eles, incluindo código-fonte e arquivos “readme”. Além disso, instale, configure e implante os aplicativos que você preferir.

Próximas etapas

- Para um tutorial sobre a publicação de um aplicativo de amostra noAWS Serverless Application Repository, consulteQuick Start: Publishing Applications.

- Para obter instruções sobre como implantar aplicativos a partir doAWS Serverless Application Repository, consulteComo implantar aplicativos.

Quick Start: Publishing Applications

Este guia orienta você pelas etapas de download, criação, teste e publicação de um aplicativo de exemplo sem servidor no AWS Serverless Application Repository usando a CLI do AWS SAM. É possível usar esse aplicativo de exemplo como ponto inicial para desenvolver e publicar o seu próprio aplicativo sem servidor.

Visão geral

As etapas a seguir definem como baixar, criar e publicar um exemplo de aplicativo sem servidor:

1. Inicializar. Faça download de um exemplo de aplicativo a partir do modelo usando `sam init`.
2. Testar localmente. Teste o aplicativo localmente usando `sam local invoke` e/ou `sam local start-api`. Observe que, com esses comandos, mesmo que sua função Lambda seja invocada localmente, ela lê e grava emAWS recursos naAWS nuvem.
3. Pacote. Quando estiver satisfeito com sua função do Lambda, agrupe a função, oAWS SAM modelo e quaisquer dependências do Lambda em um pacote deAWS CloudFormation implantação usando`sam package`. Nesta etapa você também incluirá as informações sobre o aplicativo que será carregado no AWS Serverless Application Repository.
4. Publicar. Publique o aplicativo no AWS Serverless Application Repository usando `sam publish`. Ao final desta etapa, você poderá visualizar seu aplicativoAWS Serverless Application Repository e implantá-lo naAWS nuvem usandoAWS Serverless Application Repository.

O exemplo Aplicativo Hello World na seção a seguir guia você por essas etapas na criação e publicação de um aplicativo sem servidor.

Aplicativo Hello World

Neste exercício, baixe e teste um aplicativo Hello World sem servidor que representa um simples back-end da API. Consiste em um endpoint do Amazon API Gateway que suporta uma operação GET e uma função Lambda. Quando uma solicitação GET é enviada ao endpoint, o API Gateway invoca a função do Lambda. Depois, o AWS Lambda executará a função, que basicamente retornará uma mensagem `hello world`.

O aplicativo tem os seguintes componentes:

- UmAWS SAM modelo que define doisAWS recursos para o aplicativo Hello World: um serviço de API Gateway com uma operação GET e uma função Lambda. O modelo também define o mapeamento entre a operação GET do API Gateway e a função Lambda.
- Código do aplicativo gravado em Python.

Antes de começar

Certifique-se de que tenha a configuração necessária para este exercício:

- Você deve ter umaAWS conta com um usuário do IAM que tenha permissões administrativas. Consulte Configurar umaAWS conta.
- É necessário que a CLI (interface da linha de comando) do AWS SAM tenha sido instalada. Consulte Instalar a CLI do AWS SAM.
- É necessário ter a versão 1.16.77 ou posterior da AWS CLI instalada. Consulte Instalar a AWS Command Line Interface.

Etapa 1: Inicializar o aplicativo

Nesta seção, baixe o exemplo de aplicativo, que consiste em um modelo do AWS SAM e no código do aplicativo.

Como inicializar o aplicativo

1. Execute o seguinte comando em um prompt de comando da CLI do AWS SAM.

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

2. Revise o conteúdo do diretório criado pelo comando (`sam-app/`):

   - `template.yaml`— Define doisAWS recursos que o aplicativo Hello World precisa: uma função Lambda e um endpoint do API Gateway que suporte uma operação GET. O modelo também define o mapeamento entre os dois recursos.
   - Conteúdo relacionado ao código do aplicativo Hello World:
     - `hello_world/`diretório — Contém o código do aplicativo, que retorna`hello world` quando você o executa.

> **Note**
>
> Para este exercício, o código do aplicativo está gravado em Python e você especifica o tempo de execução no comando `init`. O AWS Lambda oferece suporte a idiomas adicionais para criar o código do aplicativo. Se você especificar outro tempo de execução compatível, o comando `init` oferecerá o código Hello World no idioma especificado e um arquivo `README.md` que você poderá seguir no idioma correspondente. Para obter informações sobre tempos de execução compatíveis, consulte Ambiente de execução do Lambda e bibliotecas disponíveis.

Etapa 2: Testar o aplicativo localmente

Agora que você tem o aplicativo do AWS SAM na máquina local, siga as etapas abaixo para testá-la localmente.

Como testar o aplicativo localmente

1. Inicie o endpoint do gateway da API localmente. É necessário executar o comando a seguir a partir do diretório que contém o arquivo `template.yaml`.

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

   O comando retorna um endpoint do API Gateway, para o qual você pode enviar solicitações para testes locais.

2. Testar o aplicativo Copie o URL do endpoint do API Gateway, cole-o no navegador e escolha Enter. Um exemplo de URL de endpoint do API Gateway é`http://127.0.0.1:3000/hello`.

   O API Gateway invoca localmente a função Lambda para a qual o endpoint está mapeado. A função Lambda é executada no contêiner local do Docker e retorna`hello world`. O API Gateway retorna uma resposta ao navegador que contém o texto.

Exercício: alterar a string da mensagem.

Depois de testar o exemplo de aplicativo com êxito, é possível testar fazendo uma simples modificação: alterar a string da mensagem retornada.

1. Edite o arquivo `/hello_world/app.py` para alterar a string da mensagem de `'hello world'` para `'Hello World!'`.
2. Recarregue o URL de teste no navegador e observe a nova string.

Você notará que o novo código será carregado de forma dinâmica, sem precisar reiniciar o processo do `sam local`.

Etapa 3: Empacotar o aplicativo

Depois de testar o aplicativo localmente, use a CLI do AWS SAM para criar um pacote de implantação e um modelo do AWS SAM empacotado.

> **Note**
>
> Nas etapas a seguir, crie um arquivo .zip para o conteúdo do diretório `hello_world/`, que contém o código do aplicativo. Este arquivo .zip é o pacote de implantação para o seu aplicativo sem servidor. Para obter mais informações, consulte Criando um Package de implantação (Python) no Guia doAWS Lambda desenvolvedor.

Como Como Como Como Como Como Como Lambda Como Como Como Como

1. Adicione uma seção de `Metadata` ao arquivo do modelo do AWS SAM fornecendo as informações necessárias do aplicativo. Para obter mais informações sobre a seção `Metadata` dos modelos do AWS SAM, consulte Propriedades da seção dos metadados do modelo do AWS SAM no Guia do desenvolvedor do AWS Serverless Application Model.

   Veja um exemplo da seção `Metadata`:

   ```
   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`As propriedades`LicenseUrl` and podem ser referências a arquivos locais (como no exemplo acima) ou podem ser links para buckets do Amazon S3 que já hospedam esses artefatos.

2. Crie um bucket do S3 no local onde deseja salvar o código empacotado. Se você quiser usar um bucket do S3 existente, ignore esta etapa.

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

3. Crie o pacote de implantação da função Lambda executando o seguinte comando da`package`AWS SAM CLI.

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

O comando faz o seguinte:

- Compacta o conteúdo do`aws-sam/hello_world/` diretório e faz o upload para o Amazon S3.
- Carrega o pacote de implantação, o arquivo README e o arquivo LICENSE para o bucket do Amazon S3 especificado pela`--s3-bucket` opção.
- Gera um novo arquivo de modelo, chamado `packaged.yaml`, que será usado na próxima etapa para publicar o aplicativo no AWS Serverless Application Repository. O arquivo de`packaged.yaml` modelo é semelhante ao arquivo de modelo original (`template.yaml`), mas tem uma diferença fundamental: as`ReadmeUrl` propriedades`CodeUri`,`LicenseUrl`, e apontam para o bucket do Amazon S3 e para os objetos que contêm os respectivos artefatos. O seguinte trecho de um arquivo de modelo de exemplo `packaged.yaml` mostra a propriedade `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/fbd77a3647a4f47a352fc0ObjectGUID

...
```

Etapa 4: publicar o aplicativo

Agora que você criou o pacote de implantação, use-o para publicar o aplicativo no AWS Serverless Application Repository.

Como publicar o aplicativo sem servidor no AWS Serverless Application Repository

- Execute o comando a seguir para publicar o novo aplicativo no AWS Serverless Application Repository com a primeira versão criada como 0.0.1.

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

> **Note**
>
> O aplicativo será criado como privado por padrão. Você deve compartilhar o aplicativo antes que outrasAWS contas possam visualizar e implantar seu aplicativo. Consulte Next Steps (Próximas etapas) abaixo para obter mais detalhes sobre como compartilhar o aplicativo.

Próximas etapas

Agora que você publicou o exemplo de aplicativo, veja algumas coisas que poderá fazer com ele.

- Exibir seu aplicativo emAWS Serverless Application Repository — A saída do`sam publish` comando incluirá um linkAWS Serverless Application Repository diretamente para a página de detalhes do seu aplicativo. Também é possível ir para a página inicial do AWS Serverless Application Repository e pesquisar o aplicativo.

- Compartilhe seu aplicativo — Como seu aplicativo está definido como privado por padrão, ele não é visível para outrasAWS contas. Para compartilhar sua inscrição com outras pessoas, você deve torná-la pública ou conceder permissão a uma lista específica deAWS contas. Para obter informações sobre como compartilhar o aplicativo usando a AWS CLI, consulte AWS Serverless Application RepositoryExemplos de políticas de aplicativos. Para obter informações sobre como compartilhar o aplicativo usando o console, consulte Compartilhar um aplicativo.

Mais informações

Para obter mais informações sobre a seção `Metadata` dos modelos do AWS SAM, sobre os comandos `sam package` e `sam publish` da CLI do AWS SAM, consulte Publicar aplicativos usando a CLI do AWS SAM no Guia do desenvolvedor do AWS Serverless Application Model.

Publicar aplicativos

Quando você publica um aplicativo sem servidor no AWS Serverless Application Repository, ele será disponibilizado para que outros o encontrem e o implantem.

Primeiro você define o aplicativo com um modelo do AWS Serverless Application Model (AWS SAM). Ao definir o aplicativo, é necessário considerar se os consumidores do aplicativo precisarão confirmar os recursos dele. Para obter mais informações sobre como usar o AWS SAM e confirmar os recursos, consulte Como usar o AWS SAM com a AWS Serverless Application Repository.

Você pode publicar aplicativos sem servidor usando oAWS Management Console, oAWS SAMInterface da linha de comando do (AWS SAMCLI), ou umAWSSDK. Para saber mais sobre os procedimentos para publicar aplicativos no AWS Serverless Application Repository, consulte Como publicar aplicativos.

Quando você publica seu aplicativo, ele é inicialmente definido comoprivado, o que significa que só está disponível para oAWSconta que a criou. Para compartilhar seu aplicativo com outras pessoas, você deve defini-lo comoCompartilhado de modo privado(Compartilhado apenas com um conjunto específico deAWScontas), ouPublicly shared(compartilhado com todos).

Quando você publica um aplicativo no AWS Serverless Application Repository e o define como público, o serviço disponibilizará o aplicativo para consumidores de todas as regiões. Quando um consumidor implanta um aplicativo em uma região diferente daquela na qual o aplicativo foi publicado inicialmente, oAWS Serverless Application Repositorycopia os artefatos de implantação do aplicativo para um bucket do Amazon S3 na região de destino. Ele atualiza todos os recursos noAWS SAMModelo que usa esses artefatos para referenciar os arquivos no bucket do Amazon S3 para a região de destino. Os artefatos de implantação podem incluir o código da função do Lambda, os arquivos de definição da API, entre outros.

> **Note**
>
> PrivateeCompartilhado de modo privadoOs aplicativos só estão disponíveis noAWSRegião em que eles são criados. Publicly sharedOs aplicativos estão disponíveis em todosAWSRegiões. Para saber mais sobre como compartilhar aplicativos, consulte AWS Serverless Application RepositoryExemplos de políticas de aplicativos.

Tópicos

- Como usar o AWS SAM com a AWS Serverless Application Repository
- Como publicar aplicativos
- Selo de autor verificado
- Compartilhar camadas do Lambda

Como usar o AWS SAM com a AWS Serverless Application Repository

OAWS Serverless Application Model(AWS SAM) é uma estrutura de código aberto que você pode usar para criaraplicativos sem servidoremAWS. Para obter mais informações sobre como usar AWS SAM para criar seu aplicativo sem servidor, consulte o Guia do desenvolvedor AWS Serverless Application Model.

Ao criar aplicativos que serão publicados noAWS Serverless Application Repository, você deve considerar o conjunto desuportadoAWSRecursos e modelos de políticas disponíveis para uso. As seções abaixo descrevem esses tópicos em mais detalhes.

CompatívelAWSRecursos noAWS Serverless Application Repository

O AWS Serverless Application Repository oferece suporte a aplicativos sem servidor compostos por vários recursos do AWS SAM e do AWS CloudFormation. Para ver a lista completa deAWSrecursos que são suportados porAWS Serverless Application Repository, consulteLista de compatíveisAWSRecursos.

Se você quiser solicitar suporte para um adicionalAWSrecurso, contatoAWSSuporte para.

> **Important**
>
> Se o modelo de aplicativo contém uma das seguintes funções do IAM ou políticas de recursos personalizadas, o aplicativo não será exibido nos resultados da pesquisa por padrão. Além disso, os clientes devem reconhecer as funções do IAM e as políticas de recursos antes de implantá-lo. Para mais informações, consulte Reconhecer recursos do aplicativo.
>
> A lista de recursos ao que ele se aplica são:
>
> - Funções do IAM:AWS::IAM::Group,AWS:: IAM::InstanceProfile,AWS::IAM::Policy, eAWS::IAM::Role.

- Políticas de recursos: AWS:: Lambda::LayerVersionPermissão,AWS::Lambda::Permission,AWS:: Events::EventBusPolítica,AWS:: IAM:Policy,AWS::ApplicationAutoEscalabilidade::ScalingPolicy,AWS:: S3::BucketPolicy,AWS:: SQS::QueuePolicy, eAWS:: SNS:TopicPolicy.

Se o aplicativo contém o recurso AWS::Serverless::Application, os clientes precisam confirmar que o aplicativo contém um aplicativo aninhado antes de poderem implantá-lo. Para mais informações sobre aplicativos aninhados, consulte Aplicativos aninhados no AWS Serverless Application Model Guia do desenvolvedor. Para mais informações sobre reconhecer recursos, consulte Reconhecer recursos do aplicativo.

Modelos de política

AWS SAMO fornece uma lista de modelos de política para definir o escopo das permissões de funções do Lambda para os recursos que são usados pelo aplicativo. O uso de modelos de políticas não exige confirmações adicionais do cliente para pesquisar, navegar ou implantar o aplicativo.

Para ver a lista de padrãoAWS SAMmodelos de política, consulteAWS SAMModelos de política donoAWS Serverless Application ModelGuia do desenvolvedor.

Lista de compatíveisAWSRecursos

Esta é a lista completa deAWSrecursos que são suportados peloAWS 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`

Como publicar aplicativos

Esta seção oferece procedimentos para publicar o aplicativo sem servidor no AWS Serverless Application Repository usando a CLI do AWS SAM ou o AWS Management Console. Ela também mostra como compartilhar seu aplicativo para permitir que outras pessoas o implantem e como excluir seu aplicativo do AWS Serverless Application Repository.

> **Important**
>
> As informações que você insere ao publicar um aplicativo não são criptografadas. Essas informações incluem dados como o nome do autor. Se você tiver informações de identificação pessoal que você não deseja que sejam armazenadas ou tornadas públicas, recomendamos que você não insira essas informações ao publicar o aplicativo.

Publicar um aplicativo (AWS CLI)

A maneira mais fácil de publicar um aplicativo no AWS Serverless Application Repository é usar um conjunto de comandos da CLI do AWS SAM. Para obter mais informações, consulte Publicar um aplicativo usando a CLI do AWS SAM no Guia do desenvolvedor do AWS Serverless Application Model (AWS SAM).

Publicar um novo aplicativo (console)

Esta seção mostra como usar o AWS Management Console para publicar um novo aplicativo no AWS Serverless Application Repository. Para obter instruções sobre como publicar uma nova versão de um aplicativo existente, consulte Publicar a nova versão de um aplicativo existente.

Pré-requisitos

Antes de publicar um aplicativo no AWS Serverless Application Repository, você precisa do seguinte:

- Um válidoAWSconta.
- Um válidoAWS Serverless Application Model(AWS SAM) modelo que define oAWSRecursos que são usados. Para obter mais informações sobre os modelos do AWS SAM, consulte Noções básicas sobre modelos do AWS SAM.
- Um pacote para o aplicativo que você criou usando o comando AWS CloudFormation do `package` para a AWS CLI. Este comando cria um pacote dos artefatos locais (caminhos locais) que são referências do seu modelo AWS SAM. Para obter mais detalhes, consulte package (pacote) na documentação do AWS CloudFormation.
- Um URL que aponta para o código-fonte do aplicativo, caso você queira torná-lo público.
- Um arquivo "readme.txt". Esse arquivo descreve como os clientes podem usar seu aplicativo e como configurá-lo antes de implantá-lo por conta própriaAWScontas.
- Um arquivo license.txt ou um identificador de licença válido do site SPDX. Observe que uma licença é necessária somente se você quiser compartilhar o aplicativo publicamente. Se você quiser manter o aplicativo como privado ou apenas compartilhá-lo de forma privada, não será necessário especificar uma licença.
- Uma política de buckets do Amazon S3 válida que concede as permissões de leitura de serviço para os artefatos que foram carregados no Amazon S3 quando você criou o pacote do seu aplicativo. Para definir essa política, siga essas etapas:

  1. Abra o console do Amazon S3 em https://console.aws.amazon.com/s3/.
  2. Escolha o bucket do Amazon S3 que você usou para empacotar seu aplicativo.
  3. Escolha a guia Permissions (Permissões).
  4. Clique no botão de Política de bucket.
  5. Cole a declaração da política a seguir no Bucket policy editor (Editor de política do bucket). Certifique-se de substituir o nome do bucket no`Resource`elemento e seuAWSUm ID da conta da`Condition`Elemento. A expressão no`Condition`Elementos garantemAWS Serverless

Application Repositorysó tem permissão para acessar aplicativos do especificadoAWSconta. Para obter mais informações sobre declarações de política, consulteReferência de elementos de política JSON do IAMnoManual do usuário do 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. Clique no botão Salvar.

Procedimento

Crie um novo aplicativo no AWS Serverless Application Repository usando o seguinte procedimento.

Para criar um novo aplicativo no AWS Serverless Application Repository

1. Abra o console AWS Serverless Application Repository e escolha Publicar aplicativos.
2. Na página Publish an application (Publicar um aplicativo), insira as seguintes informações do aplicativo e escolha Publish application (Publicar aplicativo):

Propriedade	Obrigatório	Descrição
Application name (Nome do aplicativo)	TRUE	O nome do aplicativo.

Propriedade	Obrigatório	Descrição
		Comprimento mínimo = 1. Tamanho máximo = 140. Padrão: "[a-zA-Z0-9\\-]+";
Author (Autor)	TRUE	O nome do autor que publica o aplicativo. Comprimento mínimo = 1. Tamanho máximo = 127. Padrão: "^[a-z0-9](([a-z0-9]|-(?!-))*[a-z0-9])?$";
Home page (Página inicial)	FALSE	Um URL com mais informações sobre o aplicativo — por exemplo, o local do GitHub repositório para o aplicativo.
Descrição	TRUE	A descrição do aplicativo. Comprimento mínimo = 1. Tamanho máximo = 256.
Rótulos	FALSE	Os rótulos que melhoram a descoberta de aplicativos em resultados de pesquisa. Comprimento mínimo = 1. Tamanho máximo = 127. Número máximo de rótulos: 10. Padrão: "^[a-zA-Z0-9+\\-_:\\/@]+$";

Propriedade	Obrigatório	Descrição
Spdx license (drop-down list) (Licença Spdx (lista suspensa))	FALSE	Escolha um identificador de licença válido no menu suspenso que contenha as licenças disponíveis no site SPDX. Escolher um item no menu suspenso preencher á a caixa de texto License (Licença) abaixo. Observaçã o: Escolher uma licença no menu suspenso substitui rá o conteúdo doLicense (Licença)Caixa de texto e descartará todas as edições manuais feitas.

Propriedade	Obrigatório	Descrição
License (Licença)	FALSE	Carregue um arquivo de licença .txt ou escolha uma licença no menu suspenso Spdx license (Licença Spdx) descrito na linha anterior. Escolher uma licença do menu Spdx license (Licença Spdx) automaticamente preencherá a caixa de texto License (Licença). É possível editar o conteúdo dessa caixa de texto manualmen te após carregar um arquivo de licença ou escolher um no menu suspenso Spdx license (Licença Spdx). No entanto, se outra Spdx license (Licença Spdx) for escolhida no menu suspenso, todas as edições manuais feitas serão descartadas. Esse é um campo opcional, mas é necessário fornecer uma licença para compartil har o aplicativo publicame nte.

Propriedade	Obrigatório	Descrição
Readme (Leiame)	FALSE	Carregue o conteúdo do arquivo Leiame, que pode estar no formato de texto ou markdown. Esses conteúdos são exibidos na página de detalhes do aplicativo no AWS Serverless Application Repository. É possível editar manualmente o conteúdo dessa caixa de texto após carregar um arquivo.
Semantic version	FALSE	A versão semântica do aplicativo. Para obter mais informações, consulte o site do Versionamento Semântico. Forneça um valor para esta propriedade para tornar o aplicativo público.
Source code URL (URL do código-fonte)	FALSE	Um link para um repositório público para o código-fonte do aplicativo.
SAM template (Modelo do SAM)	TRUE	Um válidoAWS Serverles s Application Model(AWS SAM) modelo que define oAWSRecursos que são usados.

Compartilhar um aplicativo

Aplicativos publicados podem ter permissões definidas em uma das três categorias a seguir:

- Privado (padrão)— aplicativos que foram criados com a mesma conta e não foram compartilhados com outrasAWSconta. Somente consumidores que compartilham seuAWSConta tem permissão para implantar aplicativos privados.
- Compartilhado de forma privada— Aplicativos que o editor compartilhou explicitamente com um conjunto específico deAWScontas ou comAWScontas em umAWSOrganização. Os consumidores têm permissão para implantar aplicativos que foram compartilhados com seusAWSconta ouAWSOrganização. Para obter mais informações sobre o AWS Organizations, consulte o Guia do usuário do AWS Organizations.
- Compartilhado publicamente— Aplicativos que o editor compartilhou com todos. Todos os consumidores têm permissão para implantar um aplicativo compartilhado publicamente.

Depois que você publica um aplicativo no AWS Serverless Application Repository, por padrão, ele é definido como privado. Esta seção mostra como compartilhar um aplicativo de modo privado com o específicoAWScontas ou umAWSOrganize ou compartilhe publicamente com todos.

Compartilhar um aplicativo por meio do console

Você tem duas opções para compartilhar seu aplicativo com outras pessoas: 1) Compartilhe com específicoAWScontas ou oAWScontas dentro do seuAWSorganização, ou 2) Compartilhe publicamente com todos. Para obter mais informações sobre o AWS Organizations, consulte o Guia do usuário do AWS Organizations.

Opção 1: Para compartilhar seu aplicativo com específicoAWSconta (s) ou contas dentro da suaAWSorganização

1. Abra o console do AWS Serverless Application Repository.
2. No painel de navegação, escolha Published Applications (Aplicativos publicados) para ver a lista de aplicativos que você criou.
3. Escolha o aplicativo que você quer compartilhar.
4. Escolha a guia Sharing (Compartilhamento).
5. Na seção Application policy statements (Declarações de política de aplicativo), escolha o botão Create Statement (Criar declaração).
6. Na janela Statement Configuration (Configuração da declaração) preencha os campos com base em como você deseja compartilhar seu aplicativo.

> **Note**
>
> Se você estiver compartilhando com uma organização, só será possível especificar a organização que seuAWSconta é membro do. Se você tentar especificar umAWSUma organização da qual você não é membro, ocorrerá um erro.
> Para compartilhar seu aplicativo com seuAWSOrganização, você deve reconhecer que o`UnshareApplication`Uma ação será adicionada à declaração de política, caso o compartilhamento precise ser revogado no future.

7. Clique no botão Salvar.

Opção 2: Para compartilhar seu aplicativo publicamente com todos

1. Abra o console do AWS Serverless Application Repository.
2. No painel de navegação, escolha Published Applications (Aplicativos publicados) para ver a lista de aplicativos que você criou.
3. Escolha o aplicativo que você quer compartilhar.
4. Escolha a guia Sharing (Compartilhamento).
5. Na seção Public Sharing (Compartilhamento público) escolha o botão Edit (Editar).
6. Em Public sharing (Compartilhamento público) escolha o botão de opção Enabled (Habilitado).
7. Na caixa de texto digite o nome do aplicativo e escolha o botão Save (Salvar).

> **Note**
>
> Para compartilhar um aplicativo publicamente, ele deve ter as propriedades `LicenseUrl` e `SemanticVersion` definidas.

Compartilhar um aplicativo por meio da AWS CLI

Para compartilhar uma aplicação usando aAWS CLIvocê concede permissões usando o`put-application-policy`comando para especificar oAWSConta (s) com as quais você deseja compartilhar como diretores.

Para obter mais informações sobre como compartilhar seu aplicativo usando aAWSCLI, consulteAWS Serverless Application RepositoryExemplos de políticas de aplicativos.

Cancelar o compartilhamento de um aplicativo

Há duas opções para cancelar o compartilhamento de um aplicativo de umAWSOrganização:

1. O editor do aplicativo pode remover permissões usando o comando `put-application-policy`.
2. Um usuário doconta de gerenciamentode umAWSA organização pode executar umcancelar compartilhamento de aplicativoOperation em qualquer aplicativo compartilhado com a organização, mesmo que o aplicativo tenha sido publicado por um usuário de uma conta diferente.

> **Note**
>
> Quando um aplicativo não é compartilhado de umAWSOrganização com a operação “cancelar compartilhamento de aplicativo”, ela não pode ser compartilhada comAWSOrganização novamente.

Para obter mais informações sobre o AWS Organizations, consulte o Guia do usuário do AWS Organizations.

Editor remove permissões

Editor remove permissões pelo console

Para cancelar o compartilhamento de um aplicativo por meio doAWS Management Console, remova a declaração de política que a compartilha com outrasAWScontas. Para isso, siga estas etapas:

1. Abra o console do AWS Serverless Application Repository.
2. Escolha Available Applications (Aplicativos disponíveis) no painel de navegação esquerdo.
3. Escolha o aplicativo do qual deseja cancelar o compartilhamento.
4. Escolha a guia Sharing (Compartilhamento).
5. Na seção Application policy statements (Declarações de política de aplicativo) selecione a declaração de política que está compartilhando o aplicativo com as contas das quais você deseja cancelar o compartilhamento.
6. Escolha Delete (Excluir).

7. Uma mensagem de confirmação será exibida. Escolha Delete (Excluir) novamente.

Editor remove permissões pela AWS CLI

Para cancelar o compartilhamento de um aplicativo por meio doAWS CLI, o editor pode remover ou alterar permissões usando o`put-application-policy`comando para tornar o aplicativo privado ou compartilhar com um conjunto diferente deAWScontas.

Para obter mais informações sobre como alterar permissões usando aAWSCLI, consulteAWS Serverless Application RepositoryExemplos de políticas de aplicativos.

Conta de gerenciamento cancelando o compartilhamento de um aplicativo

Conta de gerenciamento cancelando o compartilhamento de um aplicativo de umAWSOrganização pelo console do

Para cancelar o compartilhamento de um aplicativo de umAWSOrganização através doAWS Management Console, um usuário doconta de gerenciamentoO pode fazer o seguinte:

1. Abra o console do AWS Serverless Application Repository.
2. Escolha Available Applications (Aplicativos disponíveis) no painel de navegação esquerdo.
3. No bloco do aplicativo, escolha Unshare (Cancelar compartilhamento).
4. Na caixa de mensagem de cancelamento de compartilhamento, confirme que deseja cancelar o compartilhamento do aplicativo inserindo o ID da organização e o nome do aplicativo e escolhendo Save (Salvar).

Conta de gerenciamento cancelando o compartilhamento de um aplicativo de umAWSOrganização através doAWS CLI

Para cancelar o compartilhamento de um aplicativo de umAWSOrganização, um usuário doconta de gerenciamentopode executar o`aws serverlessrepo unshare-application`comando.

O comando a seguir descompartilha um aplicativo de umAWSOrganização, onde*`Application-id`*É o nome de recurso da Amazon (ARN) do aplicativo e*`id da organização`*é oAWSID da organização:

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

Excluir um aplicativo

É possível excluir aplicativos do AWS Serverless Application Repository usando o AWS Management Console ou a CLI do AWS SAM.

Excluir um aplicativo (console)

Para excluir um aplicativo publicado por meio do AWS Management Console, faça o seguinte.

1. Abra o console do AWS Serverless Application Repository.
2. Em My Applications (Meus aplicativos), escolha o aplicativo a ser excluído.
3. Na página de detalhes do aplicativo, escolha Delete application (Excluir aplicativo).
4. Escolha Delete application (Excluir aplicativo) para concluir a exclusão.

Excluir um aplicativo (AWS CLI)

Para excluir um aplicativo publicado usando a AWS CLI, execute o comando `aws serverlessrepo delete-application`.

O comando a seguir exclui um aplicativo, onde *`application-id`* é o nome de recurso da Amazon (ARN) do aplicativo:

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

Publicar a nova versão de um aplicativo existente

Esta seção mostra como publicar uma nova versão de um aplicativo existente no AWS Serverless Application Repository usando a CLI do AWS SAM ou o AWS Management Console. Para obter instruções sobre como publicar um novo aplicativo, consulte Como publicar aplicativos.

Publicar a nova versão de um aplicativo existente (AWS CLI)

A maneira mais fácil de publicar uma nova versão de um aplicativo existente é usar um conjunto de comandos da CLI do AWS SAM. Para obter mais informações, consulte Publicar um aplicativo usando a CLI do AWS SAM no Guia do desenvolvedor do AWS Serverless Application Model (AWS SAM).

Publicar a nova versão de um aplicativo existente (console)

Para publicar uma nova versão de um aplicativo que você publicou anteriormente, siga estas etapas:

1. Abra o console do AWS Serverless Application Repository.
2. No painel de navegação, escolha My Applications (Meus aplicativos) para ver a lista de aplicativos que você criou.
3. Escolha o aplicativo para o qual você quer publicar uma nova versão.
4. Escolha Publish new version (Publicar nova versão).
5. Em Versions (Versões), insira as informações do aplicativo:

Propriedade	Obrigatório	Descrição
Semantic version	TRUE	A versão semântica do aplicativo. Para obter mais informações, consulte o site do Versionamento Semântico. Forneça um valor para esta propriedade para tornar o aplicativo público.
Source code URL (URL do código-fonte)	FALSE	Um link para um repositório público para o código-fonte do aplicativo.
SAM template (Modelo do SAM)	TRUE	Um válidoAWS Serverles s Application Model(AWS SAM) modelo que define oAWSRecursos que são usados.

6. Escolha Publish version (Publicar versão).

Selo de autor verificado

Autores verificadosnoAWS Serverless Application Repositorysão aqueles para os quaisAWSÉ uma análise de boa-fé, como provedor de serviços coerente e prudente, das informações fornecidas pelo solicitante e confirma que a identidade do solicitante é verdadeira.

Os aplicativos de autores verificados exibem um selo de autor verificado, junto com um link para o perfil público do autor. O selo de autor verificado é exibido tanto nos resultados de pesquisa quanto na página de detalhes do aplicativo.

Solicitar um selo de autor verificado

É possível solicitar a aprovação como autor verificado noAWS Serverless Application Repositoryenviando um e-mail paraserverlessrepo-verified-author@amazon .com. Você precisará fornecer as seguintes informações:

- Nome do autor
- AWSID da conta da
- Link do perfil publicamente acessível, como oGitHubouLinkedInperfil

Depois de enviar uma solicitação para obter um selo de autor verificado, a resposta doAWSDentro de alguns dias. É possível ser solicitado a enviar informações adicionais antes da solicitação ser aprovada.

Após a solicitação ser aprovada, o selo de autor verificado será exibido para seus aplicativos em um dia.

> **Note**
>
> O selo de autor verificado será exibido para todos os aplicativos que correspondem aos doisAWSnome da conta e do autor. ComoAWSAs contas podem ter vários autores, os selos não são exibidos em aplicativos com um nome de autor diferente. Para que os selos de autores sejam exibidos nos aplicativos com diferentes nomes de autor, é necessário enviar outra solicitação para esse autor.

Compartilhar camadas do Lambda

Se você implementou a funcionalidade em uma camada do Lambda, talvez queira compartilhar a camada sem hospedar uma instância global dela. Compartilhar camadas dessa forma permite que outros implantem uma instância da camada na própria conta. Isso impede que os aplicativos cliente dependam de uma instância global da camada. OAWS Serverless Application RepositoryIsso permite compartilhar camadas do Lambda facilmente dessa forma.

Para obter mais informações sobre camadas do Lambda, consulteAWS LambdaCamadas donoAWS LambdaGuia do desenvolvedor.

Como funcionam

Veja a seguir as etapas para compartilhar a camada usando o AWS Serverless Application Repository. Isso permite que uma cópia da camada seja criada noAWSconta.

1. Defina um aplicativo sem servidor com umAWS SAMModelo do que inclua a camada como recurso — ou seja, um`AWS::Serverless::LayerVersion`ou um`AWS::Lambda::LayerVersion`recurso.
2. Publique o aplicativo no AWS Serverless Application Repository e compartilhe-o (pública ou privadamente).
3. Um cliente implanta o aplicativo, o que cria uma cópia da camada por conta própriaAWSconta. O cliente pode referenciar o nome de recurso da Amazon (ARN) da camada noAWSConta no aplicativo cliente.

Exemplo

Veja a seguir um exemplo doAWS SAMModelo do para um aplicativo que contém a camada do Lambda que deseja compartilhar:

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

```
Outputs:
  LayerArn:
    Value: !Ref SharedLayer
```

Quando um cliente implanta o aplicativo a partir doAWS Serverless Application Repository, uma camada é criada em seusAWSconta. O ARN da camada se parece com o seguinte:

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

O cliente pode referenciar esse ARN para o próprio aplicativo cliente, como neste exemplo:

```
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
```

Como implantar aplicativos

Esta seção ajuda você a aprender a localizar e implantar aplicativos sem servidor que foram publicados no AWS Serverless Application Repository. É possível procurar aplicativos que estão disponíveis publicamente sem ter umAWSconta visitando osite público. Se preferir, você pode pesquisar aplicativos no console do AWS Lambda.

Alguns aplicativos têm um selo de autor verificado, com um link para o perfil do autor. Um autor é considerado umautor verificadoquandoAWSEle fez uma análise de boa-fé, como provedor de serviços coerente e prudente, das informações fornecidas pelo solicitante e confirma que a identidade do solicitante é verdadeira.

Antes de implantar aplicativos do AWS Serverless Application Repository, consulte os seguintes tópicos para saber mais sobre as permissões de implantação e os recursos do aplicativo.

Tópicos

- Permissões de implantação do aplicativo
- Recursos do aplicativo: funções do IAM, políticas de recursos e aplicativos aninhados
- Como implantar aplicativos

Permissões de implantação do aplicativo

Para implantar um aplicativo no AWS Serverless Application Repository, é necessário ter uma permissão. Existem três categorias de aplicativos para os quais você tem permissão para implantar:

- Private— Os aplicativos que foram criados com a mesma conta e não foram compartilhados com outras contas. Você tem permissão para implantar aplicativos que foram criados usando oAWSconta.
- Compartilhado de forma privada— Compartilhado explicitamente aplicativos que o editor compartilhou com um conjunto específico deAWScontas. Você tem permissão para implantar aplicativos que foram compartilhados com o seuAWSconta.
- Compartilhado publicamente— Compartilhado pelo editor aplicativos que o editor compartilhou com todos. Você tem permissão para implantar aplicativos compartilhados publicamente.

É possível apenas pesquisar e explorar os aplicativos para os quais tem permissão. Isso inclui aplicativos que foram criados usando oAWSconta compartilhada em particular com suaAWSconta e compartilhada publicamente. Nenhum dos outros aplicativos será exibido para você.

> **Important**
>
> Os aplicativos que contêm aplicativos aninhados herdam as restrições de compartilhamento dos aplicativos aninhados. Por exemplo, suponha que um aplicativo seja compartilhado publicamente, mas contém um aplicativo aninhado que é compartilhado apenas de forma privada com o aplicativoAWSconta que criou o aplicativo pai. Nesse caso, se o seuAWSA conta não tem permissão para implantar o aplicativo aninhado e não é possível implantar o aplicativo pai. Para mais informações sobre aplicativos aninhados, consulte Aplicativos aninhados no AWS Serverless Application Model Guia do desenvolvedor.

Recursos do aplicativo: funções do IAM, políticas de recursos e aplicativos aninhados

Antes de implantar um aplicativo, eleAWS Serverless Application Repository verifica o modelo do aplicativo em busca de funções do IAM, políticas deAWS recursos e aplicativos aninhados que o modelo especifica que ele deve criar. Os recursos do IAM, como uma função do IAM com acesso completo, podem modificar qualquer recurso na suaAWS conta da. Portanto, recomendamos que você revise as permissões associadas ao aplicativo antes de prosseguir, para não criar recursos acidentalmente com permissões escalonadas. Para garantir que você fez isso, confirme que o aplicativo possui os recursos antes que AWS Serverless Application Repository implante o aplicativo por você.

Os aplicativos podem conter qualquer um dos quatro seguintes recursos: `CAPABILITY_IAM`, `CAPABILITY_NAMED_IAM`, `CAPABILITY_RESOURCE_POLICY` e `CAPABILITY_AUTO_EXPAND`.

Os seguintes recursos exigem que você especifique`CAPABILITY_IAM` ou`CAPABILITY_NAMED_IAM`: AWS::IAM::GroupAWS::IAM::InstanceProfileAWS::IAM::Policy,, AWS::IAM::Role e. Se o aplicativo contém recursos do IAM com nomes personalizados, você deverá especificar `CAPABILITY_NAMED_IAM`. Para obter um exemplo de como especificar recursos, consulte Encontrar e confirmar recursos do aplicativo (AWS CLI).

Os seguintes recursos exigem que você especifique`CAPABILITY_RESOURCE_POLICY`: AWS::Lambda::LayerVersionPermissão AWS::Lambda::Permission, AWS::Events::EventBusPolítica,

AWS: :iam:Policy AWS::ApplicationAutoScaling::ScalingPolicyAWS::S3::BucketPolicy, AWS::SQS::QueuePolicy,, AWS::SNS::TopicPolicye.

Os aplicativos que contêm um ou mais aplicativos aninhados exigem que você especifique `CAPABILITY_AUTO_EXPAND`. Para mais informações sobre aplicativos aninhados, consulte Aplicativos aninhados no AWS Serverless Application Model Guia do desenvolvedor.

Encontrar e confirmar recursos do aplicativo (console)

Você pode encontrar aplicativos disponíveisAWS Serverless Application Repository no AWS Serverless Application Repositorysite ou por meio do console do Lambda (na página Criar função, abaixo daAWS Serverless Application Repository guia).

Os aplicativos que exigem a confirmação de recursos para criar funções do IAM ou políticas de recursos personalizadas não aparecem em resultados de pesquisa por padrão. Para pesquisar aplicativos que contenham esses recursos, marque a caixa de seleção Show apps that create custom IAM roles or resource policies (Mostrar aplicativos que criam funções do IAM ou políticas de recursos personalizadas).

Você pode revisar os recursos de um aplicativo na guia Permissões ao selecionar o aplicativo. Para implantar o aplicativo, você precisa selecionar I acknowledge this application creates custom IAM roles or resource policies (Entendo que esse aplicativo cria funções do IAM ou políticas de recursos personalizadas). Se você não reconhecer esses recursos, verá esta mensagem de erro: Confirmação necessária. Para implantar, marque a caixa na seção Configurar parâmetros do aplicativo.

Visualizar recursos do aplicativo (AWS CLI)

Para visualizar os recursos de um aplicativo usando a AWS CLI, primeiro você precisa do nome de recurso da Amazon (ARN). Depois, poderá executar o seguinte comando:

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

A resposta requiredCapabilities contém uma lista de recursos do aplicativo que você precisará reconhecer antes de implantar o aplicativo. Observe que se a propriedade requiredCapabilities estiver vazia, o aplicativo não possui os recursos necessários.

Como implantar aplicativos

Esta seção oferece procedimentos para implantar aplicativos sem servidor no AWS Serverless Application Repository usando o AWS Management Console ou a AWS CLI.

Implantar um novo aplicativo (console)

Esta seção mostra como implantar um novo aplicativo do AWS Serverless Application Repository usando o AWS Management Console. Para obter instruções sobre como implantar uma nova versão de um aplicativo existente, consulte Atualizar aplicativos.

Navegar, pesquisar e implantar aplicativos

Encontre, configure e implemente um aplicativo no AWS Serverless Application Repository usando o seguinte procedimento.

Para encontrar e configurar um aplicativo no AWS Serverless Application Repository

1. Abra a página inicial pública do AWS Serverless Application Repository ou abra o AWS Lambda console. Escolha Create function (Criar função) e selecione Browse serverless app repository (Navegar repositório de aplicativos sem servidor).
2. Pesquise um navegue até um aplicativo.

   > **Note**
   >
   > Para mostrar os aplicativos que contêm funções do IAM ou políticas de recursos personalizadas, selecione o Mostrar aplicativos que criam funções do IAM ou políticas de recursos personalizadas na caixa de seleção . Para obter mais informações sobre funções do IAM e políticas de recursos personalizadas, consulte Reconhecer recursos do aplicativo.

3. Escolha um aplicativo para ver detalhes como suas permissões, recursos e o número de vezes que ele foi implantado pelosAWS clientes.

   As contagens de implantação são mostradas para aAWS região em que você está tentando implantar o aplicativo.
4. Na página de detalhes do aplicativo, visualize as permissões e os recursos do aplicativo visualizando o modeloAWS SAM, a licença e o arquivo Leia-me. Nesta página, você também pode encontrar o link Source code URL (URL do código-fonte) de aplicativos que são

compartilhados publicamente. Se o aplicativo inclui algum aplicativo aninhado, você também pode visualizar os detalhes desses aplicativos nesta página.

5. Configure o aplicativo na seção Configurações do aplicativo. Para obter orientações sobre como configurar um aplicativo específico, consulte o arquivo "readme" do aplicativo em questão.

   Por exemplo, os requisitos de configuração podem incluir a especificação do nome de um recurso ao qual você deseja que o aplicativo tenha acesso. Esse recurso pode ser uma tabela do Amazon DynamoDB, um bucket do Amazon S3 ou uma API do Amazon API Gateway.

6. Escolha Implantar. Isso direcionará você para a página Deployment status.

   > **Note**
   >
   > Se o aplicativo tiver recursos que exigem confirmação, marque a caixa de seleção I acknowledge this application creates custom IAM roles or resource polices (Eu entendo que este aplicativo cria funções do IAM ou políticas de recursos personalizadas) antes de implantar o aplicativo. Caso contrário, ocorrerá um erro. Para obter mais informações sobre funções do IAM e políticas de recursos personalizadas, consulte Reconhecer recursos do aplicativo.

7. Na página Deployment status (Status da implantação) você pode visualizar o progresso da implantação. Enquanto aguarda a conclusão da implantação, você pode pesquisar e procurar outros aplicativos e retornar a esta página por meio do console do Lambda.

Depois que seu aplicativo for implantado com êxito, você poderá revisar e gerenciar os recursos que foram criados usandoAWS as ferramentas existentes.

Implantar um novo aplicativo (AWS CLI)

Esta seção mostra como implantar um novo aplicativo do AWS Serverless Application Repository usando a AWS CLI. Para obter instruções sobre como implantar uma nova versão de um aplicativo existente, consulte Atualizar aplicativos.

Encontrar e confirmar recursos do aplicativo (AWS CLI)

Para confirmar os recursos de um aplicativo usando o AWS CLI, siga estas etapas:

1. Analise os recursos do aplicativo.Use oAWS CLI comando a seguir para revisar os recursos de um aplicativo:

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

A resposta requiredCapabilities contém uma lista de recursos do aplicativo que você precisará reconhecer antes de implantar o aplicativo. Você também pode usar a GetApplication API nosAWS SDKs para obter esses dados.

2. Crie o conjunto de alterações.Você deve fornecer o conjunto de recursos necessários ao criar oAWS CloudFormation conjunto de alterações. Por exemplo, use o comando a seguir AWS CLI para implantar um aplicativo ao reconhecer os recursos:

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

O ID do conjunto de alterações é retornado quando esse comando for executado com êxito. Você precisa do ID do conjunto de alterações para ir para a próxima etapa. Você também pode usar a CreateCloudFormationChangeSet API nosAWS SDKs para criar o conjunto de alterações.

Por exemplo, oAWS CLI comando a seguir reconhece um aplicativo que contém um AWS::IAM::Rolerecurso com um nome personalizado e um ou mais aplicativos aninhados:

```
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. Execute o conjunto de alterações.Executar o conjunto de alterações realiza a implantação. Forneça o ID do conjunto de alterações que foi retornado ao criar o conjunto de alterações na etapa anterior.

O comando da AWS CLI do exemplo a seguir executa o conjunto de alterações do aplicativo para implantar o aplicativo:

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

Você também pode usar a ExecuteChangeSet API nosAWS SDKs para executar o conjunto de alterações.

Exclusão de pilhas do aplicativo

Para excluir um aplicativo implantado anteriormente usando o AWS Serverless Application Repository, siga o mesmo procedimento para excluir uma pilha do AWS CloudFormation:

- AWS Management Console: Para excluir um aplicativo usando oAWS Management Console, consulte Excluindo uma pilha noAWS CloudFormation console no Guia doAWS CloudFormation usuário.
- AWS CLI: Para excluir um aplicativo usando oAWS CLI, consulte Excluindo uma pilha no Guia doAWS CloudFormation usuário.

Atualizar aplicativos

Após implantar um aplicativo do AWS Serverless Application Repository, talvez seja necessário atualizá-lo. Por exemplo, talvez queira alterar a configuração de um aplicativo, ou atualizar o aplicativo para a versão mais recente publicada.

As seções a seguir descrevem como implantar uma nova versão de um aplicativo ao inserir o AWS Management Console ou a AWS CLI.

Atualizar aplicativos (console)

Para atualizar um aplicativo implantado previamente, use o mesmo procedimento usado para implantar um novo aplicativo e forneça o mesmo nome de aplicativo que foi usado originalmente para implantá-lo. Em particular, o AWS Serverless Application Repository anexa o `serverlessrepo-` ao nome do aplicativo. No entanto, para implantar uma nova versão do aplicativo, forneça o nome do aplicativo original sem o `serverlessrepo-` anexado.

Por exemplo, se você implantou um aplicativo com o nome `MyApplication`, o nome da pilha será `serverlessrepo-MyApplication`. Para atualizar esse aplicativo, você forneceria o nome`MyApplication` novamente — não especifique o nome completo da pilha de`serverlessrepo-MyApplication`.

Para todas as demais configurações de aplicativo, é possível manter os valores iguais aos da implantação anterior ou fornecer novos valores.

Atualizar aplicativos (AWS CLI)

Para atualizar um aplicativo implantado previamente, use o mesmo procedimento usado para implantar um novo aplicativo e forneça o mesmo `--stack-name` que foi usado originalmente para implantá-lo. Em particular, o AWS Serverless Application Repository anexa o `serverlessrepo-` ao nome da pilha. No entanto, para implantar uma nova versão do aplicativo, forneça o nome da pilha original sem o `serverlessrepo-` anexado.

Por exemplo, se você implantou um aplicativo com o nome da pilha `MyApplication`, o nome da pilha criada será `serverlessrepo-MyApplication`. Para atualizar esse aplicativo, você forneceria o nome`MyApplication` novamente — não especifique o nome completo da pilha de`serverlessrepo-MyApplication`.

Segurança no AWS Serverless Application Repository

A segurança da nuvem na AWS é a nossa maior prioridade. Como cliente da AWS, você se contará com um datacenter e uma arquitetura de rede criados para atender aos requisitos das organizações com as maiores exigências de segurança.

A segurança é uma responsabilidade compartilhada entre a AWS e você. O modelo de responsabilidade compartilhada descreve isto como segurança da nuvem e segurança na nuvem:

- Segurança da nuvem: a AWS é responsável pela proteção da infraestrutura que executa produtos da AWS na Nuvem AWS. A AWS também fornece serviços que podem ser usados com segurança. Auditores de terceiros testam e verificam regularmente a eficácia da nossa segurança como parte dos programas de conformidade da AWS. Para saber mais sobre os programas de conformidade que se aplicam ao AWS Serverless Application Repository, consulte Serviços da AWS em escopo por programa de conformidade.
- Segurança da nuvem: sua responsabilidade é determinada pelo serviço da AWS que você usa. Você também é responsável por outros fatores, incluindo a confidencialidade de seus dados, os requisitos da sua empresa e as leis e regulamentos aplicáveis.

Esta documentação ajuda você a entender como aplicar o modelo de responsabilidade compartilhada ao usar o AWS Serverless Application Repository. Os tópicos a seguir mostram como configurar a AWS Serverless Application Repository para atender aos seus objetivos de segurança e conformidade. Saiba também como usar outros produtos da AWS que ajudam a monitorar e proteger os recursos do AWS Serverless Application Repository.

Tópicos

- Proteção de dados no AWS Serverless Application Repository
- Identity and Access Management para o AWS Serverless Application Repository
- Registro em log e monitoramento no AWS Serverless Application Repository
- Validação de conformidade do AWS Serverless Application Repository
- Resiliência no AWS Serverless Application Repository
- Segurança da infraestrutura no AWS Serverless Application Repository

Proteção de dados no AWS Serverless Application Repository

O modelo de responsabilidade compartilhada da AWS se aplica à proteção de dados no AWS Serverless Application Repository. Conforme descrito nesse modelo, a AWS é responsável por proteger a infraestrutura global que executa toda a Nuvem AWS. Você é responsável por manter o controle sobre seu conteúdo hospedado nessa infraestrutura. Você também é responsável pelas tarefas de configuração e gerenciamento de segurança dos Serviços da AWS que usa. Para ter mais informações sobre a privacidade de dados, consulte as Perguntas frequentes sobre privacidade de dados. Para ter mais informações sobre a proteção de dados na Europa, consulte a postagem do blog AWS Shared Responsibility Model and GDPR no Blog de segurança da AWS.

Para fins de proteção de dados, recomendamos que você proteja as credenciais da Conta da AWS e configure as contas de usuário individuais com o AWS IAM Identity Center ou o AWS Identity and Access Management (IAM). Dessa maneira, cada usuário receberá apenas as permissões necessárias para cumprir suas obrigações de trabalho. Recomendamos também que você proteja seus dados das seguintes formas:

- Use uma autenticação multifator (MFA [multi-factor authentication]) com cada conta.
- Use SSL/TLS para se comunicar com os recursos da AWS. Exigimos TLS 1.2 e recomendamos TLS 1.3.
- Configure o registro em log das atividades da API e do usuário com o AWS CloudTrail.
- Use as soluções de criptografia da AWS, juntamente com todos os controles de segurança padrão dos Serviços da AWS.
- Use serviços gerenciados de segurança avançada, como o Amazon Macie, que ajuda a localizar e proteger dados sigilosos armazenados no Amazon S3.
- Se você precisar de módulos criptográficos validados pelo FIPS 140-2 ao acessar a AWS por meio de uma interface de linha de comando ou uma API, use um endpoint do FIPS. Para ter mais informações sobre endpoints do FIPS disponíveis, consulte Federal Information Processing Standard (FIPS) 140-2.

É altamente recomendável que nunca sejam colocadas informações de identificação confidenciais, como endereços de email dos seus clientes, em marcações ou campos de formato livre, como um campo Nome. Isso inclui trabalhar com a AWS Serverless Application Repository ou outros Serviços da AWS usando o console, a API, a AWS CLI ou os AWS SDKs. Quaisquer dados inseridos em tags ou campos de texto de formato livre usados para nomes podem ser usados para logs de faturamento ou de diagnóstico. Se você fornecer um URL para um servidor externo, recomendemos

fortemente que não sejam incluídas informações de credenciais no URL para validar a solicitação a esse servidor.

Criptografia em trânsito

Os endpoints da API do AWS Serverless Application Repository oferecem suporte a conexões seguras somente em HTTPS. Ao gerenciar recursos do AWS Serverless Application Repository com o AWS Management Console, o AWS SDK ou a API do AWS Serverless Application Repository, toda a comunicação é criptografada com Transport Layer Security (TLS).

Para obter uma lista completa dos endpoints da API, consulte AWSRegiões e endpoints no. Referência geral da AWS

Criptografia em repouso

O AWS Serverless Application Repository criptografa os arquivos que você faz upload para o AWS Serverless Application Repository, incluindo pacotes de implantação e arquivos de camada.

Identity and Access Management para o AWS Serverless Application Repository

AWS Identity and Access Management (IAM) é um serviço da AWS service (Serviço da AWS) que ajuda o administrador no controle de segurança de acesso aos recursos da AWS de forma segura. Os administradores do IAM controlam quem pode ser autenticado (fazer login) e autorizado (ter permissões) a usar os recursos do AWS Serverless Application Repository. O IAM é um AWS service (Serviço da AWS) que pode ser usado sem custo adicional.

Para ter uma visão geral de como o IAM funciona, consulte Entendendo como o IAM funciona no Guia do usuário do IAM.

Tópicos

- Público
- Autenticação com identidades
- Gerenciamento do acesso usando políticas
- Como aAWS Serverless Application Repository funcionam com o IAM
- Exemplos de políticas baseadas em identidade do AWS Serverless Application Repository

- AWS Serverless Application RepositoryExemplos de políticas de aplicativos
- AWS Serverless Application RepositoryPermissões da API do: Referência de ações e recursos
- Solução de problemas de identidade e acesso AWS Serverless Application Repository

Público

O uso do AWS Identity and Access Management (IAM) varia dependendo do trabalho que for realizado no AWS Serverless Application Repository.

Usuário do serviço – Se você usa o serviço AWS Serverless Application Repository para fazer o trabalho, o administrador fornece as credenciais e as permissões necessárias. À medida que usar mais recursos do AWS Serverless Application Repository para seu trabalho, você poderá precisar de permissões adicionais. Entender como o acesso é gerenciado pode ajudá-lo a solicitar as permissões corretas ao seu administrador. Se não for possível acessar um recurso no AWS Serverless Application Repository, consulte Solução de problemas de identidade e acesso AWS Serverless Application Repository.

Administrador do serviço – Se você for o responsável pelos recursos do AWS Serverless Application Repository na empresa, provavelmente terá acesso total ao AWS Serverless Application Repository. Cabe a você determinar quais funcionalidades e recursos do AWS Serverless Application Repository os usuários do serviço devem acessar. Assim, você deve enviar solicitações ao administrador do IAM para alterar as permissões dos usuários de seu serviço. Analise as informações nesta página para entender os conceitos básicos do IAM. Saiba mais sobre como sua empresa pode usar o IAM com AWS Serverless Application Repository consultando Como aAWS Serverless Application Repository funcionam com o IAM.

Administrador do IAM – Se você for um administrador do IAM, talvez queira saber detalhes sobre como pode gravar políticas para gerenciar acesso ao AWS Serverless Application Repository. Veja o exemplo de políticas baseadas em identidade AWS Serverless Application Repository que podem ser usadas no IAM consultando Exemplos de políticas baseadas em identidade do AWS Serverless Application Repository.

Autenticação com identidades

A autenticação é a forma como você faz login na AWS usando suas credenciais de identidade. Você deve ser autenticado (ter feito login em AWS) como o usuário raiz da Usuário raiz da conta da AWS, como usuário do IAM ou assumindo um perfil do IAM.

Você pode ter feito login em AWS como uma identidade federada usando credenciais fornecidas por meio de origem de identidade. AWS IAM Identity Center Os usuários do IAM Identity Center [Centro de Identidade], a autenticação única da empresa e as suas credenciais do Google ou do Facebook são exemplos de identidades federadas. Quando você faz login como identidade federada, o administrador já configurou anteriormente a federação de identidades usando perfis do IAM. Quando você acessa a AWS usando a federação, está indiretamente assumindo um perfil.

A depender do tipo de usuário que você seja, você pode fazer login no AWS Management Console ou no portal de acesso da AWS. Para obter mais informações sobre como fazer login em AWS, consulte Como Fazer Login na sua Conta da AWS no Início de Sessão da AWS Manual do usuário.

Se você acessar AWS programaticamente, a AWS fornecerá um kit de desenvolvimento de software (SDK) e uma interface de linha de comando (CLI) para assinar criptograficamente suas solicitações usando as suas credenciais. Se você não utilizar as ferramentas da AWS, deverá assinar as solicitações por conta própria. Para obter mais informações sobre como usar o método recomendado para assinar solicitações por conta própria, consulte Assinando Solicitações de API AWS noGuia do Usuário do IAM.

Independente do método de autenticação usado, também pode ser exigido que você forneça informações adicionais de segurança. Por exemplo, a AWS recomenda o uso da autenticação multifator (MFA) para aumentar a segurança de sua conta. Para saber mais, consulte Autenticação Multifator no AWS IAM Identity CenterGuia do Usuário. Usar a autenticação multifator (MFA) na AWS no Guia do Usuário do IAM.

Usuário raiz da Conta da AWS

Ao criar uma Conta da AWS, você começa com uma identidade de login que tem acesso completo a todos os recursos e Serviços da AWS na conta. Essa identidade, denominada usuário raiz da Conta da AWS, e é acessada por login com o endereço de e-mail e a senha que você usou para criar a conta. É altamente recomendável não usar o usuário raiz para tarefas diárias. Proteja as credenciais do usuário raiz e use-as para executar as tarefas que somente ele pode executar. Para obter a lista completa das tarefas que exigem login como usuário raiz, consulte Tarefas que exigem credenciais de usuário raiz no Guia do usuário do IAM.

Grupos e usuários do IAM

Um usuário do IAM é uma identidade dentro da Conta da AWS que tem permissões específicas para uma única pessoa ou aplicativo. Sempre que possível, recomendamos depender de credenciais temporárias em vez de criar usuários do IAM com credenciais de longo prazo, como senhas e

chaves de acesso. No entanto, se você tiver casos de uso específicos que exijam credenciais de longo prazo com usuários do IAM, recomendamos alternar as chaves de acesso. Para obter mais informações, consulte Alterne Chaves de Acesso Regularmente para Casos de Uso que exijam Credenciais de Longo Prazo no Guia do Usuário do IAM.

Um grupo do IAM é uma identidade que especifica uma coleção de usuários do IAM. Não é possível fazer login como um grupo. É possível usar grupos para especificar permissões para vários usuários de uma vez. Os grupos facilitam o gerenciamento de permissões para grandes conjuntos de usuários. Por exemplo, você pode ter um nome de grupo IAMAdmins e atribuir a esse grupo permissões para administrar recursos do IAM.

Usuários são diferentes de perfis. Um usuário é exclusivamente associado a uma pessoa ou a um aplicativo, mas uma função pode ser assumida por qualquer pessoa que precisar dela. Os usuários têm credenciais permanentes de longo prazo, mas os perfis fornecem credenciais temporárias. Para saber mais, consulte Quando Criar um Usuário do IAM (Ao Invés de uma Função) no Guia do Usuário do IAM.

Funções do IAM

Um perfil do IAM é uma identidade dentro da Conta da AWS que tem permissões específicas. Ele é semelhante a um usuário do IAM, mas não está associado a uma pessoa específica. É possível assumir temporariamente um perfil do IAM em AWS Management Console alternando perfis. É possível assumir um perfil chamando uma operação de API da AWS CLI ou da AWS, ou mesmo usando URL personalizada. Para obter mais informações sobre métodos para usar perfis, consulte Usando Funções do IAM no Guia do Usuário do IAM.

Funções do IAM com credenciais temporárias são úteis nas seguintes situações:

- Acesso de usuário federado –- para atribuir permissões a identidades federadas, você pode criar um perfil e definir permissões para ele. Quando uma identidade federada é autenticada, essa identidade é associada ao perfil e recebe as permissões definidas pelo mesmo. Para obter mais informações sobre perfis para federação, consulte Criando um Perfil para um Provedor de Identidades Terceirizado no Guia do Usuário do IAM. Se você usa o IAM Identity Center, configure um conjunto de permissões. Para controlar o que suas identidades podem acessar após a autenticação, o IAM Identity Center correlaciona o conjunto de permissões a um perfil no IAM. Para obter informações sobre conjuntos de permissões, consulte Conjuntos de Permissões no AWS IAM Identity CenterManual do Usuário.
- Permissões de usuários temporárias do IAM –- um usuário ou perfil do IAM pode assumir um perfil do IAM para obter temporariamente permissões diferentes para uma tarefa específica.

- Acesso entre contas –- você pode usar um perfil do IAM para permitir que alguém (uma entidade principal confiável) acesse recursos na sua conta de uma conta diferente. As funções são a forma primária de conceder acesso entre contas. No entanto, com alguns Serviços da AWS, você pode anexar uma política diretamente a um recurso (em vez de usar uma função como proxy). Para aprender a diferença entre funções e políticas baseadas em recurso para acesso entre contas, consulte Como as Funções do IAM Diferem das Políticas Baseadas em Recurso no Guia do Usuário do IAM.
- Acesso entre serviços –- alguns Serviços da AWS usam recursos em outros Serviços da AWS. Por exemplo, quando você faz uma chamada em um serviço, é comum que esse serviço execute aplicativos no Amazon EC2 ou armazene objetos no Amazon S3. Um serviço pode fazer isso usando as permissões de chamada da entidade principal, uma função de serviço ou uma função vinculada ao serviço.
  - Encaminhamento de sessões de acesso (FAS): qualquer pessoa que utilizar uma função ou usuário do IAM para realizar ações na AWS é considerada uma entidade principal. Ao usar alguns serviços, você pode executar uma ação que inicia outra ação em um serviço diferente. O recurso FAS utiliza as permissões da entidade principal que chama um AWS service (Serviço da AWS), combinadas às permissões do AWS service (Serviço da AWS) solicitante, para realizar solicitações para serviços downstream. As solicitações de FAS só são feitas quando um serviço recebe uma solicitação que exige interações com outros Serviços da AWS ou com recursos para serem concluídas. Nesse caso, você deve ter permissões para executar ambas as ações. Para obter detalhes da política ao fazer solicitações de FAS, consulte Encaminhamento de sessões de acesso.
  - Função de Serviço –- uma função de serviço é uma função do IAM que um serviço assume para realizar ações em seu nome. Um administrador do IAM pode criar, modificar e excluir um perfil de serviço do IAM. Para obter mais informações, consulte Criando um Perfil para Delegar Permissões a um AWS service (Serviço da AWS) no Guia do Usuário do IAM.
  - Função vinculada a serviço –- uma função vinculada a serviço é um tipo de função de serviço vinculada a um AWS service (Serviço da AWS). O serviço pode assumir o perfil de executar uma ação em seu nome. Funções vinculadas ao serviço aparecem em sua Conta da AWS e são de propriedade do serviço. Um administrador do IAM pode visualizar, mas não pode editar as permissões para funções vinculadas a serviço.
- Aplicações em execução no Amazon EC2: é possível usar uma função do IAM para gerenciar credenciais temporárias para aplicativos em execução em uma instância do EC2 e fazer solicitações da AWS CLI ou da AWS API. É preferível fazer isso a armazenar chaves de acesso na instância do EC2. Para atribuir uma função da AWS a uma instância do EC2 e disponibilizá-la

para todos os seus aplicativos, crie um perfil de instância que esteja anexado a ela. Um perfil de instância contém o perfil e permite que os programas em execução na instância do EC2 obtenham credenciais temporárias. Para mais informações, consulte Usar uma função do IAM para conceder permissões a aplicativos em execução nas instâncias do Amazon EC2 no Guia do usuário do IAM.

Para aprender se deseja usar perfis do IAM, consulte Quando Criar uma Função do IAM (em Vez de um Usuário) no Guia do Usuário do IAM.

Gerenciamento do acesso usando políticas

Você controla o acesso na AWS criando políticas e anexando-as a identidades ou recursos da AWS. Uma política é um objeto na AWS que, quando associado a uma identidade ou recurso, define suas permissões. A AWS avalia essas políticas quando uma entidade principal (usuário, usuário raiz ou sessão de função) faz uma solicitação. As permissões nas políticas determinam se a solicitação será permitida ou negada. A maioria das políticas são armazenadas na AWS como documentos JSON. Para obter mais informações sobre a estrutura e o conteúdo de documentos de políticas JSON, consulte Visão Geral das Políticas JSON no Guia do Usuário do IAM.

Os administradores podem usar as políticas JSON da AWS para especificar quem tem acesso a quê. Ou seja, qual entidade principal pode executar ações em quais recursos e sob quais condições.

Por padrão, usuários e funções não têm permissões. Para conceder aos usuários permissão para executar ações nos recursos de que precisam, um administrador do IAM pode criar políticas do IAM. O administrador pode então adicionar as políticas do IAM as funções e os usuários podem assumir as funções.

As políticas do IAM definem permissões para uma ação, independente do método usado para executar a operação. Por exemplo, suponha que você tenha uma política que permite a ação `iam:GetRole`. Um usuário com essa política pode obter informações de perfis do AWS Management Console, da AWS CLI ou da API da AWS.

Políticas baseadas em identidade

As políticas baseadas em identidade são documentos de políticas de permissões JSON que você pode anexar a uma identidade, como usuário, grupo de usuários ou perfil do IAM. Essas políticas controlam quais ações os usuários e funções podem realizar, em quais recursos e em quais condições. Para saber como criar uma política baseada em identidade, consulte Criar Políticas do IAM no Guia do Usuário do IAM.

As políticas baseadas em identidade também podem ser categorizadas como políticas em linha ou políticas gerenciadas. As políticas em linha são incorporadas diretamente a um único usuário, grupo ou função. As políticas gerenciadas são políticas independentes que podem ser anexadas a vários usuários, grupos e perfis na Conta da AWS. As políticas gerenciadas incluem políticas gerenciadas pela AWS e políticas gerenciadas pelo cliente. Para saber como selecionar entre uma política gerenciada ou uma política em linha, consulte Selecionar entre políticas gerenciadas e políticas em linha no Guia do usuário do IAM.

Políticas com base em recurso

Políticas baseadas em recurso são documentos de políticas JSON que você anexa a um recurso. São exemplos de políticas baseadas em recursos as políticas de confiança de função do IAM e as políticas do bucket do Amazon S3. Em serviços que suportem políticas baseadas em recursos, os administradores de serviço podem usá-las para controlar o acesso a um recurso específico. Para o recurso ao qual a política está anexada, a política define quais ações uma entidade principal especificada pode executar nesse recurso e em que condições. Você deve especificar uma entidade principal em uma política baseada em recursos. As entidades principais podem incluir contas, usuários, perfis, usuários federados ou Serviços da AWS.

Políticas baseadas em recursos são políticas em linha que estão localizadas nesse serviço. Não é possível usar as políticas gerenciadas da AWS do IAM em uma política baseada em recursos.

Listas de controle de acesso (ACLs)

As listas de controle de acesso (ACLs) controlam quais entidades principais (membros, usuários ou funções da conta) têm permissão para acessar um recurso. As ACLs são semelhantes as políticas baseadas em recursos, embora não usem o formato de documento de política JSON.

Amazon S3, AWS WAF e Amazon VPC são exemplos de serviços que oferecem suporte a ACLs. Saiba mais sobre ACLs em Configurações da lista de controle de acesso (ACL) no Guia do Desenvolvedor do Amazon Simple Storage Service.

Outros tipos de política

A AWS oferece suporte a tipos de política menos comuns. Esses tipos de política podem definir o máximo de permissões concedidas a você pelos tipos de política mais comuns.

- Limites de permissões –- um limite de permissões é um recurso avançado no qual você define o máximo de permissões que uma política baseada em identidade pode conceder a uma entidade do IAM (usuário ou perfil do IAM). É possível definir um limite de permissões para uma entidade.

As permissões resultantes são a interseção das políticas baseadas em identidade de uma entidade e dos seus limites de permissões. As políticas baseadas em recurso que especificam o usuário ou perfil no campo `Principal` não são limitadas pelo limite de permissões. Uma negação explícita em qualquer uma dessas políticas substitui a permissão. Para obter mais informações sobre limites de permissões, consulte Limites de Permissões para Entidades do IAM no Guia do Usuário do IAM.

- Políticas de controle de serviço (SCPs) — SCPs são políticas JSON que especificam as permissões máximas para uma organização ou unidade organizacional (UO) no AWS Organizations. O AWS Organizations é um serviço para agrupar e gerenciar centralmente várias Contas da AWS pertencentes à sua empresa. Se você habilitar todos os atributos em uma organização, poderá aplicar políticas de controle de serviço (SCPs) a qualquer uma ou a todas as contas. O SCP limita as permissões para entidades em contas-membro, inclusive cada Usuário raiz da conta da AWS. Para obter mais informações sobre o Organizações e SCPs, consulte Como os SCPs Funcionam no AWS OrganizationsManual do Usuário do.
- Políticas de sessão — são políticas avançadas que você transmite como um parâmetro quando cria de forma programática uma sessão temporária para uma função ou um usuário federado. As permissões da sessão resultante são a interseção das políticas baseadas em identidade do usuário ou do perfil e das políticas de sessão. As permissões também podem ser provenientes de uma política baseada em recurso. Uma negação explícita em qualquer uma dessas políticas substitui a permissão. Para obter mais informações, consulte Políticas de sessão no Guia do usuário do IAM.

Vários tipos de política

Quando vários tipos de política são aplicáveis a uma solicitação, é mais complicado compreender as permissões resultantes. Saiba como a AWS determina se deve permitir uma solicitação quando há vários tipos de política envolvidos consultando Lógica da Avaliação de Políticas no Guia do Usuário do IAM.

Como aAWS Serverless Application Repository funcionam com o IAM

Antes de usar o IAM para gerenciar o acesso aoAWS Serverless Application Repository, você deve entender quais recursos do IAM estão disponíveis para uso com oAWS Serverless Application Repository.

Para obter uma visão geral de como o IAM funciona, consulte Entendendo como o IAM funciona no Guia do usuário do IAM. Para obter uma visão de alto nível de como o serviçoAWS Serverless

Application Repository e outrosAWS serviços da funcionam com o IAM, consulte AWSServiços da que funcionam com o IAM no Guia do usuário do usuário do IAM.

Tópicos

- AWS Serverless Application Repository Políticas baseadas em identidade
- AWS Serverless Application RepositoryPolíticas de aplicativos
- Autorização baseada em tags do AWS Serverless Application Repository
- AWS Serverless Application RepositoryFunções do IAM

AWS Serverless Application Repository Políticas baseadas em identidade

Com as políticas baseadas em identidade do IAM, é possível especificar ações ou recursos permitidos ou negados, bem como as condições sob as quais as ações são permitidas ou negadas. O AWS Serverless Application Repository oferece suporte a ações, recursos e chaves de condição específicos. Para conhecer todos os elementos usados em uma política JSON, consulte Referência de elementos de política JSON do IAM no Manual do usuário do IAM.

A seguir, um exemplo de uma política de permissões.

```
{

    "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"
        }
```

```
    ]
}
```

A política tem duas instruções:

- A primeira instrução concede permissões para a ação `serverlessrepo:CreateApplication` do AWS Serverless Application Repository em todos os recursos do AWS Serverless Application Repository, conforme especificado pelo caractere curinga (*) como o valor `Resource`.
- A segunda declaração concede permissão para aAWS Serverless Application Repository ação`serverlessrepo:CreateApplicationVersion` em umAWS recurso usando o Amazon Resource Name (ARN) para umAWS Serverless Application Repository aplicativo. O aplicativo é especificado pelo valor `Resource`.

A política não especifica o elemento `Principal` porque, em uma política baseada em identidade, não se especifica o principal que obtém as permissões. Quando você anexar uma política um usuário, o usuário será o principal implícito. Quando você anexa uma política de permissão a um perfil do IAM, o principal identificado na política de confiança do perfil obtém as permissões.

Para ver uma tabela mostrando todas as operações daAWS Serverless Application Repository API e osAWS recursos aos quais elas se aplicam, consulte[AWS Serverless Application RepositoryPermissões da API do: Referência de ações e recursos](#).

Ações

Os administradores podem usar AWS as políticas JSON da para especificar quem tem acesso a quê. Ou seja, qual principal pode executar ações em quais recursos, e em que condições.

O elemento `Action` de uma política JSON descreve as ações que você pode usar para permitir ou negar acesso em uma política. As ações de política geralmente têm o mesmo nome que a operação de API da AWS associada. Existem algumas exceções, como ações somente de permissão, que não têm uma operação de API correspondente. Há também algumas operações que exigem várias ações em uma política. Essas ações adicionais são chamadas de ações dependentes.

Inclua ações em uma política para conceder permissões para executar a operação associada.

As ações de política no AWS Serverless Application Repository usam o seguinte prefixo antes da ação: `serverlessrepo:`. Por exemplo, para conceder permissão a alguém para executar uma instância do AWS Serverless Application Repository com a operação

da API `SearchApplications` do AWS Serverless Application Repository, inclua a ação `serverlessrepo:SearchApplications` na política da pessoa. As instruções de política devem incluir um elemento `Action` ou `NotAction`. O AWS Serverless Application Repository define seu próprio conjunto de ações que descrevem as tarefas que você pode executar com esse serviço.

Para especificar várias ações em uma única declaração, separe-as com vírgulas, conforme o seguinte:

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

Você também pode especificar várias ações usando caracteres curinga (*). Por exemplo, para especificar todas as ações que começam com a palavra `List`, inclua a seguinte ação:

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

Para ver uma lista de ações do AWS Serverless Application Repository, consulte Ações definidas pelo AWS Serverless Application Repository no Manual do usuário do IAM.

Recursos

Os administradores podem usar AWS as políticas JSON da para especificar quem tem acesso a quê. Ou seja, qual principal pode executar ações em quais recursos, e em que condições.

O elemento `Resource` de política JSON especifica o objeto ou os objetos aos quais a ação se aplica. As instruções devem incluir um elemento `Resource` ou um elemento `NotResource`. Como prática recomendada, especifique um recurso usando seu Nome do recurso da Amazon (ARN). Isso pode ser feito para ações que oferecem suporte a um tipo de recurso específico, conhecido como permissões em nível de recurso.

Para ações que não oferecem suporte a permissões em nível de recurso, como operações de listagem, use um curinga (*) para indicar que a instrução se aplica a todos os recursos.

```
"Resource": "*"
```

NoAWS Serverless Application Repository, oAWS recurso principal é umAWS Serverless Application Repository aplicativo. AWS Serverless Application Repositoryaplicativos têm nomes de recurso da Amazon (ARNs) exclusivos associados, conforme mostrado na tabela a seguir.

Tipo de recurso do AWS	Formato do nome de recurso da Amazon (ARN)
Aplicação	arn:*partition* :serverlessrepo:*region*:*account-id* :applications/*application-name*

Para obter mais informações sobre o formato de ARNs, consulte Nomes de recursos da Amazon (ARNs) e namespaces de serviços da AWS.

Veja a seguir um exemplo de política que concede permissões para `aserverlessrepo:ListApplications` ação em todos osAWS recursos. Na implementação atual, oAWS Serverless Application Repository não oferece suporte à identificação deAWS recursos específicos usando ARNs deAWS recursos (também chamados de permissões de recursos) para algumas das ações de API. Nesses casos, você deve especificar um caractere curinga (*).

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

Para obter uma tabela que mostra todas as ações de API doAWS Serverless Application Repository API do API do API do API do API do API Gateway eAWS os recursos aos quais elas se aplicam, consulteAWS Serverless Application RepositoryPermissões da API do: Referência de ações e recursos.

Chaves de condição

O AWS Serverless Application Repository não fornece nenhuma chave de condição específica ao serviço, mas oferece suporte ao uso de algumas chaves de condição globais. Para ver todas as chaves de condição globais da AWS, consulte Chaves de contexto de condição globais da AWS no Manual do usuário do IAM.

Exemplos

Para visualizar exemplos de políticas baseadas em identidade do AWS Serverless Application Repository, consulte Exemplos de políticas baseadas em identidade do AWS Serverless Application Repository.

AWS Serverless Application RepositoryPolíticas de aplicativos

As políticas do aplicativo determinam as ações que um diretor específico ou PrincipalOrg pode executar em umAWS Serverless Application Repository aplicativo.

Você pode adicionar permissões à política associada a um aplicativo do AWS Serverless Application Repository. As políticas de permissões anexadas aosAWS Serverless Application Repository aplicativos são chamadas de políticas de aplicativos. As políticas de aplicativos são extensões das políticas baseadas em recursos do IAM. O recurso principal é oAWS Serverless Application Repository aplicativo. Você pode usar políticas de aplicativo do AWS Serverless Application Repository para gerenciar permissões a implantação de aplicativos.

As políticas do aplicativo do AWS Serverless Application Repository são usadas principalmente por editores para conceder permissão aos consumidores para implantar os aplicativos e as operações relacionadas, como pesquisar e visualizar detalhes dos aplicativos. Os editores podem definir permissões de aplicativos às três seguintes categorias:

- Privado — Aplicativos que foram criados com a mesma conta e não foram compartilhados com nenhuma outra conta. Você tem permissão para implantar aplicativos que foram criados usando suaAWS conta.
- Compartilhado de forma privada — Aplicativos que o editor compartilhou explicitamente com um conjunto específico deAWS contas ouAWS Organizations. Você tem permissão para implantar aplicativos que foram compartilhados com suaAWS conta ouAWS organização.
- Compartilhado publicamente — Aplicativos que o editor compartilhou com todos. Você tem permissão para implantar aplicativos compartilhados publicamente.

Você pode conceder permissões usando osAWS CLI,AWS os SDKs ouAWS Management Console o.

Exemplos

Para ver exemplos de gerenciamento de políticas deAWS Serverless Application Repository aplicativos, consulteAWS Serverless Application RepositoryExemplos de políticas de aplicativos.

Autorização baseada em tags do AWS Serverless Application Repository

O AWS Serverless Application Repository não oferece suporte para controlar o acesso a recursos ou ações com base em tags.

AWS Serverless Application RepositoryFunções do IAM

Perfil do IAM é uma entidade dentro da sua conta da AWS que tem permissões específicas.

Usar credenciais temporárias com o AWS Serverless Application Repository

É possível usar credenciais temporárias para fazer login com federação, assumir uma função do IAM ou assumir uma função entre contas. As credenciais de segurança temporárias são obtidas chamando operações de API de API doAWS STS API do API do API do API do API do API do AssumeRoleGetFederationToken

O AWS Serverless Application Repository oferece suporte ao uso de credenciais temporárias.

Funções vinculadas ao serviço

O AWS Serverless Application Repository não oferece suporte a funções vinculadas ao serviço.

Funções de serviço

O AWS Serverless Application Repository não oferece suporte a funções de serviço.

Exemplos de políticas baseadas em identidade do AWS Serverless Application Repository

Por padrão, os usuários e as funções do IAM não têm permissão para criar ou modificar recursos do AWS Serverless Application Repository. Eles também não podem executar tarefas usando o AWS Management Console, a AWS CLI ou uma API da AWS. Um administrador do IAM deve criar políticas do IAM que concedam aos usuários e perfis permissão para executarem operações de API específicas nos recursos especificados de que precisam. O administrador deve anexar essas políticas aos usuários ou grupos do IAM que exigem essas permissões.

Para saber como criar uma política baseada em identidade do IAM usando esses exemplos de documentos de política JSON, consulte Criar políticas na guia JSON no Guia do usuário do IAM.

Tópicos

- Melhores práticas de políticas
- Usar o console AWS Serverless Application Repository
- Permitir que os usuários visualizem suas próprias permissões
- Exemplos de política gerenciada pelo cliente

Melhores práticas de políticas

As políticas baseadas em identidade são muito eficientes. Elas determinam se alguém pode criar, acessar ou excluir recursos do AWS Serverless Application Repository em sua conta. Essas ações podem incorrer em custos para sua conta da AWS. Ao criar ou editar políticas baseadas em identidade, siga estas diretrizes e recomendações:

- Conceder privilégio mínimo: ao criar políticas personalizadas, conceda apenas as permissões necessárias para executar uma tarefa. Comece com um conjunto mínimo de permissões e conceda permissões adicionais conforme necessário. Fazer isso é mais seguro do que começar com permissões que são muito lenientes e tentar restringi-las superiormente. Para obter mais informações, consulte Conceder privilégio mínimo no Guia do usuário do IAM.
- Habilitar MFA para operações confidenciais: para aumentar a segurança, exija que os usuários do IAM usem Multi-Factor Authentication (MFA) para acessar recursos ou operações de API confidenciais. Para obter mais informações, consulte Uso da autenticação multifator (MFA) na AWS no Manual do ujsuário do IAM.
- Usar condições de política para segurança adicional: na medida do possível, defina as condições sob as quais suas políticas baseadas em identidade permitem o acesso a um recurso. Por exemplo, você pode gravar condições para especificar um intervalo de endereços IP permitidos do qual a solicitação deve partir. Você também pode escrever condições para permitir somente solicitações em uma data especificada ou período ou para exigir o uso de SSL ou MFA. Para obter mais informações, consulte Elementos de política JSON do IAM: condição no Guia do usuário do IAM.

Usar o console AWS Serverless Application Repository

O console do AWS Serverless Application Repository fornece um ambiente integrado para você descobrir e gerenciar aplicativos do AWS Serverless Application Repository. O console fornece recursos e fluxos de trabalho que, muitas vezes, exigem permissões para gerenciar um aplicativo do AWS Serverless Application Repository, além das permissões específicas à API documentadas em AWS Serverless Application RepositoryPermissões da API do: Referência de ações e recursos

Para obter mais informações sobre permissões necessárias para usar o console do AWS Serverless Application Repository, consulte Exemplos de política gerenciada pelo cliente.

Permitir que os usuários visualizem suas próprias permissões

Este exemplo mostra como você pode criar uma política que permite que os usuários do IAM visualizem as políticas gerenciadas e em linha anexadas a sua identidade de usuário. Essa política inclui permissões para concluir essa ação no console ou de forma programática usando a AWS CLI ou a API da AWS.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ViewOwnUserInfo",
            "Effect": "Allow",
            "Action": [
                "iam:GetUserPolicy",
                "iam:ListGroupsForUser",
                "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": "*"
        }
    ]
}
```

Exemplos de política gerenciada pelo cliente

Os exemplos desta seção apresentam um grupo de políticas de amostra que você pode anexar a um usuário. Se você não estiver familiarizado com a criação de políticas, recomendamos primeiro criar um usuário do IAM na conta e anexar as políticas ao usuário sequencialmente. Você também pode usar esses exemplos para criar uma única política personalizada que inclui permissões para executar várias ações e depois anexá-la ao usuário.

Para obter mais informações sobre como anexar políticas aos usuários, consulte Adicionar permissões a um usuário no Guia do usuário do IAM.

Exemplos

- Exemplo 1 do editor: Permitir que um editor liste os aplicativos
- Exemplo 2 do editor: Permitir que um editor visualize os detalhes de um aplicativo ou de uma versão do aplicativo
- Exemplo 3 do editor: Permitir que um editor crie um aplicativo ou uma versão do aplicativo
- Exemplo 4 do editor: Permitir que um editor crie uma política de aplicativo para compartilhar aplicativos com outros usuários
- Exemplo 1 de consumidor: Permitir que um consumidor pesquise por aplicativos
- Exemplo 2 de consumidor: Permitir que um consumidor visualize detalhes de um aplicativo
- Exemplo 3 de consumidor: Permitir que um consumidor implante um aplicativo
- Exemplo 4 do consumidor: negar acesso aos ativos de implantação
- Exemplo 5 do consumidor: impedir que um consumidor pesquise e implante aplicativos públicos

Exemplo 1 do editor: Permitir que um editor liste os aplicativos

Um usuário do IAM em sua conta deve ter permissões para a operação `serverlessrepo:ListApplications` antes que o usuário possa ver qualquer coisa no console. Quando você concede essas permissões, o console pode mostrar a lista deAWS Serverless Application Repository aplicativos naAWS conta criada naAWS região específica à qual o usuário pertence.

```
{
    "Version": "2012-10-17",
    "Statement": [
```

```
        {
            "Sid": "ListExistingApplications",
            "Effect": "Allow",
            "Action": [
                "serverlessrepo:ListApplications"
            ],
            "Resource": "*"
        }
    ]
}
```

Exemplo 2 do editor: Permitir que um editor visualize os detalhes de um aplicativo ou de uma versão do aplicativo

Um usuário pode selecionar um aplicativo do AWS Serverless Application Repository e visualizar os detalhes do aplicativo. Esses detalhes incluem autor, descrição, versões e outras informações de configuração. Para isso, o usuário precisa de permissões para as operações `serverlessrepo:ListApplicationVersions` e `serverlessrepo:GetApplication` da API para o AWS Serverless Application Repository.

No exemplo a seguir, essas permissões são concedidas ao aplicativo específico cujo Amazon Resource Name (ARN - Nome de recurso da Amazon) é especificado como o valor de `Resource`.

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

Exemplo 3 do editor: Permitir que um editor crie um aplicativo ou uma versão do aplicativo

Para conceder permissões a um usuário para criar aplicativos do AWS Serverless Application Repository, é necessário conceder permissões para as operações `serverlessrepo:CreateApplication` e `serverlessrepo:CreateApplicationVersions`, conforme exibido na política a seguir.

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

Exemplo 4 do editor: Permitir que um editor crie uma política de aplicativo para compartilhar aplicativos com outros usuários

Para que os usuários compartilhem aplicativos com outros usuários, você deve conceder a eles permissões para criar políticas de aplicativo, conforme exibido na política a seguir.

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

```
    ]
}
```

Exemplo 1 de consumidor: Permitir que um consumidor pesquise por aplicativos

Para que os consumidores pesquisem por aplicativos, você deve conceder as seguintes permissões.

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

Exemplo 2 de consumidor: Permitir que um consumidor visualize detalhes de um aplicativo

Um usuário pode selecionar um aplicativo do AWS Serverless Application Repository e visualizar detalhes do aplicativo, como autor, descrição, versões e outras informações de configuração. Para isso, o usuário deve ter permissões para as seguintes operações do AWS Serverless Application Repository.

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

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

Exemplo 3 de consumidor: Permitir que um consumidor implante um aplicativo

Para que os clientes implantem aplicativos, você deve conceder a eles permissões para executar várias operações. A política a seguir fornece as permissões necessárias aos clientes.

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

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

> **Note**
>
> A implantação de um aplicativo pode exigir permissões para usarAWS recursos adicionais. Como oAWS Serverless Application Repository usa o mesmo mecanismo de implantação subjacente queAWS CloudFormation, consulte Controlando o acesso comAWS Identity and Access Management para obter mais informações. Para obter ajuda com problemas de implantação relacionados a permissões, consulte Solução de problemas: permissões do IAM insuficientes.

Exemplo 4 do consumidor: negar acesso aos ativos de implantação

Quando um aplicativo é compartilhado de forma privada com umaAWS conta, por padrão, todos os usuários dessa conta podem acessar os ativos de implantação de todos os outros usuários na mesma conta. A política a seguir impede que os usuários de uma conta acessem os ativos de implantação, que são armazenados no bucket do Amazon S3 paraAWS Serverless Application Repository o.

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

Exemplo 5 do consumidor: impedir que um consumidor pesquise e implante aplicativos públicos

É possível impedir que os usuários executem determinadas ações em aplicativos.

A política a seguir se aplica a aplicativos públicos especificando `serverlessrepo:applicationType` como `public`. Ela impede que os usuários realizem uma série de ações especificando `Effect` como `Deny`. Para obter mais informações sobre as chaves de condição disponíveis para o AWS Serverless Application Repository, consulte [Ações, recursos e chaves de condição do 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**
>
> Essa declaração de política também pode ser usada como uma Política de Controle de Serviços e aplicada a umaAWS organização. Para obter mais informações sobre políticas de controle de serviço, consulte Políticas de controle de serviço no GuiaAWS Organizations do usuário.

AWS Serverless Application RepositoryExemplos de políticas de aplicativos

As políticas de permissões anexadas aosAWS Serverless Application Repository aplicativos são chamadas de políticas de aplicativos. As políticas do aplicativo determinam as ações que um diretor específico ou PrincipalOrg pode executar em umAWS Serverless Application Repository aplicativo.

UmAWS Serverless Application Repository aplicativo é o principalAWS recurso noAWS Serverless Application Repository. AWS Serverless Application Repositoryas políticas de aplicativos são usadas principalmente pelos editores para conceder permissão aos consumidores para implantar seus aplicativos e operações relacionadas, como pesquisar e visualizar detalhes desses aplicativos.

Os editores podem definir permissões de aplicativos às três seguintes categorias:

- Privado — Aplicativos que foram criados com a mesma conta e não foram compartilhados com nenhuma outra conta. Somente consumidores que compartilham suaAWS conta têm permissão para implantar aplicativos privados.

- Compartilhado de forma privada — Aplicativos que o editor compartilhou explicitamente com um conjunto específico deAWS contas ou comAWS contas em umaAWS organização. Os consumidores têm permissão para implantar aplicativos que foram compartilhados com suaAWS conta ouAWS organização. Para obter mais informações sobreAWS organizações, consulte o GuiaAWS Organizations do usuário.
- Compartilhado publicamente — Aplicativos que o editor compartilhou com todos. Todos os consumidores têm permissão para implantar um aplicativo compartilhado publicamente.

> **Note**
>
> Para aplicativos compartilhados de forma privada, oAWS Serverless Application Repository único compatível com AWScontas como diretores. Os editores podem conceder ou negar todos os usuários de umaAWS conta como um único grupo para umAWS Serverless Application Repository aplicativo. Os editores não podem conceder ou negar usuários individuais de umaAWS conta a umAWS Serverless Application Repository aplicativo.

Para obter instruções sobre como configurar permissões para aplicativos usando o AWS Management Console, consulte Compartilhar um aplicativo.

Para obter instruções sobre como configurar permissões de aplicativos usando a AWS CLI e exemplos, consulte as seguintes seções.

Permissões de aplicativos (AWS CLIeAWS SDKs)

Ao usar oAWS CLI ou osAWS SDKs para definir permissões para umAWS Serverless Application Repository aplicativo, você pode especificar as seguintes ações:

Ação	Descrição
GetApplication	Concede permissão para visualizar informações sobre o aplicativo.
CreateCloudFormationChangeSet	Concede permissão para o aplicativo ser implantado. Observação: essa ação não concede nenhuma outra permissão além da permissão de implantação.

Ação	Descrição
CreateCloudFormationTemplate	Concede permissão para criar um modelo AWS CloudFormation para o aplicativo.
ListApplicationVersions	Concede permissão para listar as versões do aplicativo.
ListApplicationDependencies	Concede permissão para relacionar os aplicativos que estão aninhados dentro do aplicativo.
SearchApplications	Concede permissão para o aplicativo ser pesquisado.
Implante	Essa ação habilita todas as ações listadas anteriormente na tabela. Ou seja, concede permissão para o aplicativo ser visualizado, implantado, as versões serem listadas e o aplicativo ser pesquisado.

Exemplos de política de usuário

Os exemplos a seguir mostram como conceder permissões usando a AWS CLI. Para obter informações sobre como conceder permissões usando o AWS Management Console, consulte Compartilhar um aplicativo.

Todos os exemplos nesta seção usam esses comandos da AWS CLI para gerenciar políticas de permissões associadas aos aplicativos do AWS Serverless Application Repository:

- put-application-policy
- get-application-policy

Tópicos

- Exemplo 1: compartilhar um aplicativo com outra conta
- Exemplo 2: Compartilhar um aplicativo publicamente
- Exemplo 3: Tornar um aplicativo privado
- Exemplo 4: Especificação de várias contas e permissões
- Exemplo 5: compartilhar um aplicativo com todas as contas em umaAWS organização
- Exemplo 6: Compartilhando um aplicativo com algumas contas em umaAWS organização
- Exemplo 7: recuperar uma política de aplicativo

- Exemplo 8: permitir que o aplicativo seja aninhado por contas específicas

Exemplo 1: compartilhar um aplicativo com outra conta

Para compartilhar um aplicativo com outra conta específica, mas evitar que ele seja compartilhado com outras pessoas, você especifica o ID daAWS conta com a qual deseja compartilhar como principal. Isso também é conhecido como configurar o aplicativo como compartilhado de forma privada. Para fazer isso, execute o seguinte comando da AWS CLI.

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

> **Note**
>
> Aplicativos compartilhados de forma privada só podem ser usados na mesmaAWS região em que o aplicativo foi criado.

Exemplo 2: Compartilhar um aplicativo publicamente

Para tornar um aplicativo público, você o compartilha com todos os usuários especificando"*" como a entidade principal, como no exemplo a seguir. Os aplicativos compartilhados publicamente ficam disponíveis em todas as regiões.

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

> **Note**
>
> Para compartilhar um aplicativo publicamente, ele deve ter as propriedades `LicenseUrl` e `SemanticVersion` definidas.

Exemplo 3: Tornar um aplicativo privado

Você pode tornar um aplicativo privado, para que ele não seja compartilhado com ninguém e só possa ser implantado pelaAWS conta que o possui. Para fazer isso, você elimina os princípios e ações da política, o que também remove as permissões de outras contas da suaAWS organização para implantar seu aplicativo.

```
aws serverlessrepo put-application-policy \
--region region \
--application-id application-arn \
--statements '[]'
```

> **Note**
>
> Aplicativos privados só podem ser usados na mesmaAWS região em que o aplicativo foi criado.

Exemplo 4: Especificação de várias contas e permissões

Você pode conceder várias permissões e concedê-las a mais de umaAWS conta ao mesmo tempo. Para fazer isso, especifique listas como o principal e como ações, conforme exibido no exemplo a seguir.

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

Exemplo 5: compartilhar um aplicativo com todas as contas em umaAWS organização

As permissões podem ser concedidas a todos os usuários de umaAWS organização. É possível fazer isso especificando o ID da organização, como no exemplo a seguir.

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

Para obter mais informações sobreAWS organizações, consulte o GuiaAWS Organizations do usuário.

> **Note**
>
> Você só pode especificar aAWS organização da qual suaAWS conta é membro. Se você tentar especificar umaAWS organização da qual não é membro, ocorrerá um erro.
> Para compartilhar seu aplicativo com suaAWS organização, você deve incluir permissão para a`UnshareApplication` ação, caso o compartilhamento precise ser revogado no future.

Exemplo 6: Compartilhando um aplicativo com algumas contas em umaAWS organização

As permissões podem ser concedidas a contas específicas dentro de umaAWS organização. Você faz isso especificando uma lista deAWS contas como principal e o ID da sua organização, como no exemplo a seguir.

```
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**
>
> Você só pode especificar aAWS organização da qual suaAWS conta é membro. Se você tentar especificar umaAWS organização da qual não é membro, ocorrerá um erro.
> Para compartilhar seu aplicativo com suaAWS organização, você deve incluir permissão para a`UnshareApplication` ação, caso o compartilhamento precise ser revogado no future.

Exemplo 7: recuperar uma política de aplicativo

Para visualizar a política atual de um aplicativo, por exemplo, para ver se ele está sendo compartilhado no momento, use o comando `get-application-policy`, como no exemplo a seguir.

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

```
--application-id application-arn
```

Exemplo 8: permitir que o aplicativo seja aninhado por contas específicas

Aplicativos públicos podem ser aninhados por qualquer pessoa. Caso você queira permitir que seu aplicativo seja aninhado por contas específicas, você deve ajustar as permissões mínimas abaixo, conforme exibido no exemplo a seguir.

```
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 RepositoryPermissões da API do: Referência de ações e recursos

Ao configurar o controle de acesso e elaborar políticas de permissões que você pode associar a uma identidade do IAM (políticas baseadas em identidade), use a tabela a seguir como referência. OcadaAWS Serverless Application Repositoryoperação de API do, as ações correspondentes às quais você pode conceder permissões para executar a ação e oAWSO recurso da para você pode conceder as permissões. Você especifica as ações no campo `Action` da política e o valor do recurso no campo `Resource` da política.

Para especificar uma ação, use o prefixo `serverlessrepo:` seguido do nome da operação da API (por exemplo, `serverlessrepo:ListApplications`).

Operação	URI	Método	AWSRecursos (ARNs)
operação: ListApplications Permissões obrigatórias:serverlessrepo:ListApplications	/applications	GET	*
operação: CreateApplication	/applications	POST	*

Operação	URI	Método	AWSRecursos (ARNs)
Permissões obrigatórias:serverlessrepo:CreateApplication			
operação: GetApplication Permissões obrigatórias:serverlessrepo:GetApplication	/applications/*id-do-aplicativo*	GET	arn:aws:serverlessrepo:*região*:*id-da-conta* :applications/*nome-do-aplicativo*
operação: DeleteApplication Permissões obrigatórias:serverlessrepo:DeleteApplication	/applications/*id-do-aplicativo*	DELETE	arn:aws:serverlessrepo:*região*:*id-da-conta* :applications/*nome-do-aplicativo*
operação: UpdateApplication Permissões obrigatórias:serverlessrepo:UpdateApplication	/applications/*id-do-aplicativo*	PATCH	arn:aws:serverlessrepo:*região*:*id-da-conta* :applications/*nome-do-aplicativo*
operação: CreateCloudFormationChangeDefina Permissões obrigatórias:serverlessrepo:CreateCloudFormationChangeDefina	/applications/*id-do-aplicativo* /changesets	POST	arn:aws:serverlessrepo:*região*:*id-da-conta* :applications/*nome-do-aplicativo*

Operação	URI	Método	AWSRecursos (ARNs)
operação: GetApplicationPolítica Permissões obrigatórias:serverlessrepo:GetApplicationPolítica	/applications/*id-do-aplicativo* /policy	GET	arn:aws:serverlessrepo:*região*:*id-da-conta* :applications/*nome-do-aplicativo*
operação: PutApplicationPolítica Permissões obrigatórias:serverlessrepo:PutApplicationPolítica	/applications/*id-do-aplicativo* /policy	PUT	arn:aws:serverlessrepo:*região*:*id-da-conta* :applications/*nome-do-aplicativo*
operação: ListApplicationVersões do : Permissões obrigatórias:serverlessrepo:ListApplicationVersões do :	/applications/*id-do-aplicativo* /versions	GET	arn:aws:serverlessrepo:*região*:*id-da-conta* :applications/*nome-do-aplicativo*
operação: CreateApplicationVersão Permissões obrigatórias:serverlessrepo:CreateApplicationVersão	/applications/*id-do-aplicativo* /versions/*versão-semântica*	PUT	arn:aws:serverlessrepo:*região*:*id-da-conta* :applications/*nome-do-aplicativo*

Operação	URI	Método	AWSRecursos (ARNs)
operação: ListApplicationDependências do Permissões obrigatórias:serverlessrepo:ListApplicationDependências do	/aplicativos/*id-do-aplicativo* /dependências	GET	arn:aws:serverlessrepo:*região*:*id-da-conta* :applications/*nome-do-aplicativo*
operação: SearchApplications Permissões obrigatórias:serverlessrepo:SearchApplications	n/a	n/a	*

Solução de problemas de identidade e acesso AWS Serverless Application Repository

Use as seguintes informações para ajudar a diagnosticar e corrigir problemas comuns que podem ser encontrados ao trabalhar com oAWS Serverless Application Repository e o IAM.

Tópicos

- Não tenho autorização para executar uma ação no AWS Serverless Application Repository
- Não estou autorizado a realizar o IAM:PassRole
- Sou administrador e desejo conceder acesso ao AWS Serverless Application Repository para outros usuários
- Quero permitir que as pessoas fora da minha conta da AWS acessem meus recursos do AWS Serverless Application Repository

Não tenho autorização para executar uma ação no AWS Serverless Application Repository

Se o AWS Management Console informar que você não está autorizado a executar uma ação, você deverá entrar em contato com o administrador para obter assistência. O administrador é a pessoa que forneceu a você o seu nome de usuário e senha.

O exemplo de erro a seguir ocorre quando o usuário domateojackson IAM tenta usar o console para visualizar detalhes sobre um aplicativo, mas não temserverlessrepo:*GetApplication* as permissões.

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

Nesse caso, Mateo pede ao administrador para atualizar suas políticas a fim de obter acesso ao recurso *my-example-application* usando a operação serverlessrepo:*GetApplication*.

Não estou autorizado a realizar o IAM:PassRole

Se você receber uma mensagem de erro informando que não tem autorização para executar a ação iam:PassRole, as suas políticas deverão ser atualizadas para permitir que você passe um perfil para o AWS Serverless Application Repository.

Alguns Serviços da AWS permitem que você transmita um perfil existente para o serviço, em vez de criar um perfil de serviço ou um perfil vinculado ao serviço. Para fazer isso, um usuário deve ter permissões para passar o perfil para o serviço.

O erro de exemplo a seguir ocorre quando uma usuária do IAM chamada marymajor tenta usar o console para executar uma ação no AWS Serverless Application Repository. No entanto, a ação exige que o serviço tenha permissões concedidas por um perfil de serviço. Mary não tem permissões para passar a função para o serviço.

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

Nesse caso, as políticas de Mary devem ser atualizadas para permitir que ela realize a ação iam:PassRole.

Se você precisar de ajuda, entre em contato com seu administrador da AWS. Seu administrador é a pessoa que forneceu a você suas credenciais de login.

Sou administrador e desejo conceder acesso ao AWS Serverless Application Repository para outros usuários

Para permitir que outros usuários acessem o AWS Serverless Application Repository, crie uma entidade do IAM (usuário ou função) para a pessoa ou a aplicação que precisa do acesso. Elas usarão as credenciais dessa entidade para acessar a AWS. Você deve anexar uma política à entidade que concede a eles as permissões corretas no AWS Serverless Application Repository.

Para começar a usar imediatamente, consulte Criar os primeiros usuário e grupo delegados do IAM no Guia do usuário do IAM.

Quero permitir que as pessoas fora da minha conta da AWS acessem meus recursos do AWS Serverless Application Repository

Você pode criar uma função que os usuários de outras contas ou pessoas fora da sua organização podem usar para acessar seus recursos. Você pode especificar quem é confiável para assumir a função. Para serviços que oferecem suporte a políticas baseadas em recursos ou listas de controle de acesso (ACLs), você pode usar essas políticas para conceder às pessoas acesso aos seus recursos.

Para saber mais, consulte o seguinte:

- Para saber se o AWS Serverless Application Repository oferece suporte a esses recursos, consulte Como aAWS Serverless Application Repository funcionam com o IAM.
- Para saber como conceder acesso a seus recursos em todas as Contas da AWS pertencentes a você, consulte Fornecimento de acesso a um usuário do IAM em outra Conta da AWS pertencente a você no Guia de usuário do IAM.
- Para saber como conceder acesso a seus recursos para terceiros Contas da AWS, consulte Fornecimento de acesso a Contas da AWS pertencentes a terceiros no Guia do usuário do IAM.
- Para saber como conceder acesso por meio da federação de identidades, consulte Conceder acesso a usuários autenticados externamente (federação de identidades) no Guia do usuário do IAM.
- Para saber a diferença entre usar perfis e políticas baseadas em recursos para acesso entre contas, consulte Como os perfis do IAM diferem de políticas baseadas em recursos no Guia do usuário do IAM.

Registro em log e monitoramento no AWS Serverless Application Repository

O monitoramento é uma parte importante da manutenção da confiabilidade, disponibilidade e performance das suas soluções da AWS. Você deve coletar dados de monitoramento de todas as partes da solução da AWS para depurar uma falha de vários pontos com mais facilidade, caso ocorra. A AWS fornece várias ferramentas para monitorar seus recursos do AWS Serverless Application Repository e responder a incidentes em potencial, tais como:

Logs do AWS CloudTrail

OAWS Serverless Application RepositoryO é integrado aAWS CloudTrail, um serviço que fornece um registro de ações executadas por um usuário, uma função ou umAWSserviçoServiço naAWS Serverless Application Repository. CloudTrailO captura todas as chamadas à API doAWS Serverless Application Repositorycomo eventos.

Tópicos

- Registro em log de chamadas à API do AWS Serverless Application Repository com o AWS CloudTrail

Registro em log de chamadas à API do AWS Serverless Application Repository com o AWS CloudTrail

AWS Serverless Application RepositoryO é integrado aoAWS CloudTrail, que é um serviço que fornece um registro das ações realizadas por um usuário, uma função ou umAWSserviçoServiço noAWS Serverless Application Repository. CloudTrailO captura todas as chamadas à API doAWS Serverless Application Repositorycomo eventos. As chamadas capturadas incluem as chamadas do console do AWS Serverless Application Repository e as chamadas de código para as operações da API do AWS Serverless Application Repository.

Se você criar uma trilha, poderá habilitar a entrega contínua doCloudTraileventos para um bucket do Amazon S3, incluindo eventos para oAWS Serverless Application Repository. Se não configurar uma trilha, você ainda poderá visualizar os eventos mais recentes no console do CloudTrail em Event history.

Usando as informações coletadas pelo CloudTrail, é possível determinar a solicitação que foi feita ao AWS Serverless Application Repository. Também é possível determinar o endereço IP do qual a solicitação foi feita, quem fez a solicitação, quando ela foi feita, e detalhes adicionais.

Para saber mais sobre CloudTrail, consulte o Guia do usuário de AWS CloudTrail.

Informações sobre o AWS Serverless Application Repository no CloudTrail

O CloudTrail está habilitado na sua conta da AWS ao criá-la. Quando ocorre uma atividade no AWS Serverless Application Repository, ela é registrada em um evento do CloudTrail com outros eventos de serviços da AWS em Event history (Histórico de eventos). Você pode visualizar, pesquisar e baixar eventos recentes em sua conta da AWS. Para obter mais informações, consulte Visualizar eventos com o histórico de eventos do CloudTrail.

Para obter um registro contínuo de eventos na conta da AWS, incluindo eventos do AWS Serverless Application Repository, crie uma trilha. UMAtrilhaativaCloudTrailpara enviar arquivos de log para um bucket do Amazon S3. Por padrão, quando você cria uma trilha no console, ela é aplicada a todas as regiões da AWS. A trilha registra em log eventos de todas as regiões da AWS na partição da AWS e entrega os arquivos de log para o bucket do Amazon S3 especificado por você. Além disso, é possível configurar outros serviços da AWS para analisar mais profundamente e agir sobre os dados de evento coletados nos logs do CloudTrail. Para obter mais informações, consulte as informações a seguir:

- Visão geral da criação de uma trilha
- CloudTrail Serviços compatíveis e integrações do
- Configuração de notificações do Amazon SNS para o CloudTrail
- Receber arquivos de log do CloudTrail de várias regiões e receber arquivos de log do CloudTrail de várias contas

Todas as ações do AWS Serverless Application Repository são registradas em log pelo CloudTrail e documentadas na página Recursos do AWS Serverless Application Repository. Por exemplo, as chamadas para as operações `CreateApplication`, `UpdateApplications` e `ListApplications` geram entradas nos arquivos de log do CloudTrail.

Cada entrada de log ou evento contém informações sobre quem gerou a solicitação. As informações de identidade ajudam a determinar:

- Se a solicitação foi feita com credenciais de usuário raiz ou do AWS Identity and Access Management (IAM).
- Se a solicitação foi feita com credenciais de segurança temporárias de uma função ou de um usuário federado.
- Se a solicitação foi feita por outro serviço da AWS.

Para obter mais informações, consulte Elemento userIdentity do CloudTrail.

Noções básicas das entradas dos arquivos de log do AWS Serverless Application Repository

Uma trilha é uma configuração que permite a entrega de eventos como arquivos de log a um bucket do Amazon S3 especificado. Os arquivos de log do CloudTrail contêm uma ou mais entradas de log. Um evento representa uma única solicitação de qualquer fonte e inclui informações sobre a ação solicitada, a data e a hora da ação, os parâmetros de solicitação e assim por diante. Os arquivos de log do CloudTrail não são um rastreamento de pilha ordenada de chamadas de API pública. Dessa forma, eles não são exibidos em uma ordem específica.

O exemplo a seguir mostra uma entrada de log do CloudTrail que demonstra a ação `CreateApplication`.

```
{
    "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"
}
```

Validação de conformidade do AWS Serverless Application Repository

Auditores externos avaliam a segurança e a conformidade do AWS Serverless Application Repository como parte de vários programas de conformidade da AWS. Isso inclui SOC, PCI, FedRAMP e outros.

Para uma lista deAWSserviços da no escopo de programas de conformidade específicos, consulteAWSServiços da no escopo pelo programa de conformidade. Para obter informações gerais, consulte Programas de conformidade da AWS.

É possível baixar os relatórios de auditoria de terceiros usando o AWS Artifact. Para obter mais informações, consulte Download de relatórios no AWS Artifact.

Sua responsabilidade com relação à conformidade ao usar o AWS Serverless Application Repository é determinada pela confidencialidade dos dados, pelos objetivos de conformidade da empresa e pelos regulamentos e leis aplicáveis. A AWS fornece os seguintes recursos para ajudar com a conformidade:

- Guias de início rápido de segurança e conformidade— esses guias de implantação abordam as considerações de arquitetura e fornecem etapas para implantação de ambientes de linha de base focados em conformidade e segurança naAWS.
- Recursos de compatibilidade da AWS – Esta coleção de guias e pastas de trabalho pode ser aplicada ao seu setor e local.
- AWS Config: esse serviço da AWS avalia até que ponto suas configurações de recursos atendem adequadamente às práticas internas e às diretrizes e regulamentações do setor.

- AWS Security Hub: esse serviço da AWS fornece uma visão abrangente do estado de sua segurança na AWS que ajuda você a conferir sua conformidade com padrões e práticas recomendadas de segurança do setor.

Resiliência no AWS Serverless Application Repository

A infraestrutura global da AWS é criada com base em regiões da AWS e zonas de disponibilidade da AWS. As regiões fornecem várias zonas de disponibilidade separadas e isoladas fisicamente, as quais são conectadas com baixa latência, altas taxas de transferência e redes altamente redundantes. Com as zonas de disponibilidade, você pode projetar e operar aplicativos e bancos de dados que executam o failover automaticamente entre as zonas de disponibilidade sem interrupção. As zonas de disponibilidade são mais altamente disponíveis, tolerantes a falhas e escaláveis que uma ou várias infraestruturas de data center tradicionais.

Para obter mais informações sobre regiões e zonas de disponibilidade da AWS, consulte Infraestrutura global da AWS.

Segurança da infraestrutura no AWS Serverless Application Repository

Como um serviço gerenciado,AWS Serverless Application Repositoryé protegido porAWSsegurança de rede global. Para obter informações sobre serviços de segurança da AWS e como a AWS protege a infraestrutura, consulte Segurança na Nuvem AWS. Para projetar seu ambiente da AWS usando as práticas recomendadas de segurança de infraestrutura, consulte Proteção de infraestrutura em Pilar segurança: AWS Well-Architected Framework.

Você usa chamadas de API publicadas pela AWS para acessar o AWS Serverless Application Repository por meio da rede. Os clientes devem oferecer suporte para:

- Transport Layer Security (TLS). Exigimos TLS 1.2 e recomendamos TLS 1.3.
- Conjuntos de criptografia com perfect forward secrecy (PFS) como DHE (Ephemeral Diffie-Hellman) ou ECDHE (Ephemeral Elliptic Curve Diffie-Hellman). A maioria dos sistemas modernos, como Java 7 e versões posteriores, comporta esses modos.

Além disso, as solicitações devem ser assinadas usando um ID da chave de acesso e uma chave de acesso secreta associada a uma entidade principal do IAM. Ou você pode usar o AWS

Security Token Service (AWS STS) para gerar credenciais de segurança temporárias para assinar solicitações.

Cotas do AWS Serverless Application Repository

OAWS Serverless Application RepositoryO tem uma cota para o número de aplicativos públicos queAWSconta pode ter em cadaAWSRegião : Essa cota é aplicada por região e pode ser aumentada. Para solicitar um aumento, use o Console do centro de suporte.

Recurso	Cota padrão
Aplicativos públicos (porAWSPor conta daAWSRegião)	100

As cotas a seguir se aplicam ao armazenamento disponível para pacotes de códigos e políticas de aplicativos. Não é possível alterar essas cotas.

Recurso	Quota
Armazenamento gratuito do Amazon S3 para pacotes de código (porAWSPor conta daAWSRegião)	5 GB
Duração da política do aplicativo	6.144 caracteres

Solucionar problemas do AWS Serverless Application Repository

Ao usar o AWS Serverless Application Repository, você pode encontrar problemas ao criar, atualizar ou excluir seus aplicativos. Use esta seção para ajudar a solucionar problemas comuns que podem ocorrer. Também é possível pesquisar respostas e postar perguntas no Fóruns do AWS Serverless Application Repository.

> **Note**
>
> Os aplicativos no AWS Serverless Application Repository são implantados usando o AWS CloudFormation. Para obter informações sobre a solução de problemas do AWS CloudFormation consulte o AWS CloudFormation Guia de solução de problemas.

Tópicos

- Não é possível tornar um aplicativo público
- Uma cota foi excedida
- Um arquivo Readme atualizado não aparece imediatamente
- Você não pode implantar um aplicativo devido a permissões do IAM insuficientes
- Não é possível implantar o mesmo aplicativo duas vezes
- Por que meu aplicativo não está publicamente disponível?
- Como entrar em contato com o Support

Não é possível tornar um aplicativo público

Se não for possível tornar seu aplicativo público, pode haver um arquivo de licença ausente para o seu aplicativo aprovado pela Open Source Initiative (OSI).

Para tornar seu aplicativo público, você precisa de um arquivo de licença aprovado pela OSI e também de uma versão publicada desse aplicativo com um URL de código-fonte. Não é possível atualizar a licença de um aplicativo depois que ele é criado.

Se você não puder tornar seu aplicativo público devido à ausência de um arquivo de licença, exclua o aplicativo e crie um novo com o mesmo nome. Forneça a ele uma ou mais licenças de código aberto aprovadas pela organização Open Source Initiative (OSI).

Uma cota foi excedida

Se você receber uma mensagem de erro indicando que uma cota foi excedida, verifique se você atingiu a cota do recurso. Para as cotas do AWS Serverless Application Repository, consulte Cotas do AWS Serverless Application Repository.

Um arquivo Readme atualizado não aparece imediatamente

Quando você torna seu aplicativo público, o conteúdo dele pode levar até 24 horas para ser atualizado. Se a atualização levar mais do que 24 horas, entre em contatoAWSSupport para ajuda. Para detalhes, veja o seguinte.

Você não pode implantar um aplicativo devido a permissões do IAM insuficientes

Para implantar um aplicativo do AWS Serverless Application Repository, você precisa de permissões para recursos do AWS Serverless Application Repository; e pilhas do AWS CloudFormation. Você também pode precisar de permissão para usar os serviços subjacentes descritos no aplicativo. Por exemplo, se estiver criando um bucket do Amazon S3 ou uma tabela do Amazon DynamoDB, você precisa de permissões para o Amazon S3 ou para o DynamoDB.

Se tiver esse tipo de problema, analise sua política do AWS Identity and Access Management (IAM) e verifique se você tem as permissões necessárias. Para obter mais informações, consulteComo controlar o acesso com oAWSIdentity and Access Management.

Não é possível implantar o mesmo aplicativo duas vezes

O nome de aplicativo que você fornece é usado como o nome da pilha do AWS CloudFormation. Se você tiver problemas para implantar um aplicativo, verifique há alguma pilha do AWS CloudFormation com o mesmo nome. Se houver, forneça um nome de aplicativo diferente ou exclua a pilha existente para implantar o aplicativo com o mesmo nome.

Por que meu aplicativo não está publicamente disponível?

Os aplicativos são privados por padrão. Para tornar seu aplicativo público, siga as etapas descritas aqui.

Como entrar em contato com o Support

Em alguns casos, pode não ser possível encontrar soluções de problemas nesta seção ou por meio dos fóruns do AWS Serverless Application Repository. Se você tiver umAWSPremium Support, você poderá criar um caso de suporte técnico emAWSSuporte para.

Antes de entrar em contatoAWSSupport, tenha em conta o nome de recurso da Amazon (ARN) do aplicativo para o qual você precisa de ajuda. É possível encontrar o ARN do aplicativo no console do AWS Serverless Application Repository.

Operações

A API REST do AWS Serverless Application Repository inclui as seguintes operações.

- CreateApplication

  Cria um aplicativo, opcionalmente, incluindo umAWSArquivo SAM para criar a primeira versão do aplicativo na mesma chamada.
- CreateApplicationVersion

  Cria uma versão do aplicativo.
- CreateCloudFormationChangeSet

  Cria umAWS CloudFormationConjunto de alterações do aplicativo fornecido.
- CreateCloudFormationTemplate

  Cria umAWS CloudFormationTemplate.
- DeleteApplication

  Exclui o aplicativo especificado.
- GetApplication

  Obtém o aplicativo especificado.
- GetApplicationPolicy

  Recupera a política para o aplicativo.
- GetCloudFormationTemplate

  Obtém o especificadoAWS CloudFormationTemplate.
- ListApplicationDependencies

  Recupera a lista de aplicativos aninhados no aplicativo que os contém.
- ListApplications

  Lista os aplicativos de propriedade do solicitante.
- ListApplicationVersions

  Lista as versões do aplicativo especificado.

- PutApplicationPolicy

  Define a política de permissão para um aplicativo. Para obter a lista de ações compatíveis com esta operação, consultePermissões para aplicativos.

- UnshareApplication

  Descompartilha um aplicativo de umAWSOrganização.

  Essa operação só pode ser chamada da conta de gerenciamento da organização.

- UpdateApplication

  Atualiza o aplicativo especificado.

Recursos

A API AWS Serverless Application Repository REST inclui os seguintes recursos.

Tópicos

- Applications
- Aplicativos ApplicationID
- Aplicativos ApplicationID Changesets
- Applications applicationId Dependencies
- Política de ApplicationID de aplicativos
- Applications applicationId Templates
- Applications applicationId Templates templateId
- Applications applicationId Unshare
- Versões do ApplicationID de aplicativos
- Aplicativos: ApplicationID Versions SemanticVersion

Applications

URI

`/applications`

Métodos HTTP

GET

ID da operação: `ListApplications`

Lista os aplicativos de propriedade do solicitante.

Parâmetros de consulta

Nome	Type	Obrigatório	Descrição
`Máximo de itens`	String	False	O número total de itens a serem devolvidos.
`nextToken`	String	False	Um token para especificar por onde começar a paginar.

Respostas

Código de status	Modelo de resposta	Descrição
`200`	`ApplicationPage`	Bem-sucedida
`400`	`BadRequestException`	Um dos parâmetros na solicitação é inválido.
`403`	`ForbiddenException`	O cliente não está autenticado.
`404`	`NotFoundException`	O recurso (por exemplo, uma declaração de política de acesso) especificado na solicitação não existe.
`500`	`InternalServerErrorException`	O serviço AWS Serverless Application Repository encontrou um erro interno.

POST

ID da operação: `CreateApplication`

Cria um aplicativo, incluindo opcionalmente um arquivo AWS SAM para criar a primeira versão do aplicativo na mesma chamada.

Respostas

Código de status	Modelo de resposta	Descrição
201	Application	Bem-sucedida
400	BadRequestException	Um dos parâmetros na solicitação é inválido.
403	ForbiddenException	O cliente não está autenticado.
409	ConflictException	O recurso já existe.
429	TooManyRequestsException	O cliente está enviando mais do que o número permitido de solicitações por unidade de tempo.
500	InternalServerErrorException	O serviço AWS Serverless Application Repository encontrou um erro interno.

OPTIONS

Respostas

Código de status	Modelo de resposta	Descrição
200	None	200 respostas

Esquemas

Corpos da solicitação

Esquema POST

```
{
  "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"
}
```

Corpos da resposta

Esquema ApplicationPage

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

Esquema Application

```
{
```

```
  "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
  }
}
```

Esquema BadRequestException

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

Esquema ForbiddenException

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

Esquema NotFoundException

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

Esquema ConflictException

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

Esquema TooManyRequestsException

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

```
}
```

Esquema InternalServerErrorException

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

Propriedades

Application

Detalhes sobre o aplicativo.

applicationId

O nome de recurso da Amazon (ARN) do aplicativo do .

Tipo: string
Obrigatório: true

name

O nome da aplicação.

Comprimento mínimo = 1. Comprimento máximo = 140

Padrão: "[a-zA-Z0-9\\-]+";

Tipo: string
Obrigatório: true

description

A descrição do aplicativo.

Comprimento mínimo = 1. Comprimento máximo = 256

Tipo: string
Obrigatório: true

author

O nome do autor que publica o aplicativo.

Comprimento mínimo = 1. Tamanho máximo = 127.

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

Tipo: string
Obrigatório: true

isVerifiedAuthor

Especifica se o autor desse aplicativo foi verificado. Isso significa que AWS fez uma análise de boa fé, como prestador de serviços razoável e prudente, das informações fornecidas pelo solicitante e confirmou que a identidade do solicitante é a reivindicada.

Tipo: booliano
Obrigatório: falso

verifiedAuthorUrl

O URL do perfil público de um autor verificado. Esse URL foi enviado pelo autor.

Tipo: string
Obrigatório: falso

spdxLicenseId

Um identificador válido de https://spdx.org/licenses/.

Tipo: string
Obrigatório: falso

licenseUrl

Um link para um arquivo de licença do aplicativo que corresponde ao valor SPDXLicenseID do seu aplicativo.

Tamanho máximo 5 MB

Tipo: string
Obrigatório: falso

readmeUrl

Um link para o arquivo readme na linguagem Markdown que contém uma descrição mais detalhada do aplicativo e de como ele funciona.

Tamanho máximo 5 MB

Tipo: string
Obrigatório: falso

labels

Rótulos para melhorar a descoberta de aplicativos nos resultados de pesquisa.

Comprimento mínimo = 1. Tamanho máximo = 127. Número máximo de etiquetas: 10

Padrão: "^[a-zA-Z0-9+\\-_:\\/@]+$";

Tipo: matriz do tipo string
Obrigatório: falso

creationTime

A data e a hora em que esse recurso foi criado.

Tipo: string
Obrigatório: falso

homePageUrl

Uma URL com mais informações sobre o aplicativo, por exemplo, a localização do seu GitHub repositório para o aplicativo.

Tipo: string
Obrigatório: falso

version

Informações sobre a versão do aplicativo.

Tipo: Versão
Obrigatório: falso

ApplicationPage

Uma lista dos detalhes do aplicativo.

applications

Uma série de resumos de aplicativos.

Tipo: matriz do tipo ApplicationSummary
Obrigatório: true

nextToken

O token para solicitar a próxima página de resultados.

Tipo: string
Obrigatório: falso

ApplicationSummary

Resumo dos detalhes sobre o aplicativo.

applicationId

O nome de recurso da Amazon (ARN) do aplicativo do .

Tipo: string
Obrigatório: true

name

O nome da aplicação.

Comprimento mínimo = 1. Comprimento máximo = 140

Padrão: "[a-zA-Z0-9\\-]+";

Tipo: string
Obrigatório: true

description

A descrição do aplicativo.

Comprimento mínimo = 1. Comprimento máximo = 256

Tipo: string
Obrigatório: true

author

O nome do autor que publica o aplicativo.

Comprimento mínimo = 1. Tamanho máximo = 127.

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

Tipo: string
Obrigatório: true

spdxLicenseId

Um identificador válido de https://spdx.org/licenses/.

Tipo: string
Obrigatório: falso

labels

Rótulos para melhorar a descoberta de aplicativos nos resultados de pesquisa.

Comprimento mínimo = 1. Tamanho máximo = 127. Número máximo de etiquetas: 10

Padrão: "^[a-zA-Z0-9+\\-_:\\/@]+$";

Tipo: matriz do tipo string
Obrigatório: falso

creationTime

A data e a hora em que esse recurso foi criado.

Tipo: string
Obrigatório: falso

homePageUrl

Uma URL com mais informações sobre o aplicativo, por exemplo, a localização do seu GitHub repositório para o aplicativo.

Tipo: string
Obrigatório: falso

BadRequestException

Um dos parâmetros na solicitação é inválido.

message

Um dos parâmetros na solicitação é inválido.

Tipo: string
Obrigatório: falso

errorCode

400

Tipo: string
Obrigatório: falso

Capability

Valores que devem ser especificados para implantar alguns aplicativos.

```
CAPABILITY_IAM
CAPABILITY_NAMED_IAM
CAPABILITY_AUTO_EXPAND
CAPABILITY_RESOURCE_POLICY
```

ConflictException

O recurso já existe.

message

O recurso já existe.

Tipo: string
Obrigatório: falso

errorCode

409

Tipo: string
Obrigatório: falso

CreateApplicationInput

Crie uma solicitação de aplicativo.

name

O nome do aplicativo que você deseja publicar.

Comprimento mínimo = 1. Comprimento máximo = 140

Padrão: "[a-zA-Z0-9\\-]+";

Tipo: string
Obrigatório: true

description

A descrição do aplicativo.

Comprimento mínimo = 1. Comprimento máximo = 256

Tipo: string
Obrigatório: true

author

O nome do autor que publica o aplicativo.

Comprimento mínimo = 1. Tamanho máximo = 127.

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

Tipo: string
Obrigatório: true

spdxLicenseId

Um identificador válido de https://spdx.org/licenses/.

Tipo: string
Obrigatório: falso

licenseBody

Um arquivo de texto local que contém a licença do aplicativo que corresponde ao valor SPDXLicenseID do seu aplicativo. O arquivo tem o formato`file://<path>/<filename>`.

Tamanho máximo 5 MB

Você pode especificar somente um dos `licenseBody` `elicenseUrl`; caso contrário, ocorrerá um erro.

Tipo: string
Obrigatório: falso

licenseUrl

Um link para o objeto S3 que contém a licença do aplicativo que corresponde ao valor SPDXLicenseID do seu aplicativo.

Tamanho máximo 5 MB

Você pode especificar somente um dos `licenseBody` `elicenseUrl`; caso contrário, ocorrerá um erro.

Tipo: string
Obrigatório: falso

readmeBody

Um arquivo readme de texto local na linguagem Markdown que contém uma descrição mais detalhada do aplicativo e de como ele funciona. O arquivo tem o formato`file://<path>/<filename>`.

Tamanho máximo 5 MB

Você pode especificar somente um dos `readmeBody` `ereadmeUrl`; caso contrário, ocorrerá um erro.

Tipo: string
Obrigatório: falso

readmeUrl

Um link para o objeto S3 na linguagem Markdown que contém uma descrição mais detalhada do aplicativo e de como ele funciona.

Tamanho máximo 5 MB

Você pode especificar somente um dos `readmeBody` `ereadmeUrl`; caso contrário, ocorrerá um erro.

Tipo: string
Obrigatório: falso

labels

Rótulos para melhorar a descoberta de aplicativos nos resultados de pesquisa.

Comprimento mínimo = 1. Tamanho máximo = 127. Número máximo de etiquetas: 10

Padrão: "^[a-zA-Z0-9+\\-_:\\/@]+$";

Tipo: matriz do tipo string

Obrigatório: falso

homePageUrl

Uma URL com mais informações sobre o aplicativo, por exemplo, a localização do seu GitHub repositório para o aplicativo.

Tipo: string

Obrigatório: falso

semanticVersion

A versão semântica do aplicativo:

https://semver.org/

Tipo: string

Obrigatório: falso

templateBody

O arquivo de AWS SAM modelo empacotado bruto local do seu aplicativo. O arquivo tem o formato`file://<path>/<filename>`.

Você pode especificar somente um dos `templateBody` e`templateUrl`; caso contrário, ocorrerá um erro.

Tipo: string

Obrigatório: falso

templateUrl

Um link para o objeto S3 contendo o AWS SAM modelo empacotado do seu aplicativo.

Você pode especificar somente um dos `templateBody` `etemplateUrl`; caso contrário, ocorrerá um erro.

Tipo: string
Obrigatório: falso

sourceCodeUrl

Um link para um repositório público para o código-fonte do seu aplicativo, por exemplo, o URL de um GitHub commit específico.

Tipo: string
Obrigatório: falso

sourceCodeArchiveUrl

Um link para o objeto S3 que contém o arquivo ZIP do código-fonte dessa versão do seu aplicativo.

Tamanho máximo 50 MB

Tipo: string
Obrigatório: falso

ForbiddenException

O cliente não está autenticado.

message

O cliente não está autenticado.

Tipo: string
Obrigatório: falso

errorCode

403

Tipo: string
Obrigatório: falso

InternalServerErrorException

O serviço AWS Serverless Application Repository encontrou um erro interno.

message

O serviço AWS Serverless Application Repository encontrou um erro interno.

Tipo: string
Obrigatório: falso

errorCode

500

Tipo: string
Obrigatório: falso

NotFoundException

O recurso (por exemplo, uma declaração de política de acesso) especificado na solicitação não existe.

message

O recurso (por exemplo, uma declaração de política de acesso) especificado na solicitação não existe.

Tipo: string
Obrigatório: falso

errorCode

404

Tipo: string
Obrigatório: falso

ParameterDefinition

Parâmetros suportados pelo aplicativo.

name

O nome do parâmetro.

Tipo: string
Obrigatório: true

defaultValue

Um valor do tipo apropriado para o modelo a ser usado se nenhum valor for especificado quando uma pilha é criada. Se definir restrições para o parâmetro, você deverá especificar um valor que esteja de acordo com essas restrições.

Tipo: string
Obrigatório: falso

description

Uma sequência de até 4.000 caracteres que descreve o parâmetro.

Tipo: string
Obrigatório: falso

type

O tipo do parâmetro.

Valores válidos: `String | Number | List<Number> | CommaDelimitedList`

`String`: Uma sequência de caracteres literal.

Por exemplo, os usuários podem especificar`"MyUserName"`.

`Number`: Um número inteiro ou flutuante. AWS CloudFormationvalida o valor do parâmetro como um número. No entanto, quando você usa o parâmetro em outro lugar em seu modelo (por exemplo, usando a função `Ref` intrínseca), o valor do parâmetro se torna uma string.

Por exemplo, os usuários podem especificar`"8888"`.

`List<Number>`: uma matriz de números inteiros ou flutuantes separados por vírgulas. AWS CloudFormationvalida o valor do parâmetro como números. No entanto, quando você usa o parâmetro em outro lugar do seu modelo (por exemplo, usando a função `Ref` intrínseca), o valor do parâmetro se torna uma lista de cadeias de caracteres.

Por exemplo, os usuários podem especificar “80,20" e, em seguida, resultar em`Ref`. `["80","20"]`

`CommaDelimitedList`: uma matriz de cadeias de caracteres literais separadas por vírgulas. O total de sequências deve ser um número a mais que o número total de vírgulas. Além disso, cada sequência de membros é cortada com espaço.

Por exemplo, os usuários podem especificar “test, dev, prod” e, em seguida, `Ref` resultar em`["test","dev","prod"]`.

Tipo: string
Obrigatório: falso

noEcho

Se o valor do parâmetro deve ser mascarado sempre que alguém fizer uma chamada que descreva a pilha. Se você definir o valor como verdadeiro, o valor do parâmetro será mascarado com asteriscos (*****).

Tipo: booliano
Obrigatório: falso

allowedPattern

Uma expressão regular que representa os padrões a serem permitidos para tipos `String`.

Tipo: string

Obrigatório: falso

constraintDescription

Uma sequência que explica uma restrição quando a restrição é violada. Por exemplo, sem uma descrição da restrição, um parâmetro que tem um padrão permitido de `[A-Za-z0-9]+` exibe a seguinte mensagem de erro quando o usuário especifica um valor inválido:

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

Ao adicionar uma descrição de restrição, como “deve conter somente letras maiúsculas e minúsculas e números”, você pode exibir a seguinte mensagem de erro personalizada:

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

Tipo: string
Obrigatório: falso

minValue

Um valor numérico que determina o menor valor numérico que você deseja permitir para `Number` tipos.

Tipo: inteiro
Obrigatório: falso

maxValue

Um valor numérico que determina o maior valor numérico que você deseja permitir para `Number` tipos.

Tipo: inteiro
Obrigatório: falso

minLength

Um valor inteiro que determina o menor número de caracteres que você deseja permitir para `String` tipos.

Tipo: inteiro
Obrigatório: falso

maxLength

Um valor inteiro que determina o maior número de caracteres que você deseja permitir para `String` tipos.

Tipo: inteiro
Obrigatório: falso

allowedValues

Uma matriz que contém a lista de valores permitidos para o parâmetro.

Tipo: matriz do tipo string
Obrigatório: falso

referencedByResources

Uma lista de AWS SAM recursos que usam esse parâmetro.

Tipo: matriz do tipo string
Obrigatório: true

TooManyRequestsException

O cliente está enviando mais do que o número permitido de solicitações por unidade de tempo.

message

O cliente está enviando mais do que o número permitido de solicitações por unidade de tempo.

Tipo: string
Obrigatório: falso

errorCode

429

Tipo: string

Obrigatório: falso

Version

Detalhes da versão do aplicativo.

applicationId

O nome de recurso da Amazon (ARN) do aplicativo do .

Tipo: string

Obrigatório: true

semanticVersion

A versão semântica do aplicativo:

https://semver.org/

Tipo: string

Obrigatório: true

sourceCodeUrl

Um link para um repositório público para o código-fonte do seu aplicativo, por exemplo, o URL de um GitHub commit específico.

Tipo: string

Obrigatório: falso

sourceCodeArchiveUrl

Um link para o objeto S3 que contém o arquivo ZIP do código-fonte dessa versão do seu aplicativo.

Tamanho máximo 50 MB

Tipo: string

Obrigatório: falso

templateUrl

Um link para o AWS SAM modelo empacotado do seu aplicativo.

Tipo: string
Obrigatório: true

creationTime

A data e a hora em que esse recurso foi criado.

Tipo: string
Obrigatório: true

parameterDefinitions

Uma matriz de tipos de parâmetros suportados pelo aplicativo.

Tipo: matriz do tipo ParameterDefinition
Obrigatório: true

requiredCapabilities

Uma lista de valores que você deve especificar antes de poder implantar determinados aplicativos. Alguns aplicativos podem incluir recursos que podem afetar as permissões em sua AWS conta, por exemplo, criando novos usuários AWS Identity and Access Management (IAM). Para esses aplicativos, você deve reconhecer explicitamente seus recursos especificando esse parâmetro.

Os únicos valores válidos são `CAPABILITY_IAM` `CAPABILITY_NAMED_IAMCAPABILITY_RESOURCE_POLICY`,, `CAPABILITY_AUTO_EXPAND` e.

Os recursos a seguir exigem que você especifique `CAPABILITY_IAM` ou`CAPABILITY_NAMED_IAM`: AWS::IAM::GroupAWS::IAM::InstanceProfileAWS::IAM::Policy,, AWS::IAM::Roleе. Se o aplicativo contiver recursos do IAM, você poderá especificar `CAPABILITY_IAM` ou`CAPABILITY_NAMED_IAM`. Se o aplicativo contém recursos do IAM com nomes personalizados, você deverá especificar `CAPABILITY_NAMED_IAM`.

Os recursos a seguir exigem que você especifique`CAPABILITY_RESOURCE_POLICY`: AWS::Lambda::Permission, AWS: :iam:policy,, AWS::ApplicationAutoScaling::ScalingPolicy, e. AWS::S3::BucketPolicyAWS::SQS::QueuePolicyAWS::SNS::TopicPolicy

Os aplicativos que contêm um ou mais aplicativos aninhados exigem que você especifique `CAPABILITY_AUTO_EXPAND`.

Se seu modelo de aplicativo contiver algum dos recursos acima, recomendamos que você revise todas as permissões associadas ao aplicativo antes da implantação. Se você não especificar esse parâmetro para um aplicativo que requer recursos, a chamada falhará.

Tipo: matriz do tipo Capability
Obrigatório: true

resourcesSupported

Se todos os AWS recursos contidos nesse aplicativo são suportados na região em que ele está sendo recuperado.

Tipo: booliano
Obrigatório: true

Consulte também

Para obter mais informações sobre como usar essa API em um dos AWS SDKs e referências de uma linguagem específica, consulte o seguinte:

ListApplications

- Interface de linha de comando da AWS
- AWS SDK para .NET
- AWS SDK para C++
- AWS SDK para Go
- AWS SDK para Java V2
- SDK da AWS para JavaScript V3
- AWS SDK para PHP V3
- AWS SDK para Python

- AWS SDK para Ruby V3

CreateApplication

- Interface de linha de comando da AWS
- AWS SDK para .NET
- AWS SDK para C++
- AWS SDK para Go
- AWS SDK para Java V2
- SDK da AWS para JavaScript V3
- AWS SDK para PHP V3
- AWS SDK para Python
- AWS SDK para Ruby V3

Aplicativos ApplicationID

URI

`/applications/applicationId`

Métodos HTTP

GET

ID da operação: `GetApplication`

Obtém o aplicativo especificado.

Parâmetros de caminho

Nome	Type	Obrigatório	Descrição
applicationId	String	True	O nome de recurso da Amazon (ARN) do aplicativo.

Parâmetros de consulta

Nome	Type	Obrigatório	Descrição
`Versão semântica`	String	False	A versão semântica do aplicativo a ser obtida.

Respostas

Código de status	Modelo de resposta	Descrição
`200`	`Application`	Bem-sucedida
`400`	`BadRequestException`	Um dos parâmetros na solicitação é inválido.
`403`	`ForbiddenException`	O cliente não está autenticado.
`404`	`NotFoundException`	O recurso (por exemplo, uma declaração de política de acesso) especificado na solicitação não existe.
`429`	`TooManyRequestsException`	O cliente está enviando mais do que o número permitido de solicitações por unidade de tempo.
`500`	`InternalServerErrorException`	O serviço AWS Serverless Application Repository encontrou um erro interno.

DELETE

ID da operação: `DeleteApplication`

Exclui o aplicativo especificado.

Parâmetros de caminho

Nome	Type	Obrigatório	Descrição
applicationId	String	True	O nome de recurso da Amazon (ARN) do aplicativo.

Respostas

Código de status	Modelo de resposta	Descrição
204	None	Bem-sucedida
400	BadRequestException	Um dos parâmetros na solicitação é inválido.
403	ForbiddenException	O cliente não está autenticado.
404	NotFoundException	O recurso (por exemplo, uma declaração de política de acesso) especificado na solicitação não existe.
409	ConflictException	O recurso já existe.
429	TooManyRequestsException	O cliente está enviando mais do que o número permitido de solicitações por unidade de tempo.
500	InternalServerErrorException	O serviço AWS Serverless Application Repository encontrou um erro interno.

OPTIONS

Parâmetros de caminho

Nome	Type	Obrigatório	Descrição
applicationId	String	True	O nome de recurso da Amazon (ARN) do aplicativo.

Respostas

Código de status	Modelo de resposta	Descrição
200	None	200 respostas

PATCH

ID da operação: `UpdateApplication`

Atualiza o aplicativo especificado.

Parâmetros de caminho

Nome	Type	Obrigatório	Descrição
applicationId	String	True	O nome de recurso da Amazon (ARN) do aplicativo.

Respostas

Código de status	Modelo de resposta	Descrição
200	Application	Bem-sucedida
400	BadRequestException	Um dos parâmetros na solicitação é inválido.

Código de status	Modelo de resposta	Descrição
403	ForbiddenException	O cliente não está autenticado.
404	NotFoundException	O recurso (por exemplo, uma declaração de política de acesso) especificado na solicitação não existe.
409	ConflictException	O recurso já existe.
429	TooManyRequestsException	O cliente está enviando mais do que o número permitido de solicitações por unidade de tempo.
500	InternalServerErrorException	O serviço AWS Serverless Application Repository encontrou um erro interno.

Esquemas

Corpos da solicitação

Esquema PATCH

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

Corpos da resposta

Esquema Application

```
{
  "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
  }
}
```

Esquema BadRequestException

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

Esquema ForbiddenException

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

Esquema NotFoundException

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

Esquema ConflictException

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

Esquema TooManyRequestsException

```
{
```

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

Esquema InternalServerErrorException

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

Propriedades

Application

Detalhes sobre o aplicativo.

applicationId

O nome de recurso da Amazon (ARN) do aplicativo do .

Tipo: string
Obrigatório: true

name

O nome da aplicação.

Comprimento mínimo = 1. Comprimento máximo = 140

Padrão: "[a-zA-Z0-9\\-]+";

Tipo: string
Obrigatório: true

description

A descrição do aplicativo.

Comprimento mínimo = 1. Comprimento máximo = 256

Tipo: string
Obrigatório: true

author

O nome do autor que publica o aplicativo.

Comprimento mínimo = 1. Tamanho máximo = 127.

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

Tipo: string
Obrigatório: true

isVerifiedAuthor

Especifica se o autor desse aplicativo foi verificado. Isso significa que AWS fez uma análise de boa fé, como prestador de serviços razoável e prudente, das informações fornecidas pelo solicitante e confirmou que a identidade do solicitante é a reivindicada.

Tipo: booliano
Obrigatório: falso

verifiedAuthorUrl

O URL do perfil público de um autor verificado. Esse URL foi enviado pelo autor.

Tipo: string
Obrigatório: falso

spdxLicenseId

Um identificador válido de https://spdx.org/licenses/.

Tipo: string
Obrigatório: falso

licenseUrl

Um link para um arquivo de licença do aplicativo que corresponde ao valor SPDXLicenseID do seu aplicativo.

Tamanho máximo 5 MB

Tipo: string
Obrigatório: falso

readmeUrl

Um link para o arquivo readme na linguagem Markdown que contém uma descrição mais detalhada do aplicativo e de como ele funciona.

Tamanho máximo 5 MB

Tipo: string
Obrigatório: falso

labels

Rótulos para melhorar a descoberta de aplicativos nos resultados de pesquisa.

Comprimento mínimo = 1. Tamanho máximo = 127. Número máximo de etiquetas: 10

Padrão: "^[a-zA-Z0-9+\\-_:\\/@]+$";

Tipo: matriz do tipo string
Obrigatório: falso

creationTime

A data e a hora em que esse recurso foi criado.

Tipo: string
Obrigatório: falso

homePageUrl

Uma URL com mais informações sobre o aplicativo, por exemplo, a localização do seu GitHub repositório para o aplicativo.

Tipo: string
Obrigatório: falso

version

Informações sobre a versão do aplicativo.

Tipo: Versão
Obrigatório: falso

BadRequestException

Um dos parâmetros na solicitação é inválido.

message

Um dos parâmetros na solicitação é inválido.

Tipo: string
Obrigatório: falso

errorCode

400

Tipo: string
Obrigatório: falso

Capability

Valores que devem ser especificados para implantar alguns aplicativos.

`CAPABILITY_IAM`

```
CAPABILITY_NAMED_IAM
CAPABILITY_AUTO_EXPAND
CAPABILITY_RESOURCE_POLICY
```

ConflictException

O recurso já existe.

message

O recurso já existe.

Tipo: string
Obrigatório: falso

errorCode

409

Tipo: string
Obrigatório: falso

ForbiddenException

O cliente não está autenticado.

message

O cliente não está autenticado.

Tipo: string
Obrigatório: falso

errorCode

403

Tipo: string

Obrigatório: falso

InternalServerErrorException

O serviço AWS Serverless Application Repository encontrou um erro interno.

message

O serviço AWS Serverless Application Repository encontrou um erro interno.

Tipo: string
Obrigatório: falso

errorCode

500

Tipo: string
Obrigatório: falso

NotFoundException

O recurso (por exemplo, uma declaração de política de acesso) especificado na solicitação não existe.

message

O recurso (por exemplo, uma declaração de política de acesso) especificado na solicitação não existe.

Tipo: string
Obrigatório: falso

errorCode

404

Tipo: string
Obrigatório: falso

ParameterDefinition

Parâmetros suportados pelo aplicativo.

name

O nome do parâmetro.

Tipo: string
Obrigatório: true

defaultValue

Um valor do tipo apropriado para o modelo a ser usado se nenhum valor for especificado quando uma pilha é criada. Se definir restrições para o parâmetro, você deverá especificar um valor que esteja de acordo com essas restrições.

Tipo: string
Obrigatório: falso

description

Uma sequência de até 4.000 caracteres que descreve o parâmetro.

Tipo: string
Obrigatório: falso

type

O tipo do parâmetro.

Valores válidos: `String | Number | List<Number> | CommaDelimitedList`

`String`: Uma sequência de caracteres literal.

Por exemplo, os usuários podem especificar`"MyUserName"`.

`Number`: Um número inteiro ou flutuante. AWS CloudFormationvalida o valor do parâmetro como um número. No entanto, quando você usa o parâmetro em outro lugar em seu modelo (por exemplo, usando a função `Ref` intrínseca), o valor do parâmetro se torna uma string.

Por exemplo, os usuários podem especificar`"8888"`.

`List<Number>`: uma matriz de números inteiros ou flutuantes separados por vírgulas. AWS CloudFormationvalida o valor do parâmetro como números. No entanto, quando você usa o parâmetro em outro lugar do seu modelo (por exemplo, usando a função `Ref` intrínseca), o valor do parâmetro se torna uma lista de cadeias de caracteres.

Por exemplo, os usuários podem especificar “80,20" e, em seguida, resultar em`Ref`. `["80","20"]`

`CommaDelimitedList`: uma matriz de cadeias de caracteres literais separadas por vírgulas. O total de sequências deve ser um número a mais que o número total de vírgulas. Além disso, cada sequência de membros é cortada com espaço.

Por exemplo, os usuários podem especificar “test, dev, prod” e, em seguida, `Ref` resultar em`["test","dev","prod"]`.

Tipo: string
Obrigatório: falso

noEcho

Se o valor do parâmetro deve ser mascarado sempre que alguém fizer uma chamada que descreva a pilha. Se você definir o valor como verdadeiro, o valor do parâmetro será mascarado com asteriscos (*****).

Tipo: booliano
Obrigatório: falso

allowedPattern

Uma expressão regular que representa os padrões a serem permitidos para tipos `String`.

Tipo: string
Obrigatório: falso

constraintDescription

Uma sequência que explica uma restrição quando a restrição é violada. Por exemplo, sem uma descrição da restrição, um parâmetro que tem um padrão permitido de `[A-Za-z0-9]+` exibe a seguinte mensagem de erro quando o usuário especifica um valor inválido:

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

Ao adicionar uma descrição de restrição, como “deve conter somente letras maiúsculas e minúsculas e números”, você pode exibir a seguinte mensagem de erro personalizada:

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

Tipo: string
Obrigatório: falso

minValue

Um valor numérico que determina o menor valor numérico que você deseja permitir para `Number` tipos.

Tipo: inteiro
Obrigatório: falso

maxValue

Um valor numérico que determina o maior valor numérico que você deseja permitir para `Number` tipos.

Tipo: inteiro
Obrigatório: falso

minLength

Um valor inteiro que determina o menor número de caracteres que você deseja permitir para `String` tipos.

Tipo: inteiro
Obrigatório: falso

maxLength

Um valor inteiro que determina o maior número de caracteres que você deseja permitir para `String` tipos.

Tipo: inteiro
Obrigatório: falso

allowedValues

Uma matriz que contém a lista de valores permitidos para o parâmetro.

Tipo: matriz do tipo string
Obrigatório: falso

referencedByResources

Uma lista de AWS SAM recursos que usam esse parâmetro.

Tipo: matriz do tipo string
Obrigatório: true

TooManyRequestsException

O cliente está enviando mais do que o número permitido de solicitações por unidade de tempo.

message

O cliente está enviando mais do que o número permitido de solicitações por unidade de tempo.

Tipo: string
Obrigatório: falso

errorCode

429

Tipo: string
Obrigatório: falso

UpdateApplicationInput

Atualize a solicitação do aplicativo.

description

A descrição do aplicativo.

Comprimento mínimo = 1. Comprimento máximo = 256

Tipo: string
Obrigatório: falso

author

O nome do autor que publica o aplicativo.

Comprimento mínimo = 1. Tamanho máximo = 127.

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

Tipo: string
Obrigatório: falso

readmeBody

Um arquivo readme de texto na linguagem Markdown que contém uma descrição mais detalhada do aplicativo e de como ele funciona.

Tamanho máximo 5 MB

Tipo: string
Obrigatório: falso

readmeUrl

Um link para o arquivo readme na linguagem Markdown que contém uma descrição mais detalhada do aplicativo e de como ele funciona.

Tamanho máximo 5 MB

Tipo: string
Obrigatório: falso

labels

Rótulos para melhorar a descoberta de aplicativos nos resultados de pesquisa.

Comprimento mínimo = 1. Tamanho máximo = 127. Número máximo de etiquetas: 10

Padrão: "^[a-zA-Z0-9+\\-_:\\/@]+$";

Tipo: matriz do tipo string
Obrigatório: falso

homePageUrl

Uma URL com mais informações sobre o aplicativo, por exemplo, a localização do seu GitHub repositório para o aplicativo.

Tipo: string
Obrigatório: falso

Version

Detalhes da versão do aplicativo.

applicationId

O nome de recurso da Amazon (ARN) do aplicativo do .

Tipo: string
Obrigatório: true

semanticVersion

A versão semântica do aplicativo:

https://semver.org/

Tipo: string
Obrigatório: true

sourceCodeUrl

Um link para um repositório público para o código-fonte do seu aplicativo, por exemplo, o URL de um GitHub commit específico.

Tipo: string
Obrigatório: falso

sourceCodeArchiveUrl

Um link para o objeto S3 que contém o arquivo ZIP do código-fonte dessa versão do seu aplicativo.

Tamanho máximo 50 MB

Tipo: string
Obrigatório: falso

templateUrl

Um link para o AWS SAM modelo empacotado do seu aplicativo.

Tipo: string
Obrigatório: true

creationTime

A data e a hora em que esse recurso foi criado.

Tipo: string
Obrigatório: true

parameterDefinitions

Uma matriz de tipos de parâmetros suportados pelo aplicativo.

Tipo: matriz do tipo ParameterDefinition
Obrigatório: true

requiredCapabilities

Uma lista de valores que você deve especificar antes de poder implantar determinados aplicativos. Alguns aplicativos podem incluir recursos que podem afetar as permissões em sua AWS conta, por exemplo, criando novos usuários AWS Identity and Access Management (IAM). Para esses aplicativos, você deve reconhecer explicitamente seus recursos especificando esse parâmetro.

Os únicos valores válidos são `CAPABILITY_IAM` `CAPABILITY_NAMED_IAMCAPABILITY_RESOURCE_POLICY`,, `CAPABILITY_AUTO_EXPAND` e.

Os recursos a seguir exigem que você especifique `CAPABILITY_IAM` ou`CAPABILITY_NAMED_IAM`: AWS::IAM::GroupAWS::IAM::InstanceProfileAWS::IAM::Policy,, AWS::IAM::Roleе. Se o aplicativo contiver recursos do IAM, você poderá especificar `CAPABILITY_IAM` ou`CAPABILITY_NAMED_IAM`. Se o aplicativo contém recursos do IAM com nomes personalizados, você deverá especificar `CAPABILITY_NAMED_IAM`.

Os recursos a seguir exigem que você especifique`CAPABILITY_RESOURCE_POLICY`: AWS::Lambda::Permission, AWS: :iam:policy,, AWS::ApplicationAutoScaling::ScalingPolicy, e. AWS::S3::BucketPolicyAWS::SQS::QueuePolicyAWS::SNS::TopicPolicy

Os aplicativos que contêm um ou mais aplicativos aninhados exigem que você especifique `CAPABILITY_AUTO_EXPAND`.

Se seu modelo de aplicativo contiver algum dos recursos acima, recomendamos que você revise todas as permissões associadas ao aplicativo antes da implantação. Se você não especificar esse parâmetro para um aplicativo que requer recursos, a chamada falhará.

Tipo: matriz do tipo Capability
Obrigatório: true

resourcesSupported

Se todos os AWS recursos contidos nesse aplicativo são suportados na região em que ele está sendo recuperado.

Tipo: booliano
Obrigatório: true

Consulte também

Para obter mais informações sobre como usar essa API em um dos AWS SDKs e referências de uma linguagem específica, consulte o seguinte:

GetApplication

- Interface de linha de comando da AWS
- AWS SDK para .NET
- AWS SDK para C++
- AWS SDK para Go
- AWS SDK para Java V2
- SDK da AWS para JavaScript V3
- AWS SDK para PHP V3
- AWS SDK para Python
- AWS SDK para Ruby V3

DeleteApplication

- Interface de linha de comando da AWS
- AWS SDK para .NET
- AWS SDK para C++
- AWS SDK para Go
- AWS SDK para Java V2
- SDK da AWS para JavaScript V3
- AWS SDK para PHP V3
- AWS SDK para Python
- AWS SDK para Ruby V3

UpdateApplication

- Interface de linha de comando da AWS
- AWS SDK para .NET

- AWS SDK para C++
- AWS SDK para Go
- AWS SDK para Java V2
- SDK da AWS para JavaScript V3
- AWS SDK para PHP V3
- AWS SDK para Python
- AWS SDK para Ruby V3

Aplicativos ApplicationID Changesets

URI

`/applications/applicationId/changesets`

Métodos HTTP

POST

ID da operação: `CreateCloudFormationChangeSet`

Cria um conjunto de AWS CloudFormation alterações para o aplicativo em questão.

Parâmetros de caminho

Nome	Type	Obrigatório	Descrição
applicationId	String	True	O nome de recurso da Amazon (ARN) do aplicativo.

Respostas

Código de status	Modelo de resposta	Descrição
201	ChangeSetDetails	Bem-sucedida
400	BadRequestException	Um dos parâmetros na solicitação é inválido.

Código de status	Modelo de resposta	Descrição
403	ForbiddenException	O cliente não está autenticado.
429	TooManyRequestsException	O cliente está enviando mais do que o número permitido de solicitações por unidade de tempo.
500	InternalServerErrorException	O serviço AWS Serverless Application Repository encontrou um erro interno.

OPTIONS

Parâmetros de caminho

Nome	Type	Obrigatório	Descrição
applicationId	String	True	O nome de recurso da Amazon (ARN) do aplicativo.

Respostas

Código de status	Modelo de resposta	Descrição
200	None	200 respostas

Esquemas

Corpos da solicitação

Esquema POST

```
{
```

```
  "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"
    }
  ]
}
```

Corpos da resposta

Esquema ChangeSetDetails

```
{
```

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

Esquema BadRequestException

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

Esquema ForbiddenException

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

Esquema TooManyRequestsException

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

Esquema InternalServerErrorException

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

Propriedades

BadRequestException

Um dos parâmetros na solicitação é inválido.

message

Um dos parâmetros na solicitação é inválido.

Tipo: string
Obrigatório: falso

errorCode

400

Tipo: string
Obrigatório: falso

ChangeSetDetails

Detalhes do conjunto de alterações.

applicationId

O nome de recurso da Amazon (ARN) do aplicativo do .

Tipo: string
Obrigatório: true

semanticVersion

A versão semântica do aplicativo:

https://semver.org/

Tipo: string
Obrigatório: true

changeSetId

O Amazon Resource Name (ARN) do conjunto de alterações.

Restrições de tamanho: tamanho mínimo de 1.

Padrão: ARN: [-a-zA-z0-9:/] *

Tipo: string
Obrigatório: true

stackId

O ID exclusivo da pilha.

Tipo: string
Obrigatório: true

CreateCloudFormationChangeSetInput

Crie uma solicitação de conjunto de alterações do aplicativo.

stackName

Essa propriedade corresponde ao parâmetro com o mesmo nome da AWS CloudFormationCreateChangeSetAPI.

Tipo: string
Obrigatório: true

semanticVersion

A versão semântica do aplicativo:

https://semver.org/

Tipo: string
Obrigatório: falso

templateId

O UUID retornado por. CreateCloudFormationTemplate

Padrão: [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}

Tipo: string

Obrigatório: falso

parameterOverrides

Uma lista de valores de parâmetros para os parâmetros do aplicativo.

Tipo: matriz do tipo ParameterValue
Obrigatório: falso

capabilities

Uma lista de valores que você deve especificar antes de poder implantar determinados aplicativos. Alguns aplicativos podem incluir recursos que podem afetar as permissões em sua AWS conta, por exemplo, criando novos usuários AWS Identity and Access Management (IAM). Para esses aplicativos, você deve reconhecer explicitamente seus recursos especificando esse parâmetro.

Os únicos valores válidos são `CAPABILITY_IAM` `CAPABILITY_NAMED_IAMCAPABILITY_RESOURCE_POLICY`,, `CAPABILITY_AUTO_EXPAND` e.

Os recursos a seguir exigem que você especifique `CAPABILITY_IAM` ou`CAPABILITY_NAMED_IAM`: AWS::IAM::GroupAWS::IAM::InstanceProfileAWS::IAM::Policy,, AWS::IAM::Role e. Se o aplicativo contiver recursos do IAM, você poderá especificar `CAPABILITY_IAM` ou`CAPABILITY_NAMED_IAM`. Se o aplicativo contém recursos do IAM com nomes personalizados, você deverá especificar `CAPABILITY_NAMED_IAM`.

Os recursos a seguir exigem que você especifique`CAPABILITY_RESOURCE_POLICY`: AWS::Lambda::Permission, AWS: :iam:Policy,,, AWS::ApplicationAutoScaling::ScalingPolicyAWS::S3::BucketPolicy, AWS::SQS::QueuePolicye AWS: :SNS:. TopicPolicy

Os aplicativos que contêm um ou mais aplicativos aninhados exigem que você especifique `CAPABILITY_AUTO_EXPAND`.

Se seu modelo de aplicativo contiver algum dos recursos acima, recomendamos que você revise todas as permissões associadas ao aplicativo antes da implantação. Se você não especificar esse parâmetro para um aplicativo que requer recursos, a chamada falhará.

Tipo: matriz do tipo string
Obrigatório: falso

changeSetName

Essa propriedade corresponde ao parâmetro com o mesmo nome da AWS CloudFormationCreateChangeSetAPI.

Tipo: string
Obrigatório: falso

clientToken

Essa propriedade corresponde ao parâmetro com o mesmo nome da AWS CloudFormationCreateChangeSetAPI.

Tipo: string
Obrigatório: falso

description

Essa propriedade corresponde ao parâmetro com o mesmo nome da AWS CloudFormationCreateChangeSetAPI.

Tipo: string
Obrigatório: falso

notificationArns

Essa propriedade corresponde ao parâmetro com o mesmo nome da AWS CloudFormationCreateChangeSetAPI.

Tipo: matriz do tipo string
Obrigatório: falso

resourceTypes

Essa propriedade corresponde ao parâmetro com o mesmo nome da AWS CloudFormationCreateChangeSetAPI.

Tipo: matriz do tipo string
Obrigatório: falso

rollbackConfiguration

Essa propriedade corresponde ao parâmetro com o mesmo nome da AWS CloudFormationCreateChangeSetAPI.

Tipo: RollbackConfiguration
Obrigatório: falso

tags

Essa propriedade corresponde ao parâmetro com o mesmo nome da AWS CloudFormationCreateChangeSetAPI.

Tipo: matriz do tipo Tag
Obrigatório: falso

ForbiddenException

O cliente não está autenticado.

message

O cliente não está autenticado.

Tipo: string
Obrigatório: falso

errorCode

403

Tipo: string
Obrigatório: falso

InternalServerErrorException

O serviço AWS Serverless Application Repository encontrou um erro interno.

message

O serviço AWS Serverless Application Repository encontrou um erro interno.

Tipo: string
Obrigatório: falso

errorCode

500

Tipo: string
Obrigatório: falso

ParameterValue

Valor do parâmetro do aplicativo.

name

A chave associada ao parâmetro. Se você não especificar uma chave e um valor para um parâmetro específico, AWS CloudFormation usa o valor padrão especificado no seu modelo.

Tipo: string
Obrigatório: true

value

O valor de entrada associado ao parâmetro.

Tipo: string
Obrigatório: true

RollbackConfiguration

Essa propriedade corresponde ao Tipo AWS CloudFormationRollbackConfigurationde dados.

rollbackTriggers

Essa propriedade corresponde ao conteúdo do mesmo nome para o Tipo de AWS CloudFormationRollbackConfigurationdados.

Tipo: matriz do tipo RollbackTrigger
Obrigatório: falso

monitoringTimeInMinutes

Essa propriedade corresponde ao conteúdo do mesmo nome para o Tipo de AWS CloudFormationRollbackConfigurationdados.

Tipo: inteiro
Obrigatório: falso

RollbackTrigger

Essa propriedade corresponde ao Tipo AWS CloudFormationRollbackTriggerde dados.

arn

Essa propriedade corresponde ao conteúdo do mesmo nome para o Tipo de AWS CloudFormationRollbackTriggerdados.

Tipo: string
Obrigatório: true

type

Essa propriedade corresponde ao conteúdo do mesmo nome para o Tipo de AWS CloudFormationRollbackTriggerdados.

Tipo: string
Obrigatório: true

Tag

Essa propriedade corresponde ao Tipo de Dados da AWS CloudFormationTag.

key

Essa propriedade corresponde ao conteúdo do mesmo nome para o Tipo de Dados da AWS CloudFormationTag.

Tipo: string
Obrigatório: true

value

Essa propriedade corresponde ao conteúdo do mesmo nome para o Tipo de Dados da AWS CloudFormationTag.

Tipo: string
Obrigatório: true

TooManyRequestsException

O cliente está enviando mais do que o número permitido de solicitações por unidade de tempo.

message

O cliente está enviando mais do que o número permitido de solicitações por unidade de tempo.

Tipo: string
Obrigatório: falso

errorCode

429

Tipo: string
Obrigatório: falso

Consulte também

Para obter mais informações sobre como usar essa API em um dos AWS SDKs e referências de uma linguagem específica, consulte o seguinte:

CreateCloudFormationChangeSet

- Interface de linha de comando da AWS
- AWS SDK para .NET
- AWS SDK para C++
- AWS SDK para Go
- AWS SDK para Java V2
- SDK da AWS para JavaScript V3
- AWS SDK para PHP V3
- AWS SDK para Python
- AWS SDK para Ruby V3

Applications applicationId Dependencies

URI

`/applications/applicationId/dependencies`

Métodos HTTP

GET

ID da operação: `ListApplicationDependencies`

Recupera a lista de aplicativos aninhados no aplicativo que o contém.

Parâmetros de caminho

Nome	Type	Obrigatório	Descrição
applicationId	String	True	O nome de recurso da Amazon (ARN) do aplicativo.

Parâmetros de consulta

Nome	Type	Obrigatório	Descrição
`nextToken`	String	False	Um token para especificar por onde começar a paginar.
`Máximo de itens`	String	False	O número total de itens a serem devolvidos.
`Versão semântica`	String	False	A versão semântica do aplicativo a ser obtida.

Respostas

Código de status	Modelo de resposta	Descrição
200	ApplicationDependencyPage	Bem-sucedida
400	BadRequestException	Um dos parâmetros na solicitação é inválido.
403	ForbiddenException	O cliente não está autenticado.
404	NotFoundException	O recurso (por exemplo, uma declaração de política de acesso) especificado na solicitação não existe.
429	TooManyRequestsException	O cliente está enviando mais do que o número permitido de solicitações por unidade de tempo.

Código de status	Modelo de resposta	Descrição
500	InternalServerErrorException	O serviço AWS Serverless Application Repository encontrou um erro interno.

OPTIONS

Parâmetros de caminho

Nome	Type	Obrigatório	Descrição
applicationId	String	True	O nome de recurso da Amazon (ARN) do aplicativo.

Respostas

Código de status	Modelo de resposta	Descrição
200	None	200 respostas

Esquemas

Corpos da resposta

Esquema ApplicationDependencyPage

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

Esquema BadRequestException

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

Esquema ForbiddenException

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

Esquema NotFoundException

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

Esquema TooManyRequestsException

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

Esquema InternalServerErrorException

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

Propriedades

ApplicationDependencyPage

Uma lista de resumos de aplicativos aninhados no aplicativo.

dependencies

Uma matriz de resumos de aplicativos aninhados no aplicativo.

Tipo: matriz do tipo ApplicationDependencySummary
Obrigatório: true

nextToken

O token para solicitar a próxima página de resultados.

Tipo: string
Obrigatório: falso

ApplicationDependencySummary

Um resumo aninhado do aplicativo.

applicationId

O Amazon Resource Name (ARN) do aplicativo aninhado.

Tipo: string
Obrigatório: true

semanticVersion

A versão semântica do aplicativo aninhado.

Tipo: string
Obrigatório: true

BadRequestException

Um dos parâmetros na solicitação é inválido.

message

Um dos parâmetros na solicitação é inválido.

Tipo: string
Obrigatório: falso

errorCode

400

Tipo: string
Obrigatório: falso

ForbiddenException

O cliente não está autenticado.

message

O cliente não está autenticado.

Tipo: string
Obrigatório: falso

errorCode

403

Tipo: string
Obrigatório: falso

InternalServerErrorException

O serviço AWS Serverless Application Repository encontrou um erro interno.

message

O serviço AWS Serverless Application Repository encontrou um erro interno.

Tipo: string
Obrigatório: falso

errorCode

500

Tipo: string
Obrigatório: falso

NotFoundException

O recurso (por exemplo, uma declaração de política de acesso) especificado na solicitação não existe.

message

O recurso (por exemplo, uma declaração de política de acesso) especificado na solicitação não existe.

Tipo: string
Obrigatório: falso

errorCode

404

Tipo: string
Obrigatório: falso

TooManyRequestsException

O cliente está enviando mais do que o número permitido de solicitações por unidade de tempo.

message

O cliente está enviando mais do que o número permitido de solicitações por unidade de tempo.

Tipo: string
Obrigatório: falso

errorCode

429

Tipo: string
Obrigatório: falso

Consulte também

Para obter mais informações sobre como usar essa API em um dos AWS SDKs e referências de uma linguagem específica, consulte o seguinte:

ListApplicationDependencies

- Interface de linha de comando da AWS
- AWS SDK para .NET
- AWS SDK para C++
- AWS SDK para Go
- AWS SDK para Java V2
- SDK da AWS para JavaScript V3
- AWS SDK para PHP V3
- AWS SDK para Python
- AWS SDK para Ruby V3

Política de ApplicationID de aplicativos

URI

`/applications/applicationId/policy`

Métodos HTTP

GET

ID da operação: `GetApplicationPolicy`

Recupera a política do aplicativo.

Parâmetros de caminho

Nome	Type	Obrigatório	Descrição
applicationId	String	True	O nome de recurso da Amazon (ARN) do aplicativo.

Respostas

Código de status	Modelo de resposta	Descrição
200	ApplicationPolicy	Bem-sucedida
400	BadRequestException	Um dos parâmetros na solicitação é inválido.
403	ForbiddenException	O cliente não está autenticado.
404	NotFoundException	O recurso (por exemplo, uma declaração de política de acesso) especificado na solicitação não existe.
429	TooManyRequestsException	O cliente está enviando mais do que o número permitido de solicitações por unidade de tempo.

Código de status	Modelo de resposta	Descrição
500	InternalServerErrorException	O serviço AWS Serverless Application Repository encontrou um erro interno.

PUT

ID da operação: `PutApplicationPolicy`

Define a política de permissão para um aplicativo. Para ver a lista de ações suportadas por essa operação, consulte Permissões do aplicativo.

Parâmetros de caminho

Nome	Type	Obrigatório	Descrição
applicationId	String	True	O nome de recurso da Amazon (ARN) do aplicativo.

Respostas

Código de status	Modelo de resposta	Descrição
200	ApplicationPolicy	Bem-sucedida
400	BadRequestException	Um dos parâmetros na solicitação é inválido.
403	ForbiddenException	O cliente não está autenticado.
404	NotFoundException	O recurso (por exemplo, uma declaração de política de acesso) especificado na solicitação não existe.

Código de status	Modelo de resposta	Descrição
429	TooManyRequestsException	O cliente está enviando mais do que o número permitido de solicitações por unidade de tempo.
500	InternalServerErrorException	O serviço AWS Serverless Application Repository encontrou um erro interno.

OPTIONS

Parâmetros de caminho

Nome	Type	Obrigatório	Descrição
applicationId	String	True	O nome de recurso da Amazon (ARN) do aplicativo.

Respostas

Código de status	Modelo de resposta	Descrição
200	None	200 respostas

Esquemas

Corpos da solicitação

Esquema PUT

```
{
  "statements": [
    {
      "statementId": "string",
```

```
      "principals": [
        "string"
      ],
      "actions": [
        "string"
      ],
      "principalOrgIDs": [
        "string"
      ]
    }
  ]
}
```

Corpos da resposta

Esquema ApplicationPolicy

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

Esquema BadRequestException

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

Esquema ForbiddenException

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

Esquema NotFoundException

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

Esquema TooManyRequestsException

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

Esquema InternalServerErrorException

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

Propriedades

ApplicationPolicy

Declarações de política aplicadas ao aplicativo.

statements

Uma série de declarações de política aplicadas ao aplicativo.

Tipo: matriz do tipo ApplicationPolicyStatement

Obrigatório: true

ApplicationPolicyStatement

Declaração de política aplicada ao aplicativo.

statementId

Um ID exclusivo para a declaração.

Tipo: string
Obrigatório: falso

principals

Uma matriz de IDs de AWS conta com as quais compartilhar o aplicativo ou* para tornar o aplicativo público.

Tipo: matriz do tipo string
Obrigatório: true

actions

Para ver a lista de ações suportadas por essa operação, consulte Permissões do aplicativo.

Tipo: matriz do tipo string
Obrigatório: true

principalOrgIDs

O AWS Organizations ID com o qual compartilhar o aplicativo.

Tipo: matriz do tipo string
Obrigatório: falso

BadRequestException

Um dos parâmetros na solicitação é inválido.

message

Um dos parâmetros na solicitação é inválido.

Tipo: string
Obrigatório: falso

errorCode

400

Tipo: string
Obrigatório: falso

ForbiddenException

O cliente não está autenticado.

message

O cliente não está autenticado.

Tipo: string
Obrigatório: falso

errorCode

403

Tipo: string
Obrigatório: falso

InternalServerErrorException

O serviço AWS Serverless Application Repository encontrou um erro interno.

message

O serviço AWS Serverless Application Repository encontrou um erro interno.

Tipo: string
Obrigatório: falso

errorCode

500

Tipo: string
Obrigatório: falso

NotFoundException

O recurso (por exemplo, uma declaração de política de acesso) especificado na solicitação não existe.

message

O recurso (por exemplo, uma declaração de política de acesso) especificado na solicitação não existe.

Tipo: string
Obrigatório: falso

errorCode

404

Tipo: string
Obrigatório: falso

TooManyRequestsException

O cliente está enviando mais do que o número permitido de solicitações por unidade de tempo.

message

O cliente está enviando mais do que o número permitido de solicitações por unidade de tempo.

Tipo: string
Obrigatório: falso

errorCode

429

Tipo: string

Obrigatório: falso

Consulte também

Para obter mais informações sobre como usar essa API em um dos AWS SDKs e referências de uma linguagem específica, consulte o seguinte:

GetApplicationPolicy

- Interface de linha de comando da AWS
- AWS SDK para .NET
- AWS SDK para C++
- AWS SDK para Go
- AWS SDK para Java V2
- SDK da AWS para JavaScript V3
- AWS SDK para PHP V3
- AWS SDK para Python
- AWS SDK para Ruby V3

PutApplicationPolicy

- Interface de linha de comando da AWS
- AWS SDK para .NET
- AWS SDK para C++
- AWS SDK para Go
- AWS SDK para Java V2
- SDK da AWS para JavaScript V3
- AWS SDK para PHP V3

- AWS SDK para Python
- AWS SDK para Ruby V3

Applications applicationId Templates

URI

`/applications/applicationId/templates`

Métodos HTTP

POST

ID da operação: `CreateCloudFormationTemplate`

Cria um AWS CloudFormation modelo.

Parâmetros de caminho

Nome	Type	Obrigatório	Descrição
applicationId	String	True	O nome de recurso da Amazon (ARN) do aplicativo.

Respostas

Código de status	Modelo de resposta	Descrição
201	TemplateDetails	Bem-sucedida
400	BadRequestException	Um dos parâmetros na solicitação é inválido.
403	ForbiddenException	O cliente não está autentica do.
404	NotFoundException	O recurso (por exemplo, uma declaração de política

Código de status	Modelo de resposta	Descrição
		de acesso) especificado na solicitação não existe.
429	TooManyRequestsException	O cliente está enviando mais do que o número permitido de solicitações por unidade de tempo.
500	InternalServerErrorException	O serviço AWS Serverless Application Repository encontrou um erro interno.

OPTIONS

Parâmetros de caminho

Nome	Type	Obrigatório	Descrição
applicationId	String	True	O nome de recurso da Amazon (ARN) do aplicativo.

Respostas

Código de status	Modelo de resposta	Descrição
200	None	200 respostas

Esquemas

Corpos da solicitação

Esquema POST

```
{
```

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

Corpos da resposta

Esquema TemplateDetails

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

Esquema BadRequestException

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

Esquema ForbiddenException

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

Esquema NotFoundException

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

Esquema TooManyRequestsException

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

Esquema InternalServerErrorException

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

Propriedades

BadRequestException

Um dos parâmetros na solicitação é inválido.

message

Um dos parâmetros na solicitação é inválido.

Tipo: string
Obrigatório: falso

errorCode

400

Tipo: string
Obrigatório: falso

CreateCloudFormationTemplateInput

Crie uma solicitação de modelo.

semanticVersion

A versão semântica do aplicativo:

https://semver.org/

Tipo: string
Obrigatório: falso

ForbiddenException

O cliente não está autenticado.

message

O cliente não está autenticado.

Tipo: string
Obrigatório: falso

errorCode

403

Tipo: string
Obrigatório: falso

InternalServerErrorException

O serviço AWS Serverless Application Repository encontrou um erro interno.

message

O serviço AWS Serverless Application Repository encontrou um erro interno.

Tipo: string
Obrigatório: falso

errorCode

500

Tipo: string

Obrigatório: falso

NotFoundException

O recurso (por exemplo, uma declaração de política de acesso) especificado na solicitação não existe.

message

O recurso (por exemplo, uma declaração de política de acesso) especificado na solicitação não existe.

Tipo: string
Obrigatório: falso

errorCode

404

Tipo: string
Obrigatório: falso

TemplateDetails

Detalhes do modelo.

templateId

O UUID retornado por. CreateCloudFormationTemplate

Padrão: [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}

Tipo: string
Obrigatório: true

templateUrl

Um link para o modelo que pode ser usado para implantar o aplicativo usandoAWS CloudFormation.

Tipo: string

Obrigatório: true

applicationId

O nome de recurso da Amazon (ARN) do aplicativo do .

Tipo: string
Obrigatório: true

semanticVersion

A versão semântica do aplicativo:

https://semver.org/

Tipo: string
Obrigatório: true

status

Status do fluxo de trabalho de criação do modelo.

Valores possíveis: `PREPARING | ACTIVE | EXPIRED`

Tipo: string
Obrigatório: true
Valores: `PREPARING | ACTIVE | EXPIRED`

creationTime

A data e a hora em que esse recurso foi criado.

Tipo: string
Obrigatório: true

expirationTime

A data e a hora em que esse modelo expira. Os modelos expiram 1 hora após a criação.

Tipo: string
Obrigatório: true

TooManyRequestsException

O cliente está enviando mais do que o número permitido de solicitações por unidade de tempo.

message

O cliente está enviando mais do que o número permitido de solicitações por unidade de tempo.

Tipo: string
Obrigatório: falso

errorCode

429

Tipo: string
Obrigatório: falso

Consulte também

Para obter mais informações sobre como usar essa API em um dos AWS SDKs e referências de uma linguagem específica, consulte o seguinte:

CreateCloudFormationTemplate

- Interface de linha de comando da AWS
- AWS SDK para .NET
- AWS SDK para C++
- AWS SDK para Go
- AWS SDK para Java V2
- SDK da AWS para JavaScript V3
- AWS SDK para PHP V3
- AWS SDK para Python

- AWS SDK para Ruby V3

Applications applicationId Templates templateId

URI

`/applications/applicationId/templates/templateId`

Métodos HTTP

GET

ID da operação: `GetCloudFormationTemplate`

Obtém o AWS CloudFormation modelo especificado.

Parâmetros de caminho

Nome	Type	Obrigatório	Descrição
applicationId	String	True	O nome de recurso da Amazon (ARN) do aplicativo.
ID do modelo	String	True	O UUID retornado por. CreateCloudFormationTemplate Padrão: [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}

Respostas

Código de status	Modelo de resposta	Descrição
200	TemplateDetails	Bem-sucedida

Código de status	Modelo de resposta	Descrição
400	BadRequestException	Um dos parâmetros na solicitação é inválido.
403	ForbiddenException	O cliente não está autenticado.
404	NotFoundException	O recurso (por exemplo, uma declaração de política de acesso) especificado na solicitação não existe.
429	TooManyRequestsException	O cliente está enviando mais do que o número permitido de solicitações por unidade de tempo.
500	InternalServerErrorException	O serviço AWS Serverless Application Repository encontrou um erro interno.

OPTIONS

Parâmetros de caminho

Nome	Type	Obrigatório	Descrição
applicationId	String	True	O nome de recurso da Amazon (ARN) do aplicativo.
ID do modelo	String	True	O UUID retornado por. CreateCloudFormationTemplate Padrão: [0-9a-fa-f] {8}\ - [0-9a-fa-f] {4}\ - [0-9a-fa-f] {4}\ - [0-9a-

Nome	Type	Obrigatório	Descrição
			fa-f] {4}\ - [0-9a-fa-f] {12}

Respostas

Código de status	Modelo de resposta	Descrição
200	None	200 respostas

Esquemas

Corpos da resposta

Esquema TemplateDetails

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

Esquema BadRequestException

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

Esquema ForbiddenException

```
{
```

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

Esquema NotFoundException

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

Esquema TooManyRequestsException

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

Esquema InternalServerErrorException

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

Propriedades

BadRequestException

Um dos parâmetros na solicitação é inválido.

message

Um dos parâmetros na solicitação é inválido.

Tipo: string

Obrigatório: falso

errorCode

400

Tipo: string
Obrigatório: falso

ForbiddenException

O cliente não está autenticado.

message

O cliente não está autenticado.

Tipo: string
Obrigatório: falso

errorCode

403

Tipo: string
Obrigatório: falso

InternalServerErrorException

O serviço AWS Serverless Application Repository encontrou um erro interno.

message

O serviço AWS Serverless Application Repository encontrou um erro interno.

Tipo: string
Obrigatório: falso

errorCode

500

Tipo: string
Obrigatório: falso

NotFoundException

O recurso (por exemplo, uma declaração de política de acesso) especificado na solicitação não existe.

message

O recurso (por exemplo, uma declaração de política de acesso) especificado na solicitação não existe.

Tipo: string
Obrigatório: falso

errorCode

404

Tipo: string
Obrigatório: falso

TemplateDetails

Detalhes do modelo.

templateId

O UUID retornado por. CreateCloudFormationTemplate

Padrão: [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}

Tipo: string
Obrigatório: true

templateUrl

Um link para o modelo que pode ser usado para implantar o aplicativo usandoAWS CloudFormation.

Tipo: string
Obrigatório: true

applicationId

O nome de recurso da Amazon (ARN) do aplicativo do .

Tipo: string
Obrigatório: true

semanticVersion

A versão semântica do aplicativo:

https://semver.org/

Tipo: string
Obrigatório: true

status

Status do fluxo de trabalho de criação do modelo.

Valores possíveis: `PREPARING | ACTIVE | EXPIRED`

Tipo: string
Obrigatório: true
Valores: `PREPARING | ACTIVE | EXPIRED`

creationTime

A data e a hora em que esse recurso foi criado.

Tipo: string
Obrigatório: true

expirationTime

A data e a hora em que esse modelo expira. Os modelos expiram 1 hora após a criação.

Tipo: string
Obrigatório: true

TooManyRequestsException

O cliente está enviando mais do que o número permitido de solicitações por unidade de tempo.

message

O cliente está enviando mais do que o número permitido de solicitações por unidade de tempo.

Tipo: string
Obrigatório: falso

errorCode

429

Tipo: string
Obrigatório: falso

Consulte também

Para obter mais informações sobre como usar essa API em um dos AWS SDKs e referências de uma linguagem específica, consulte o seguinte:

GetCloudFormationTemplate

- Interface de linha de comando da AWS
- AWS SDK para .NET
- AWS SDK para C++
- AWS SDK para Go
- AWS SDK para Java V2
- SDK da AWS para JavaScript V3
- AWS SDK para PHP V3
- AWS SDK para Python

- AWS SDK para Ruby V3

Applications applicationId Unshare

URI

`/applications/applicationId/unshare`

Métodos HTTP

POST

ID da operação: `UnshareApplication`

Cancela o compartilhamento de um aplicativo de uma AWS organização.

Essa operação só pode ser chamada da conta de gerenciamento da organização.

Parâmetros de caminho

Nome	Type	Obrigatório	Descrição
applicationId	String	True	O nome de recurso da Amazon (ARN) do aplicativo.

Respostas

Código de status	Modelo de resposta	Descrição
204	None	Bem-sucedida
400	BadRequestException	Um dos parâmetros na solicitação é inválido.
403	ForbiddenException	O cliente não está autentica do.
404	NotFoundException	O recurso (por exemplo, uma declaração de política

Código de status	Modelo de resposta	Descrição
		de acesso) especificado na solicitação não existe.
429	TooManyRequestsException	O cliente está enviando mais do que o número permitido de solicitações por unidade de tempo.
500	InternalServerErrorException	O serviço AWS Serverless Application Repository encontrou um erro interno.

OPTIONS

Parâmetros de caminho

Nome	Type	Obrigatório	Descrição
applicationId	String	True	O nome de recurso da Amazon (ARN) do aplicativo.

Respostas

Código de status	Modelo de resposta	Descrição
200	None	200 respostas

Esquemas

Corpos da solicitação

Esquema POST

```
{
  "organizationId": "string"
```

```
}
```

Corpos da resposta

Esquema BadRequestException

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

Esquema ForbiddenException

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

Esquema NotFoundException

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

Esquema TooManyRequestsException

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

Esquema InternalServerErrorException

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

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

Propriedades

BadRequestException

Um dos parâmetros na solicitação é inválido.

message

Um dos parâmetros na solicitação é inválido.

Tipo: string
Obrigatório: falso

errorCode

400

Tipo: string
Obrigatório: falso

ForbiddenException

O cliente não está autenticado.

message

O cliente não está autenticado.

Tipo: string
Obrigatório: falso

errorCode

403

Tipo: string

Obrigatório: falso

InternalServerErrorException

O serviço AWS Serverless Application Repository encontrou um erro interno.

message

O serviço AWS Serverless Application Repository encontrou um erro interno.

Tipo: string
Obrigatório: falso

errorCode

500

Tipo: string
Obrigatório: falso

NotFoundException

O recurso (por exemplo, uma declaração de política de acesso) especificado na solicitação não existe.

message

O recurso (por exemplo, uma declaração de política de acesso) especificado na solicitação não existe.

Tipo: string
Obrigatório: falso

errorCode

404

Tipo: string
Obrigatório: falso

TooManyRequestsException

O cliente está enviando mais do que o número permitido de solicitações por unidade de tempo.

message

O cliente está enviando mais do que o número permitido de solicitações por unidade de tempo.

Tipo: string
Obrigatório: falso

errorCode

429

Tipo: string
Obrigatório: falso

UnshareApplicationInput

Cancelar o compartilhamento da solicitação do aplicativo.

organizationId

O AWS Organizations ID do qual cancelar o compartilhamento do aplicativo.

Tipo: string
Obrigatório: true

Consulte também

Para obter mais informações sobre como usar essa API em um dos AWS SDKs e referências de uma linguagem específica, consulte o seguinte:

UnshareApplication

- Interface de linha de comando da AWS
- AWS SDK para .NET
- AWS SDK para C++

- AWS SDK para Go
- AWS SDK para Java V2
- SDK da AWS para JavaScript V3
- AWS SDK para PHP V3
- AWS SDK para Python
- AWS SDK para Ruby V3

Versões do ApplicationID de aplicativos

URI

`/applications/applicationId/versions`

Métodos HTTP

GET

ID da operação: `ListApplicationVersions`

Lista as versões do aplicativo especificado.

Parâmetros de caminho

Nome	Type	Obrigatório	Descrição
applicationId	String	True	O nome de recurso da Amazon (ARN) do aplicativo.

Parâmetros de consulta

Nome	Type	Obrigatório	Descrição
`Máximo de itens`	String	False	O número total de itens a serem devolvidos.

Nome	Type	Obrigatório	Descrição
`nextToken`	String	False	Um token para especificar por onde começar a paginar.

Respostas

Código de status	Modelo de resposta	Descrição
200	ApplicationVersionPage	Bem-sucedida
400	BadRequestException	Um dos parâmetros na solicitação é inválido.
403	ForbiddenException	O cliente não está autenticado.
404	NotFoundException	O recurso (por exemplo, uma declaração de política de acesso) especificado na solicitação não existe.
429	TooManyRequestsException	O cliente está enviando mais do que o número permitido de solicitações por unidade de tempo.
500	InternalServerErrorException	O serviço AWS Serverless Application Repository encontrou um erro interno.

OPTIONS

Parâmetros de caminho

Nome	Type	Obrigatório	Descrição
applicationId	String	True	O nome de recurso da Amazon (ARN) do aplicativo.

Respostas

Código de status	Modelo de resposta	Descrição
200	None	200 respostas

Esquemas

Corpos da resposta

Esquema ApplicationVersionPage

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

Esquema BadRequestException

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

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

Esquema ForbiddenException

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

Esquema NotFoundException

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

Esquema TooManyRequestsException

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

Esquema InternalServerErrorException

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

Propriedades

ApplicationVersionPage

Uma lista de resumos de versões do aplicativo.

versions

Uma série de resumos de versões do aplicativo.

Tipo: matriz do tipo VersionSummary
Obrigatório: true

nextToken

O token para solicitar a próxima página de resultados.

Tipo: string
Obrigatório: falso

BadRequestException

Um dos parâmetros na solicitação é inválido.

message

Um dos parâmetros na solicitação é inválido.

Tipo: string
Obrigatório: falso

errorCode

400

Tipo: string
Obrigatório: falso

ForbiddenException

O cliente não está autenticado.

message

O cliente não está autenticado.

Tipo: string
Obrigatório: falso

errorCode

403

Tipo: string
Obrigatório: falso

InternalServerErrorException

O serviço AWS Serverless Application Repository encontrou um erro interno.

message

O serviço AWS Serverless Application Repository encontrou um erro interno.

Tipo: string
Obrigatório: falso

errorCode

500

Tipo: string
Obrigatório: falso

NotFoundException

O recurso (por exemplo, uma declaração de política de acesso) especificado na solicitação não existe.

message

O recurso (por exemplo, uma declaração de política de acesso) especificado na solicitação não existe.

Tipo: string
Obrigatório: falso

errorCode

404

Tipo: string
Obrigatório: falso

TooManyRequestsException

O cliente está enviando mais do que o número permitido de solicitações por unidade de tempo.

message

O cliente está enviando mais do que o número permitido de solicitações por unidade de tempo.

Tipo: string
Obrigatório: falso

errorCode

429

Tipo: string
Obrigatório: falso

VersionSummary

Um resumo da versão do aplicativo.

applicationId

O nome de recurso da Amazon (ARN) do aplicativo do .

Tipo: string
Obrigatório: true

semanticVersion

A versão semântica do aplicativo:

https://semver.org/

Tipo: string

Obrigatório: true

sourceCodeUrl

Um link para um repositório público para o código-fonte do seu aplicativo, por exemplo, o URL de um GitHub commit específico.

Tipo: string

Obrigatório: falso

creationTime

A data e a hora em que esse recurso foi criado.

Tipo: string

Obrigatório: true

Consulte também

Para obter mais informações sobre como usar essa API em um dos AWS SDKs e referências de uma linguagem específica, consulte o seguinte:

ListApplicationVersions

- Interface de linha de comando da AWS
- AWS SDK para .NET
- AWS SDK para C++
- AWS SDK para Go
- AWS SDK para Java V2
- SDK da AWS para JavaScript V3
- AWS SDK para PHP V3
- AWS SDK para Python

- AWS SDK para Ruby V3

Aplicativos: ApplicationID Versions SemanticVersion

URI

`/applications/applicationId/versions/semanticVersion`

Métodos HTTP

PUT

ID da operação: `CreateApplicationVersion`

Cria uma versão do aplicativo.

Parâmetros de caminho

Nome	Type	Obrigatório	Descrição
applicationId	String	True	O nome de recurso da Amazon (ARN) do aplicativo.
Versão semântica	String	True	A versão semântica da nova versão.

Respostas

Código de status	Modelo de resposta	Descrição
201	Version	Bem-sucedida
400	BadRequestException	Um dos parâmetros na solicitação é inválido.
403	ForbiddenException	O cliente não está autenticado.
409	ConflictException	O recurso já existe.

Código de status	Modelo de resposta	Descrição
429	TooManyRequestsException	O cliente está enviando mais do que o número permitido de solicitações por unidade de tempo.
500	InternalServerErrorException	O serviço AWS Serverless Application Repository encontrou um erro interno.

OPTIONS

Parâmetros de caminho

Nome	Type	Obrigatório	Descrição
applicationId	String	True	O nome de recurso da Amazon (ARN) do aplicativo.
Versão semântica	String	True	A versão semântica da nova versão.

Respostas

Código de status	Modelo de resposta	Descrição
200	None	200 respostas

Esquemas

Corpos da solicitação

Esquema PUT

```
{
  "templateBody": "string",
```

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

Corpos da resposta

Esquema 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
}
```

Esquema BadRequestException

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

Esquema ForbiddenException

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

Esquema ConflictException

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

Esquema TooManyRequestsException

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

Esquema InternalServerErrorException

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

Propriedades

BadRequestException

Um dos parâmetros na solicitação é inválido.

message

Um dos parâmetros na solicitação é inválido.

Tipo: string
Obrigatório: falso

errorCode

400

Tipo: string
Obrigatório: falso

Capability

Valores que devem ser especificados para implantar alguns aplicativos.

```
CAPABILITY_IAM
CAPABILITY_NAMED_IAM
CAPABILITY_AUTO_EXPAND
CAPABILITY_RESOURCE_POLICY
```

ConflictException

O recurso já existe.

message

O recurso já existe.

Tipo: string

Obrigatório: falso

errorCode

409

Tipo: string
Obrigatório: falso

CreateApplicationVersionInput

Crie uma solicitação de versão.

templateBody

O AWS SAM modelo empacotado bruto do seu aplicativo.

Tipo: string
Obrigatório: falso

templateUrl

Um link para o AWS SAM modelo empacotado do seu aplicativo.

Tipo: string
Obrigatório: falso

sourceCodeUrl

Um link para um repositório público para o código-fonte do seu aplicativo, por exemplo, o URL de um GitHub commit específico.

Tipo: string
Obrigatório: falso

sourceCodeArchiveUrl

Um link para o objeto S3 que contém o arquivo ZIP do código-fonte dessa versão do seu aplicativo.

Tamanho máximo 50 MB

Tipo: string
Obrigatório: falso

ForbiddenException

O cliente não está autenticado.

message

O cliente não está autenticado.

Tipo: string
Obrigatório: falso

errorCode

403

Tipo: string
Obrigatório: falso

InternalServerErrorException

O serviço AWS Serverless Application Repository encontrou um erro interno.

message

O serviço AWS Serverless Application Repository encontrou um erro interno.

Tipo: string
Obrigatório: falso

errorCode

500

Tipo: string
Obrigatório: falso

ParameterDefinition

Parâmetros suportados pelo aplicativo.

name

O nome do parâmetro.

Tipo: string
Obrigatório: true

defaultValue

Um valor do tipo apropriado para o modelo a ser usado se nenhum valor for especificado quando uma pilha é criada. Se definir restrições para o parâmetro, você deverá especificar um valor que esteja de acordo com essas restrições.

Tipo: string
Obrigatório: falso

description

Uma sequência de até 4.000 caracteres que descreve o parâmetro.

Tipo: string
Obrigatório: falso

type

O tipo do parâmetro.

Valores válidos: `String | Number | List<Number> | CommaDelimitedList`

`String`: Uma sequência de caracteres literal.

Por exemplo, os usuários podem especificar`"MyUserName"`.

`Number`: Um número inteiro ou flutuante. AWS CloudFormationvalida o valor do parâmetro como um número. No entanto, quando você usa o parâmetro em outro lugar em seu modelo (por exemplo, usando a função `Ref` intrínseca), o valor do parâmetro se torna uma string.

Por exemplo, os usuários podem especificar`"8888"`.

`List<Number>`: uma matriz de números inteiros ou flutuantes separados por vírgulas. AWS CloudFormationvalida o valor do parâmetro como números. No entanto, quando você usa o parâmetro em outro lugar do seu modelo (por exemplo, usando a função `Ref` intrínseca), o valor do parâmetro se torna uma lista de cadeias de caracteres.

Por exemplo, os usuários podem especificar “80,20" e, em seguida, resultar em`Ref`. `["80","20"]`

`CommaDelimitedList`: uma matriz de cadeias de caracteres literais separadas por vírgulas. O total de sequências deve ser um número a mais que o número total de vírgulas. Além disso, cada sequência de membros é cortada com espaço.

Por exemplo, os usuários podem especificar “test, dev, prod” e, em seguida, `Ref` resultar em`["test","dev","prod"]`.

Tipo: string
Obrigatório: falso

noEcho

Se o valor do parâmetro deve ser mascarado sempre que alguém fizer uma chamada que descreva a pilha. Se você definir o valor como verdadeiro, o valor do parâmetro será mascarado com asteriscos (*****).

Tipo: booliano
Obrigatório: falso

allowedPattern

Uma expressão regular que representa os padrões a serem permitidos para tipos `String`.

Tipo: string
Obrigatório: falso

constraintDescription

Uma sequência que explica uma restrição quando a restrição é violada. Por exemplo, sem uma descrição da restrição, um parâmetro que tem um padrão permitido de `[A-Za-z0-9]+` exibe a seguinte mensagem de erro quando o usuário especifica um valor inválido:

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

Ao adicionar uma descrição de restrição, como “deve conter somente letras maiúsculas e minúsculas e números”, você pode exibir a seguinte mensagem de erro personalizada:

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

Tipo: string
Obrigatório: falso

minValue

Um valor numérico que determina o menor valor numérico que você deseja permitir para `Number` tipos.

Tipo: inteiro
Obrigatório: falso

maxValue

Um valor numérico que determina o maior valor numérico que você deseja permitir para `Number` tipos.

Tipo: inteiro
Obrigatório: falso

minLength

Um valor inteiro que determina o menor número de caracteres que você deseja permitir para `String` tipos.

Tipo: inteiro
Obrigatório: falso

maxLength

Um valor inteiro que determina o maior número de caracteres que você deseja permitir para `String` tipos.

Tipo: inteiro
Obrigatório: falso

allowedValues

Uma matriz que contém a lista de valores permitidos para o parâmetro.

Tipo: matriz do tipo string
Obrigatório: falso

referencedByResources

Uma lista de AWS SAM recursos que usam esse parâmetro.

Tipo: matriz do tipo string
Obrigatório: true

TooManyRequestsException

O cliente está enviando mais do que o número permitido de solicitações por unidade de tempo.

message

O cliente está enviando mais do que o número permitido de solicitações por unidade de tempo.

Tipo: string
Obrigatório: falso

errorCode

429

Tipo: string
Obrigatório: falso

Version

Detalhes da versão do aplicativo.

applicationId

O nome de recurso da Amazon (ARN) do aplicativo do .

Tipo: string
Obrigatório: true

semanticVersion

A versão semântica do aplicativo:

https://semver.org/

Tipo: string
Obrigatório: true

sourceCodeUrl

Um link para um repositório público para o código-fonte do seu aplicativo, por exemplo, o URL de um GitHub commit específico.

Tipo: string
Obrigatório: falso

sourceCodeArchiveUrl

Um link para o objeto S3 que contém o arquivo ZIP do código-fonte dessa versão do seu aplicativo.

Tamanho máximo 50 MB

Tipo: string
Obrigatório: falso

templateUrl

Um link para o AWS SAM modelo empacotado do seu aplicativo.

Tipo: string
Obrigatório: true

creationTime

A data e a hora em que esse recurso foi criado.

Tipo: string
Obrigatório: true

parameterDefinitions

Uma matriz de tipos de parâmetros suportados pelo aplicativo.

Tipo: matriz do tipo ParameterDefinition
Obrigatório: true

requiredCapabilities

Uma lista de valores que você deve especificar antes de poder implantar determinados aplicativos. Alguns aplicativos podem incluir recursos que podem afetar as permissões em sua AWS conta, por exemplo, criando novos usuários AWS Identity and Access Management (IAM). Para esses aplicativos, você deve reconhecer explicitamente seus recursos especificando esse parâmetro.

Os únicos valores válidos são `CAPABILITY_IAM` `CAPABILITY_NAMED_IAMCAPABILITY_RESOURCE_POLICY`,, `CAPABILITY_AUTO_EXPAND` e.

Os recursos a seguir exigem que você especifique `CAPABILITY_IAM` ou`CAPABILITY_NAMED_IAM`: AWS::IAM::GroupAWS::IAM::InstanceProfileAWS::IAM::Policy,, AWS::IAM::Roleе. Se o aplicativo contiver recursos do IAM, você poderá especificar `CAPABILITY_IAM` ou`CAPABILITY_NAMED_IAM`. Se o aplicativo contém recursos do IAM com nomes personalizados, você deverá especificar `CAPABILITY_NAMED_IAM`.

Os recursos a seguir exigem que você especifique`CAPABILITY_RESOURCE_POLICY`: AWS::Lambda::Permission, AWS: :iam:policy,, AWS::ApplicationAutoScaling::ScalingPolicy, e. AWS::S3::BucketPolicyAWS::SQS::QueuePolicyAWS::SNS::TopicPolicy

Os aplicativos que contêm um ou mais aplicativos aninhados exigem que você especifique `CAPABILITY_AUTO_EXPAND`.

Se seu modelo de aplicativo contiver algum dos recursos acima, recomendamos que você revise todas as permissões associadas ao aplicativo antes da implantação. Se você não especificar esse parâmetro para um aplicativo que requer recursos, a chamada falhará.

Tipo: matriz do tipo Capability
Obrigatório: true

resourcesSupported

Se todos os AWS recursos contidos nesse aplicativo são suportados na região em que ele está sendo recuperado.

Tipo: booliano
Obrigatório: true

Consulte também

Para obter mais informações sobre como usar essa API em um dos AWS SDKs e referências de uma linguagem específica, consulte o seguinte:

CreateApplicationVersion

- Interface de linha de comando da AWS
- AWS SDK para .NET
- AWS SDK para C++
- AWS SDK para Go
- AWS SDK para Java V2
- SDK da AWS para JavaScript V3
- AWS SDK para PHP V3
- AWS SDK para Python
- AWS SDK para Ruby V3

Histórico do documento

- Versão da API: mais recente
- Última atualização da documentação: 10 de março de 2020

A tabela a seguir descreve as alterações importantes em cada versão do Guia doAWS Serverless Application Repository desenvolvedor do. Para receber notificações sobre atualizações dessa documentação, você pode se inscrever em um feed RSS.

Alteração	Descrição	Data
Atualizações para compartilhar e restringir o acesso a aplicativos	Foi adicionado suporte para compartilhar aplicativos em contas em umaAWS organização e restringir o acesso a aplicativos públicos paraAWS contas eAWS Organizations. Para obter mais exemplos de compartilhamento de aplicativos com usuários em uma organização, consulte Exemplos de políticas deAWS Serverless Application Repository aplicativos. Para obter exemplos de restrição de acesso a aplicativos públicos, consulte Exemplos de políticas baseadas em identidade do AWS Serverless Application Repository.	10 de março de 2020
Novos recursos suportados	Suporte adicionado para diversos recursos agregados. Para obter uma lista completa de recursos compatíve	17 de janeiro de 2020

	is, consulte Lista deAWS recursos compatíveis do.	
regiões da China	O agoraAWS Serverless Application Repository está disponível nas regiões da China, Pequim e do. Para obter mais informações sobre regiões e endpoints do AWS Serverless Application Repository, consulte Regiões e endpoints no Referência geral da AWS.	15 de janeiro de 2020
Seção de segurança atualizada para consistência com outrosAWS serviços.	Para obter mais informações, consulte Segurança.	2 de janeiro de 2020
Processo simplificado para publicação de aplicativos	O novo comando `sam publish` na CLI do AWS SAM simplifica o processo para publicar aplicativos sem servidor no AWS Serverless Application Repository. Para ver um end-to-end tutorial sobre como baixar e publicar um aplicativo de amostra, consulte Início rápido: publicação de aplicativos. Para obter instruções sobre como publicar um aplicativo que você já desenvolveu e testou naAWS nuvem, consulte Publicando um aplicativo por meio daAWS SAM CLI.	21 de dezembro de 2018

Suporte para aplicativos aninhados e camadas	Suporte adicional a aplicativos aninhados e layers Isso inclui atualizações nos AWSrecursos suportados e o reconhecimento dos recursos do aplicativo.	29 de novembro de 2018
Publicando aplicativos com funções e políticas de recursos personalizadas do IAM	Suporte adicional para publicação de aplicativos com funções do IAM e política de recursos personalizadas. Isso inclui atualizações nos fluxos de trabalho de consumo de aplicativos e publicação de aplicativos e atualizações AWSdos recursos suportados e da referência de API no Guia doAWS Serverless Application Repository desenvolvedor.	16 de novembro de 2018
Atualizações do modelo de política	Atualizações dos modelos de políticas compatíveis no Guia doAWS Serverless Application Repository desenvolvedor.	26 de setembro de 2018
Atualizações da documentação	Foi adicionado o tópico Autenticação e Controle de Acesso ao Guia doAWS Serverless Application Repository Desenvolvedor.	2 de julho de 2018

Lançamento público	Lançamento público doAWS Serverless Application Repository, que agora está disponível em 14AWS regiões. Para obter mais informações sobre asAWS regiões em que oAWS Serverless Applicati on Repository está disponíve l eAWS Serverless Applicati on Repository os endpoints, consulte Regiões e endpoints no Referência geral da AWS.	20 de fevereiro de 2018
Novo guia	Esta é a primeira versão prévia do Guia doAWS Serverless Application Repository Desenvolvedor.	30 de novembro de 2017

Glossário da AWS

Para obter a terminologia mais recente da AWS, consulte o glossário da AWS na Referência do Glossário da AWS.

As traduções são geradas por tradução automática. Em caso de conflito entre o conteúdo da tradução e da versão original em inglês, a versão em inglês prevalecerá.