



Guide du développeur

AWS Serverless Application Repository



AWS Serverless Application Repository: Guide du développeur

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

Les marques et la présentation commerciale d'Amazon ne peuvent être utilisées en relation avec un produit ou un service qui n'est pas d'Amazon, d'une manière susceptible de créer une confusion parmi les clients, ou d'une manière qui dénigre ou discrédite Amazon. Toutes les autres marques commerciales qui ne sont pas la propriété d'Amazon appartiennent à leurs propriétaires respectifs, qui peuvent ou non être affiliés ou connectés à Amazon, ou sponsorisés par Amazon.

Table of Contents

Présentation d'AWS Serverless Application Repository	1
Étapes suivantes	1
Démarrage rapide : Publication d'applications	3
Présentation	3
Application Hello World	3
Avant de commencer	4
Étape 1 : Initialiser l'application	4
Étape 2 : Tester l'application localement	5
Étape 3 : Créer le package de l'application	6
Étape 4 : Publier l'application	8
Étapes suivantes	8
En savoir plus	9
Publication des applications	10
Utilisation d'AWS SAM avec l'AWS Serverless Application Repository	11
prises en chargeAWSResources dans leAWS Serverless Application Repository	11
Modèles de stratégie	12
Liste des prises en chargeAWSResources	12
Comment publier des applications	19
Publication d'une application (AWS CLI)	20
Publication d'une nouvelle application (console)	20
Suppression d'une application	26
Annuler le partage d'une application	28
Suppression d'une application	30
Publication de nouvelles versions d'application	30
Badge Auteur vérifié	32
Demande d'un badge Auteur vérifié	32
Partage de couches Lambda	33
Fonctionnement	33
Exemple (Exemple)	34
Déploiement d'applications	35
Autorisations de déploiement d'applications	35
Fonctionnalités des applications	36
Recherche et confirmation des capacités d'une application (console)	37
Affichage des fonctionnalités des applications (AWS CLI)	37

Comment déployer des applications	38
Déploiement d'une nouvelle application (console)	38
Déploiement d'une nouvelle application (AWS CLI)	39
Suppression des piles d'applications	41
Mise à jour des applications	41
Sécurité	43
Protection des données	44
Chiffrement en transit	45
Chiffrement au repos	45
Gestion de l'identité et des accès	45
Public ciblé	46
Authentification avec des identités	47
Gestion des accès à l'aide de politiques	50
Comment AWS Serverless Application Repository fonctionne-t-il avec IAM	53
Exemples de politiques basées sur l'identité	59
Exemples de stratégie de	68
Référence des autorisations d'API AWS Serverless Application Repository	74
Résolution des problèmes	78
Journalisation et surveillance	81
Journalisation des AWS Serverless Application Repository appels d'API avec AWS CloudTrail	81
Validation de la conformité	85
Résilience	86
Sécurité de l'infrastructure	86
Quotas	88
Dépannage	89
Impossible de rendre une application publique	89
Un quota a été dépassé	90
Un fichier Lisez-moi (Readme) mis à jour ne s'affiche pas immédiatement	90
Vous ne pouvez pas déployer une application en raison d'autorisations IAM insuffisantes	90
Vous ne pouvez pas déployer la même application deux fois	90
Pourquoi mon application n'est-elle pas disponible publiquement	91
Contacter Support	91
Opérations	92
Ressources	94
Applications	94

URI	94
HTTPméthodes	94
Schémas	96
Propriétés	100
Consultez aussi	118
Demandes applicationId	119
URI	119
HTTPméthodes	119
Schémas	123
Propriétés	126
Consultez aussi	140
Ensembles de applicationId modifications pour les applications	141
URI	141
HTTPméthodes	141
Schémas	142
Propriétés	144
Consultez aussi	152
Applications applicationId Dependencies	153
URI	153
HTTPméthodes	153
Schémas	155
Propriétés	157
Consultez aussi	160
applicationId Politique relative aux applications	160
URI	160
HTTPméthodes	161
Schémas	163
Propriétés	165
Consultez aussi	168
Applications applicationId Templates	169
URI	169
HTTPméthodes	170
Schémas	171
Propriétés	173
Consultez aussi	177
Applications applicationId Templates templateId	177

URI	177
HTTPméthodes	177
Schémas	179
Propriétés	181
Consultez aussi	185
Applications applicationId Unshare	185
URI	185
HTTPméthodes	185
Schémas	187
Propriétés	188
Consultez aussi	191
applicationId Versions des applications	191
URI	191
HTTPméthodes	192
Schémas	193
Propriétés	195
Consultez aussi	198
applicationId Versions des applications semanticVersion	199
URI	199
HTTPméthodes	199
Schémas	201
Propriétés	203
Consultez aussi	212
Historique du document	213
Glossaire AWS	218
.....	ccxix

Présentation d'AWS Serverless Application Repository

Le AWS Serverless Application Repository permet aux développeurs et aux entreprises de trouver, déployer et publier facilement et rapidement les applications sans serveur dans le AWS Cloud. Pour plus d'informations sur les applications sans serveur, consultez [Calcul et applications sans serveur](#) sur le AWS Site Web.

Vous pouvez publier facilement des applications, les partager publiquement avec la communauté dans son ensemble ou en privé avec votre équipe ou votre entreprise. Pour publier une application sans serveur (ou application), vous pouvez utiliser le AWS Management Console, le AWS SAM Interface de ligne de commande (AWS SAM CLI), ou l'AWS SDK pour télécharger votre code. En même temps que votre code, vous chargez un fichier manifeste simple, aussi appelé modèle AWS Serverless Application Model (AWS SAM). Pour plus d'informations sur AWS SAM, consultez le [Manuel du développeur AWS Serverless Application Model](#).

Le AWS Serverless Application Repository est complètement intégré à la console AWS Lambda. Cette intégration signifie que les développeurs de tous les niveaux peuvent démarrer dans l'informatique sans serveur sans avoir à apprendre quelque chose de nouveau. Vous pouvez utiliser les mots-clés de la catégorie pour rechercher des applications : par exemple, backends web et mobiles, applications de traitement de données ou chatbots. Vous pouvez aussi rechercher les applications par nom, éditeur ou source d'événement. Pour utiliser une application, vous la choisissez, configurez les champs requis et la déployez en quelques clics.

Dans ce guide, vous allez découvrir les deux façons d'utiliser le AWS Serverless Application Repository :

- [Publication des applications](#)— Configurez et chargez les applications pour les rendre accessibles à d'autres développeurs, puis publiez de nouvelles versions des applications.
- [Déploiement d'applications](#)— Recherchez les applications et affichez les informations à leur sujet, y compris les fichiers du code source et le fichier Lisez-moi. De même, installez, configurez et déployez les applications de votre choix.

Étapes suivantes

- Pour obtenir un didacticiel sur la publication d'un exemple d'application sur le AWS Serverless Application Repository, voir [Démarrage rapide : Publication d'applications](#).

- Pour obtenir des instructions sur le déploiement d'applications à partir duAWS Serverless Application Repository, voir [Comment déployer des applications](#).

Démarrage rapide : Publication d'applications

Ce guide vous accompagne à travers les étapes de téléchargement, de création, de test et de publication d'un exemple d'application sans serveur dans AWS Serverless Application Repository en utilisant l'interface de ligne de commande AWS SAM. Vous pouvez utiliser cet exemple d'application comme point de départ pour développer et publier votre propre application sans serveur.

Présentation

Les étapes suivantes décrivent comment télécharger, créer et publier un exemple d'application sans serveur :

1. Initialiser. Téléchargez un exemple d'application à partir du modèle à l'aide de `sam init`.
2. Testez localement. Testez l'application localement en utilisant `sam local invoke` et/ou `sam local start-api`. Notez qu'avec ces commandes, même si votre fonction Lambda est invoquée localement, elle lit et écrit dans AWS des ressources du AWS Cloud.
3. Package. Lorsque vous êtes satisfait de votre fonction Lambda, regroupez la fonction Lambda, le AWS SAM modèle et toutes les dépendances dans un package de AWS CloudFormation déployable à l'aide de `sam package`. Dans cette étape, vous allez également inclure des informations sur l'application qui sera téléchargée vers AWS Serverless Application Repository.
4. Publiez. Publiez l'application sur AWS Serverless Application Repository en utilisant `sam publish`. À la fin de cette étape, vous pouvez visualiser votre application AWS Serverless Application Repository et la déployer dans le AWS cloud à l'aide de AWS Serverless Application Repository.

L'exemple [Application Hello World](#) à la section suivante vous guide à travers ces étapes de création et de publication d'une application sans serveur.

Application Hello World

Dans cet exercice, vous téléchargez et testez une application Hello World sans serveur qui représente un back-end d'API simple. Elle possède un point de terminaison Amazon API Gateway qui prend en charge une opération GET et une fonction Lambda. Lorsqu'une requête GET est envoyée au point de terminaison, l'API Gateway appelle la fonction Lambda. Ensuite, AWS Lambda exécute la fonction, qui renvoie simplement un message `hello world`.

L'application comporte les composants suivants :

- AWS SAM Modèle qui définit deux AWS ressources pour l'application Hello World : un service API Gateway avec une opération GET et une fonction Lambda. Le modèle définit également le mappage entre l'opération GET d'API Gateway et la fonction Lambda.
- Code d'application écrit en Python.

Avant de commencer

Assurez-vous que vous avez la configuration requise pour cet exercice :

- Vous devez disposer d'un AWS compte auprès d'un utilisateur IAM qui dispose d'autorisations d'administrateur. Consultez la section [Créer un AWS compte](#).
- Vous devez avoir l'interface de ligne de commande AWS SAM installée. Veuillez consulter [Installation de l'interface de ligne de commande AWS SAM](#).
- Vous devez disposer de la version 1.16.77 ou ultérieure de la version AWS CLI installée. Veuillez consulter [Installation de AWS Command Line Interface](#).

Étape 1 : Initialiser l'application

Dans cette section, vous téléchargez l'exemple d'application, qui se compose d'un modèle AWS SAM et d'un code d'application.

Pour initialiser l'application

1. Exécutez la commande suivante à l'invite de commande de la CLI AWS SAM.

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

2. Vérifiez le contenu du répertoire créé par la commande (`sam-app/`) :
 - `template.yaml`— Définit deux AWS ressources requises par l'application Hello World : une fonction Lambda et un point de terminaison API Gateway qui prend en charge une opération GET. Le modèle définit également le mappage entre les deux ressources.
 - Contenu lié au code de l'application Hello World :
 - `hello_world/répertoire` — Contient le code de l'application, qui est renvoyé `hello world` lorsque vous l'exécutez.

Note

Pour cet exercice, le code de l'application est écrit en Python et vous spécifiez le moteur d'exécution dans la commande `init`. AWS Lambda prend en charge des langues supplémentaires pour créer du code d'application. Si vous spécifiez un autre moteur d'exécution pris en charge, la commande `init` fournit le code Hello World dans le langage spécifié et un fichier `README.md` que vous pouvez suivre pour ce langage. Pour de plus amples informations sur les runtimes pris en charge, veuillez consulter [Environnement d'exécution Lambda et bibliothèques disponibles](#).

Étape 2 : Tester l'application localement

Maintenant que vous avez l'application AWS SAM sur votre machine locale, suivez les étapes ci-dessous pour la tester localement.

Pour tester l'application localement

1. Démarrez le point de terminaison API Gateway localement. Vous devez exécuter la commande suivante à partir du répertoire qui contient le fichier `template.yaml`.

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

La commande renvoie un point de terminaison API Gateway auquel vous pouvez envoyer des demandes pour des tests locaux.

2. Testez l'application. Copiez l'URL du point de terminaison API Gateway, collez-la dans le navigateur et choisissez Enter. Voici un exemple d'URL de point de terminaison d'API Gateway `http://127.0.0.1:3000/hello`.

API Gateway appelle localement la fonction Lambda à laquelle le point de terminaison est mappé. La fonction Lambda s'exécute dans le conteneur Docker local et renvoie `hello world`. API Gateway renvoie une réponse au navigateur qui contient le texte.

Exercice : Modifier la chaîne de message

Après avoir testé avec succès l'exemple d'application, vous pouvez expérimenter avec une simple modification : modifier la chaîne de message renvoyée.

1. Modifiez le fichier `/hello_world/app.py` pour remplacer la chaîne de message `'hello world'` par `'Hello World!'`.
2. Rechargez l'URL de test dans votre navigateur et observez la nouvelle chaîne.

Vous remarquerez que votre nouveau code est chargé dynamiquement, sans que vous ayez redémarré le processus `sam local`.

Étape 3 : Créer le package de l'application

Après avoir testé votre application localement, vous utilisez l'interface de ligne de commande AWS SAM pour créer un package de déploiement et un modèle AWS SAM empaqueté.

Note

Au cours des étapes suivantes, vous créez un fichier `.zip` pour le contenu du répertoire `hello_world/`, qui contient le code de l'application. Ce fichier `.zip` est le package de déploiement de votre application sans serveur. Pour plus d'informations, consultez la section [Création d'un Package de déploiement \(Python\)](#) dans le Guide du AWS Lambda développeur.

Pour créer un package de déploiement Lambda

1. Ajoutez une section `Metadata` à votre fichier de modèle AWS SAM fournissant les informations requises sur l'application. Pour de plus amples informations sur la section `Metadata` des modèles AWS SAM, veuillez consulter [Propriétés de la section des métadonnées de modèles AWS SAM](#) dans le Guide du développeur AWS Serverless Application Model.

Voici un exemple de section `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
```

Les `ReadmeUrl` propriétés `LicenseUrl` et peuvent être des références à des fichiers locaux (comme dans l'exemple ci-dessus) ou des liens vers des compartiments Amazon S3 qui hébergent déjà ces artefacts.

2. Créez un compartiment S3 à l'emplacement où vous souhaitez enregistrer le code empaqueté. Si vous souhaitez utiliser un compartiment S3 existant, ignorez cette étape.

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

3. Créez le package de déploiement de la fonction Lambda en exécutant la commande `packageAWS` SAM CLI suivante.

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

La commande exécute les opérations suivantes :

- Comprime le contenu du `aws-sam/hello_world/` répertoire et le charge sur Amazon S3.
- Télécharge le package de déploiement, le fichier `README` et le fichier `LICENSE` dans le compartiment Amazon S3 spécifié par l'option `--s3-bucket`.
- Affiche un nouveau fichier modèle, appelé `packaged.yaml`, que vous utilisez à l'étape suivante pour publier l'application AWS Serverless Application Repository. Le fichier `depackaged.yaml` modèle est similaire au fichier de modèle original (`template.yaml`) mais présente une différence significative : le `CodeUriLicenseUrl`, et les `ReadmeUrl` propriétés pointent vers le compartiment Amazon S3 et les objets qui contiennent les artefacts respectifs. L'extrait suivant d'un exemple de fichier de modèle `packaged.yaml` montre la propriété `CodeUri` :

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

Properties:

CodeUri: s3://*bucketname*/*fb77a3647a4f47a352fc0bjectGUID*

...

Étape 4 : Publier l'application

Maintenant que vous avez créé le package de déploiement, vous l'utilisez pour publier l'application sur AWS Serverless Application Repository.

Pour publier l'application sans serveur dans le répertoire AWS Serverless Application Repository

- Exécutez la commande suivante pour publier la nouvelle application AWS Serverless Application Repository avec la première version créée en tant que 0.0.1.

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

Note

L'application sera créée comme privée par défaut. Vous devez partager l'application avant que d'autres AWS comptes soient autorisés à visualiser et à déployer l'application. Consultez Étapes suivantes ci-dessous pour plus de détails sur le partage de votre application.

Étapes suivantes

Maintenant que vous avez publié votre exemple d'application, voici comment vous pouvez l'utiliser.

- Afficher votre application dans AWS Serverless Application Repository : la sortie de la `sam publish` commande inclura un lien AWS Serverless Application Repository vers la page détaillée de votre application. Vous pouvez également accéder à la page de destination AWS Serverless Application Repository et rechercher votre application.
- Partagez votre application : votre application étant définie comme privée par défaut, elle n'est pas visible par les autres AWS comptes. Pour partager votre application avec d'autres utilisateurs, vous devez soit la rendre publique, soit accorder l'autorisation d'accéder à une liste spécifique de AWS

comptes. Pour de plus amples informations sur le partage de votre application à l'aide de AWS CLI, veuillez consulter [AWS Serverless Application Repository Exemples de politiques d'application](#). Pour de plus amples informations sur le partage de votre application à l'aide de la console, veuillez consulter [Suppression d'une application](#).

En savoir plus

Pour de plus amples informations sur la section Metadata des modèles AWS SAM, et les commandes `sam package` et `sam publish` de l'interface de ligne de commande AWS SAM, veuillez consultez [Publication d'applications à l'aide de l'interface de ligne de commande AWS SAM](#) dans le Guide du développeur AWS Serverless Application Model.

Publication des applications

Lorsque vous publiez une application sans serveur sur le AWS Serverless Application Repository, vous la mettez à la disposition d'autres utilisateurs pour qu'ils puissent la rechercher et le déployer.

Vous définissez d'abord votre application avec un modèle AWS Serverless Application Model (AWS SAM). Lorsque vous définissez votre application, vous devez déterminer si ses consommateurs seront tenus de reconnaître les capacités de l'application. Pour de plus amples informations sur l'utilisation de AWS SAM et la reconnaissance des capacités, veuillez consulter [Utilisation d'AWS SAM avec l'AWS Serverless Application Repository](#).

Vous pouvez publier des applications sans serveur à l'aide de l'AWS Management Console, le AWS SAM Interface de ligne de commande (AWS SAM CLI), ou un AWS Kit SDK. Pour de plus amples informations sur les procédures de publication d'applications dans le AWS Serverless Application Repository, veuillez consulter [Comment publier des applications](#).

Lorsque vous publiez votre application, elle est initialement définie sur privé, ce qui signifie qu'il n'est disponible que pour le compte AWS qui l'a créé. Pour partager votre application avec d'autres utilisateurs, vous devez soit la définir sur partagé de façon privée (partagé uniquement avec un ensemble spécifique de comptes AWS), ou publiquement (partagé avec tout le monde).

Lorsque vous publiez une application dans le AWS Serverless Application Repository et que vous la définissez sur publique, le service rend l'application accessible aux consommateurs de toutes les régions. Lorsqu'un consommateur déploie une application publique dans une région autre que celle dans laquelle l'application a été publiée pour la première fois, l'AWS Serverless Application Repository copie les artefacts de déploiement de l'application dans un compartiment Amazon S3 dans la région de destination. Il met à jour toutes les ressources du AWS SAM qui utilisent ces artefacts pour référencer les fichiers dans le compartiment Amazon S3 pour la région de destination. Les artefacts de déploiement peuvent inclure le code de fonction Lambda, les fichiers de définition d'API, etc.

Note

Privé et partagé de façon privée Les applications ne sont disponibles que dans la région AWS dans laquelle ils sont créés. Public Les applications sont disponibles dans toutes les régions AWS. Pour de plus amples informations sur le partage d'applications, veuillez consulter [AWS Serverless Application Repository Exemples de politiques d'application](#).

Rubriques

- [Utilisation d'AWS SAM avec l'AWS Serverless Application Repository](#)
- [Comment publier des applications](#)
- [Badge Auteur vérifié](#)
- [Partage de couches Lambda](#)

Utilisation d'AWS SAM avec l'AWS Serverless Application Repository

Le AWS Serverless Application Model (AWS SAM) est un cadre open source que vous pouvez utiliser pour construire des [applications sans serveur](#) sur AWS. Pour plus d'informations sur l'utilisation d'AWS SAM pour construire votre application sans serveur, consultez le [Guide du développeur AWS Serverless Application Model](#).

Lors de la création d'applications qui seront publiées sur le AWS Serverless Application Repository, vous devez tenir compte du jeu de ressources et modèles de stratégie disponibles à utiliser. Les sections ci-dessous décrivent ces sujets plus en détail.

Prises en charge des ressources dans le AWS Serverless Application Repository

Le AWS Serverless Application Repository prend en charge les applications sans serveur qui sont composées de nombreuses ressources AWS SAM et AWS CloudFormation. Pour afficher la liste complète des ressources prises en charge par le AWS Serverless Application Repository, voir [Liste des prises en charge des ressources](#).

Si vous souhaitez demander du support pour un supplément de ressource, contactez [AWS Support](#).

Important

Si votre modèle d'application contient un ou plusieurs rôles IAM ou stratégies de ressources personnalisés, votre application n'apparaît pas par défaut dans les résultats de la recherche. Les clients doivent également confirmer les rôles IAM ou les stratégies de ressources personnalisés pour pouvoir déployer l'application. Pour plus d'informations, consultez [Confirmation des capacités d'une application](#).

La liste des ressources impactées est la suivante :

- Rôles IAM : [AWS::IAM::Group](#), [AWS::IAM::InstanceProfile](#), [AWS::IAM::Policy](#), et [AWS::IAM::Role](#).
- Stratégies relatives aux ressources : [AWS::Lambda::LayerVersionAutorisation](#), [AWS::Lambda::Permission](#), [AWS::Events::EventBusStratégie](#), [AWS::IAM::Politique](#), [AWS::ApplicationAutoScaling::ScalingPolicy](#), [AWS::S3::BucketPolicy](#), [AWS::SQS::QueuePolicy](#), et [AWS::SNS::TopicPolicy](#).

Si votre application contient la ressource [AWS::Serverless::Application](#), les clients doivent confirmer que l'application contient une application imbriquée pour pouvoir déployer l'application. Pour plus d'informations sur les applications imbriquées, consultez la rubrique relative aux [applications imbriquées](#) dans le Guide du développeur AWS Serverless Application Model. Pour de plus amples informations sur la confirmation des capacités, veuillez consulter [Confirmation des capacités d'une application](#).

Modèles de stratégie

AWS SAM vous permet de trouver une liste de modèles de stratégie afin de limiter les autorisations des fonctions Lambda aux ressources utilisées par votre application. L'utilisation de modèles de stratégie ne nécessite pas d'accusés de réception supplémentaires pour rechercher, parcourir ou déployer l'application.

Pour la liste des standards AWS SAM modèles de stratégie, voir [AWS SAM Modèles de stratégie](#) dans le [AWS Serverless Application Model Manuel du développeur](#).

Liste des prises en charge AWS Ressources

Il s'agit de la liste complète des AWS ressources prises en charge par le AWS Serverless Application Repository.

- `AWS::AccessAnalyzer::Analyzer`
- `AWS::AmazonMQ::Broker`
- `AWS::AmazonMQ::Configuration`
- `AWS::AmazonMQ::ConfigurationAssociation`

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

- `AWS::ApiGatewayV2::Stage`
- `AWS::AppSync::ApiKey`
- `AWS::AppSync::DataSource`
- `AWS::AppSync::GraphQLApi`
- `AWS::AppSync::GraphQLSchema`
- `AWS::AppSync::Resolver`
- `AWS::ApplicationAutoScaling::AutoScalingGroup`
- `AWS::ApplicationAutoScaling::LaunchConfiguration`
- `AWS::ApplicationAutoScaling::ScalableTarget`
- `AWS::ApplicationAutoScaling::ScalingPolicy`
- `AWS::Athena::NamedQuery`
- `AWS::Athena::WorkGroup`
- `AWS::CertificateManager::Certificate`
- `AWS::Chatbot::SlackChannelConfiguration`
- `AWS::CloudFormation::CustomResource`
- `AWS::CloudFormation::Interface`
- `AWS::CloudFormation::Macro`
- `AWS::CloudFormation::WaitConditionHandle`
- `AWS::CloudFront::CachePolicy`
- `AWS::CloudFront::CloudFrontOriginAccessIdentity`
- `AWS::CloudFront::Distribution`
- `AWS::CloudFront::Function`
- `AWS::CloudFront::OriginRequestPolicy`
- `AWS::CloudFront::ResponseHeadersPolicy`
- `AWS::CloudFront::StreamingDistribution`
- `AWS::CloudTrail::Trail`
- `AWS::CloudWatch::Alarm`
- `AWS::CloudWatch::AnomalyDetector`
- `AWS::CloudWatch::Dashboard`

- `AWS::CloudWatch::InsightRule`
- `AWS::CodeBuild::Project`
- `AWS::CodeCommit::Repository`
- `AWS::CodePipeline::CustomActionType`
- `AWS::CodePipeline::Pipeline`
- `AWS::CodePipeline::Webhook`
- `AWS::CodeStar::GitHubRepository`
- `AWS::CodeStarNotifications::NotificationRule`
- `AWS::Cognito::IdentityPool`
- `AWS::Cognito::IdentityPoolRoleAttachment`
- `AWS::Cognito::UserPool`
- `AWS::Cognito::UserPoolClient`
- `AWS::Cognito::UserPoolDomain`
- `AWS::Cognito::UserPoolGroup`
- `AWS::Cognito::UserPoolResourceServer`
- `AWS::Cognito::UserPoolUser`
- `AWS::Cognito::UserPoolUserToGroupAttachment`
- `AWS::Config::AggregationAuthorization`
- `AWS::Config::ConfigRule`
- `AWS::Config::ConfigurationAggregator`
- `AWS::Config::ConfigurationRecorder`
- `AWS::Config::DeliveryChannel`
- `AWS::Config::RemediationConfiguration`
- `AWS::DataPipeline::Pipeline`
- `AWS::DynamoDB::Table`
- `AWS::EC2::EIP`
- `AWS::EC2::InternetGateway`
- `AWS::EC2::NatGateway`
- `AWS::EC2::Route`

- `AWS::EC2::RouteTable`
- `AWS::EC2::SecurityGroup`
- `AWS::EC2::SecurityGroupEgress`
- `AWS::EC2::SecurityGroupIngress`
- `AWS::EC2::Subnet`
- `AWS::EC2::SubnetRouteTableAssociation`
- `AWS::EC2::VPC`
- `AWS::EC2::VPCGatewayAttachment`
- `AWS::EC2::VPCPeeringConnection`
- `AWS::ECR::Repository`
- `AWS::Elasticsearch::Domain`
- `AWS::Events::EventBus`
- `AWS::Events::EventBusPolicy`
- `AWS::Events::Rule`
- `AWS::EventSchemas::Discoverer`
- `AWS::EventSchemas::Registry`
- `AWS::EventSchemas::Schema`
- `AWS::Glue::Classifier`
- `AWS::Glue::Connection`
- `AWS::Glue::Crawler`
- `AWS::Glue::Database`
- `AWS::Glue::DevEndpoint`
- `AWS::Glue::Job`
- `AWS::Glue::Partition`
- `AWS::Glue::SecurityConfiguration`
- `AWS::Glue::Table`
- `AWS::Glue::Trigger`
- `AWS::Glue::Workflow`
- `AWS::IAM::Group`

- `AWS::IAM::InstanceProfile`
- `AWS::IAM::ManagedPolicy`
- `AWS::IAM::OIDCProvider`
- `AWS::IAM::Policy`
- `AWS::IAM::Role`
- `AWS::IAM::ServiceLinkedRole`
- `AWS::IoT::Certificate`
- `AWS::IoT::Policy`
- `AWS::IoT::PolicyPrincipalAttachment`
- `AWS::IoT::Thing`
- `AWS::IoT::ThingPrincipalAttachment`
- `AWS::IoT::TopicRule`
- `AWS::KMS::Alias`
- `AWS::KMS::Key`
- `AWS::Kinesis::Stream`
- `AWS::Kinesis::StreamConsumer`
- `AWS::Kinesis::Streams`
- `AWS::KinesisAnalytics::Application`
- `AWS::KinesisAnalytics::ApplicationOutput`
- `AWS::KinesisFirehose::DeliveryStream`
- `AWS::Lambda::Alias`
- `AWS::Lambda::EventInvokeConfig`
- `AWS::Lambda::EventSourceMapping`
- `AWS::Lambda::Function`
- `AWS::Lambda::LayerVersion`
- `AWS::Lambda::LayerVersionPermission`
- `AWS::Lambda::Permission`
- `AWS::Lambda::Version`
- `AWS::Location::GeofenceCollection`

- `AWS::Location::Map`
- `AWS::Location::PlaceIndex`
- `AWS::Location::RouteCalculator`
- `AWS::Location::Tracker`
- `AWS::Location::TrackerConsumer`
- `AWS::Logs::Destination`
- `AWS::Logs::LogGroup`
- `AWS::Logs::LogStream`
- `AWS::Logs::MetricFilter`
- `AWS::Logs::SubscriptionFilter`
- `AWS::Route53::HealthCheck`
- `AWS::Route53::HostedZone`
- `AWS::Route53::RecordSet`
- `AWS::Route53::RecordSetGroup`
- `AWS::S3::Bucket`
- `AWS::S3::BucketPolicy`
- `AWS::SNS::Subscription`
- `AWS::SNS::Topic`
- `AWS::SNS::TopicPolicy`
- `AWS::SQS::Queue`
- `AWS::SQS::QueuePolicy`
- `AWS::SSM::Association`
- `AWS::SSM::Document`
- `AWS::SSM::MaintenanceWindowTask`
- `AWS::SSM::Parameter`
- `AWS::SSM::PatchBaseline`
- `AWS::SSM::ResourceDataSync`
- `AWS::SecretsManager::ResourcePolicy`
- `AWS::SecretsManager::RotationSchedule`

- `AWS::SecretsManager::Secret`
- `AWS::SecretsManager::SecretTargetAttachment`
- `AWS::Serverless::Api`
- `AWS::Serverless::Application`
- `AWS::Serverless::Function`
- `AWS::Serverless::HttpApi`
- `AWS::Serverless::LayerVersion`
- `AWS::Serverless::SimpleTable`
- `AWS::Serverless::StateMachine`
- `AWS::ServiceDiscovery::HttpNamespace`
- `AWS::ServiceCatalog::CloudFormationProvisionedProduct`
- `AWS::ServiceDiscovery::Instance`
- `AWS::ServiceDiscovery::PrivateDnsNamespace`
- `AWS::ServiceDiscovery::PublicDnsNamespace`
- `AWS::ServiceDiscovery::Service`
- `AWS::SES::ReceiptRule`
- `AWS::SES::ReceiptRuleSet`
- `AWS::StepFunctions::Activity`
- `AWS::StepFunctions::StateMachine`
- `AWS::Wisdom::Assistant`
- `AWS::Wisdom::AssistantAssociation`
- `AWS::Wisdom::KnowledgeBase`

Comment publier des applications

Cette section fournit des procédures de publication de votre application sans serveur vers AWS Serverless Application Repository à l'aide de l'interface de ligne de commande AWS SAM ou de l'AWS Management Console. Elle vous montre également comment partager votre application pour permettre à d'autres utilisateurs de la déployer et à la supprimer de AWS Serverless Application Repository.

⚠ Important

Les informations que vous entrez lorsque vous publiez une application ne sont pas chiffrées. Ces informations comprennent des données telles que le nom de l'auteur. Si vous disposez d'informations personnelles identifiables que vous ne souhaitez pas stocker ou rendre publiques, nous vous recommandons de ne pas les saisir lors de la publication de votre application.

Publication d'une application (AWS CLI)

La façon la plus simple de publier une application sur AWS Serverless Application Repository consiste à utiliser un ensemble de commandes d'interface de ligne de commande AWS SAM. Pour de plus amples informations, veuillez consulter [Publication d'une application à l'aide de l'interface de ligne de commande AWS SAM](#) dans le Guide du développeur AWS Serverless Application Model (AWS SAM).

Publication d'une nouvelle application (console)

Cette section vous montre comment utiliser l'AWS Management Console pour publier une nouvelle application dans AWS Serverless Application Repository. Pour obtenir des instructions sur la publication d'une nouvelle version d'une application existante, veuillez consulter [Publication d'une nouvelle version d'une application existante](#).

Prérequis

Avant de publier une application sur AWS Serverless Application Repository, vous avez besoin des éléments suivants :

- Une validitéAWS.
- Une validitéAWS Serverless Application Model(AWS SAM) qui définit le modèleAWSressources qui sont utilisées. Pour de plus amples informations sur les modèles AWS SAM, veuillez consulter [Concepts de base des modèles AWS SAM](#).
- Un package pour votre application que vous avez créé à l'aide de la commande AWS CloudFormation package pour l'AWS CLI. Cette commande empaquette les artefacts locaux (chemins locaux) auxquels votre modèle AWS SAM fait référence. Pour plus d'informations, consultez [package](#) dans la documentation AWS CloudFormation.

- Une URL qui pointe vers le code source de votre application, si vous souhaitez publier votre application publiquement.
- Un fichier `readme.txt`. Ce fichier doit décrire comment les clients peuvent utiliser votre application et comment la configurer avant de la déployer dans leur propre application et comment la configurer. `AWScomptes`.
- Un fichier `license.txt` ou un identifiant de licence valide du [site web SPDX](#). Notez qu'une licence n'est requise que si vous souhaitez partager votre application publiquement. Si vous voulez garder votre application privée ou la partager uniquement en privé, vous n'avez pas besoin de spécifier de licence.
- Une stratégie de compartiment Amazon S3 valide qui accorde les autorisations de lecture du service pour artefacts chargés sur Amazon S3 quand vous avez empaqueté votre application. Pour définir cette stratégie, procédez comme suit :
 1. Ouvrez la console Amazon S3 sur <https://console.aws.amazon.com/s3/>.
 2. Choisissez le compartiment Amazon S3 que vous avez utilisé pour empaqueter votre application.
 3. Choisissez l'onglet Permissions (Autorisations).
 4. Choisissez le bouton Stratégie de compartiment.
 5. Collez l'instruction de stratégie suivante dans l'éditeur de stratégie de compartiment. Veillez à remplacer le nom de votre compartiment dans le menu `Resource` et votre `AWSID` de compte dans le fichier `Condition` élément. L'expression de l'élément `Condition` assurer AWS Serverless Application Repository n'est autorisé à accéder aux applications qu'à partir de l'option spécifiée `AWS`. Pour plus d'informations sur les déclarations de stratégie, consultez [Référence des éléments de stratégie IAM JSON](#) dans le Guide de l'utilisateur 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. Choisissez le bouton Enregistrer.

Procédure

Créez une application dans AWS Serverless Application Repository à l'aide de la procédure suivante.

Pour créer une application dans AWS Serverless Application Repository

1. Ouvrez la [console AWS Serverless Application Repository](#) et choisissez Publish applications (Publier les applications).
2. Sur la page Publier une application entrez les informations d'application suivantes, puis choisissez Publier une application :

Propriété	Obligatoire	Description
Application name (Nom de l'application)	TRUE	Nom de l'application . Longueur minimale = 1. Longueur maximale = 140. Modèle : [a-zA-Z0-9\-\-]+
Author (Auteur)	TRUE	Nom de l'auteur qui publie l'application. Longueur minimale = 1. Longueur maximale = 127. Modèle : ^ [a-z0-9] (([a-z0-9] - (? ! -)) * [a-z0-9]) ? \$
Page d'accueil	FALSE	Une URL contenant plus d'informations sur l'application, par exemple, l'emplacement

Propriété	Obligatoire	Description
		ment de votre GitHub référentiel de l'application.
Description	TRUE	Description de l'application. Longueur minimale = 1. Longueur maximale = 256.
Étiquettes	FALSE	Les étiquettes qui améliorent la découverte d'applications dans les résultats de recherche. Longueur minimale = 1. Longueur maximale = 127. Nombre maximal d'étiquettes : 10. Modèle : <code>^[a-zA-Z0-9+\\-._:\\V@]+</code> \$
Licence Spdx (liste déroulante)	FALSE	Choisissez un identificateur de licence valide dans la liste déroulante qui contient les licences disponibles sur le site web SPDX . Le choix d'un élément dans la liste déroulante remplit la zone de texte Licence située en dessous de celui-ci. Remarque : Le choix d'une licence dans la liste déroulante remplace le contenu de l'licence et rejette toute modification manuelle que vous avez effectuée.

Propriété	Obligatoire	Description
Licence	FALSE	<p>Téléchargez un fichier de licence .txt ou choisissez une licence dans la liste déroulante de licence Spdx décrite dans la ligne précédente. Le choix d'une licence dans la liste déroulante Licence Spdx remplit automatiquement la zone de texte Licence. Vous pouvez modifier manuellement le contenu de cette zone de texte après avoir téléchargé un fichier de licence ou en avoir choisi un dans la liste déroulante Licence Spdx. Toutefois, si une autre licence Spdx est choisie dans la liste déroulante, toutes les modifications manuelles que vous avez effectuées sont supprimées.</p> <p>Il s'agit d'un champ d'option, mais vous devez fournir une licence afin de partager l'application publiquement.</p>

Propriété	Obligatoire	Description
Readme	FALSE	Téléchargez le contenu du fichier Readme, qui peut être au format texte ou balisage. Ces contenus sont affichés sur la page détaillée de l'application dans le AWS Serverless Application Repository. Vous pouvez modifier manuellement le contenu de cette zone de texte après avoir téléchargé un fichier.
Version sémantique	FALSE	Version sémantique de l'application. Pour de plus amples informations, veuillez consulter le site web Contrôle de version sémantique . Vous devez fournir une valeur pour cette propriété afin de rendre votre application publique.
URL du code source	FALSE	Lien vers un référentiel public pour le code source de votre application.
SAM template (Modèle SAM)	TRUE	Une validitéAWS Serverless Application Model(AWS SAM) qui définit le modèleAWSressources qui sont utilisées.

Suppression d'une application

Les applications publiées peuvent disposer d'autorisations définies dans l'une des trois catégories suivantes :

- Privé (par défaut)— Applications qui ont été créées avec le même compte et qui n'ont pas été partagées avec d'autres AWS. Seuls les consommateurs qui partagent votre AWS sont autorisés à déployer des applications privées.
- Partagé en privé— Applications que l'éditeur a explicitement partagées avec un ensemble spécifique de AWS comptes, ou avec AWS comptes dans un AWS Organisation. Les consommateurs sont autorisés à déployer des applications qui ont été partagées avec leur AWS compte ou AWS Organisation. Pour en savoir plus sur AWS Organizations, consultez le Guide de l'utilisateur [AWS Organizations](#).
- Partagé publiquement— Applications que l'éditeur a partagées avec tout le monde. Tous les consommateurs ont l'autorisation de déployer n'importe quelle application partagée publiquement.

Une fois que vous avez publié une application sur AWS Serverless Application Repository, elle est définie par défaut comme privée. Cette section vous montre comment partager une application en privé avec une application spécifique AWS comptes ou un AWS Organisez ou partagez-le publiquement avec tout le monde.

Partage d'une application via la console

Vous avez deux options pour partager votre application avec d'autres personnes : 1) Partagez-le avec des AWS comptes ou le AWS comptes au sein de votre AWS ou 2) Partagez-le publiquement avec tout le monde. Pour en savoir plus sur AWS Organizations, consultez le Guide de l'utilisateur [AWS Organizations](#).

Option 1 : Pour partager votre application avec des applications spécifiques AWS compte (s) ou comptes au sein de votre AWS organisation

1. Ouvrez la [console AWS Serverless Application Repository](#).
2. Dans le panneau de navigation, choisissez Published Applications (Applications publiées) pour afficher la liste des applications que vous avez créées.
3. Choisissez l'application que vous souhaitez partager.
4. Cliquez sur l'onglet Sharing (Partager) .

5. Dans la section Application policy statements (Déclarations de stratégie d'application), cliquez sur le bouton Create Statement (Créer une déclaration).
6. Dans la fenêtre Statement Configuration (Configuration de la déclaration), remplissez les champs en fonction de la façon dont vous souhaitez partager votre application.

 Note

Si vous partagez avec une organisation, vous pouvez uniquement spécifier l'organisation que votre AWS est membre de. Si vous essayez de spécifier une AWS dont vous n'êtes pas membre, une erreur se produit pour une organisation dont vous n'êtes pas membre.

Pour partager votre application avec votre AWS Organisation, vous devez reconnaître que le `UnshareApplication` sera ajoutée à votre déclaration de stratégie, au cas où le partage devrait être révoqué à l'avenir.

7. Choisissez le bouton Enregistrer.

Option 2 : Partager publiquement votre application avec tout le monde

1. Ouvrez la [console AWS Serverless Application Repository](#).
2. Dans le panneau de navigation, choisissez Published Applications (Applications publiées) pour afficher la liste des applications que vous avez créées.
3. Choisissez l'application que vous souhaitez partager.
4. Cliquez sur l'onglet Sharing (Partager) .
5. Dans la section Public Sharing (Partage public) cliquez sur le bouton Edit (Modifier) .
6. Sous Public sharing (Partage public), choisissez le bouton radio Enabled (Activé) .
7. Dans la zone de texte, saisissez le nom de votre application, puis cliquez sur le bouton Save (Enregistrer) .

 Note

Pour partager publiquement une application, elle doit avoir à la fois définies les propriétés `LicenseUrl` et `SemanticVersion`.

Partage d'une application via l'AWS CLI

Pour partager une application à l'aide de l'AWS CLI vous accordez des autorisations à l'aide du [put-application-policy](#) pour spécifier la commande AWS compte (s) avec lequel vous souhaitez partager en tant que directeurs d'entreprise.

Pour plus d'informations sur comment partager votre application à l'aide de l'AWS CLI, consultez [AWS Serverless Application Repository Exemples de politiques d'application](#).

Annuler le partage d'une application

Il existe deux options pour annuler le partage d'une application d'un AWS Organisation :

1. L'éditeur de l'application peut supprimer des autorisations à l'aide de la commande [put-application-policy](#).
2. Un utilisateur de l'compte de gestion d'un AWS L'organisation peut effectuer un [unshare application](#) opération sur toute application partagée avec l'organisation, même si l'application a été publiée par un utilisateur à partir d'un autre compte.

Note

Lorsqu'une application n'est pas partagée à partir d'un AWS Organisation avec l'opération « annulation du partage de l'application », elle ne peut pas être partagée avec AWS Organisation à nouveau.

Pour en savoir plus sur AWS Organizations, consultez le Guide de l'utilisateur [AWS Organizations](#).

Suppression des autorisations par l'éditeur

Suppression des autorisations par l'éditeur via la console

Pour annuler le partage d'une application via le module AWS Management Console, vous supprimez l'énoncé de stratégie qui la partage avec d'autres AWS comptes. Pour cela, procédez comme suit :

1. Ouvrez la [console AWS Serverless Application Repository](#).
2. Choisissez Available Applications (Applications disponibles) dans le volet de navigation gauche.
3. Choisissez l'application dont vous souhaitez annuler le partage.

4. Cliquez sur l'onglet Sharing (Partager) .
5. Dans la section Application policy statements (Déclarations de stratégie d'application), sélectionnez la déclaration de stratégie qui partage l'application avec les comptes pour lesquels vous souhaitez annuler le partage.
6. Sélectionnez Delete (Supprimer).
7. Un message de confirmation s'affiche. Choisissez Supprimer à nouveau.

Suppression des autorisations par l'éditeur via l'AWS CLI

Pour annuler le partage d'une application via le module AWS CLI, l'éditeur peut supprimer ou modifier les autorisations à l'aide de l'option [put-application-policy](#) pour rendre l'application privée ou partager avec un autre ensemble de AWS comptes.

Pour plus d'informations sur la modification des autorisations à l'aide de l'AWS CLI, consultez [AWS Serverless Application Repository Exemples de politiques d'application](#).

Annulation du partage d'une application par compte de gestion

Compte de gestion qui annule le partage d'une application depuis un AWS Organisation via la console

Pour annuler le partage d'une application d'un AWS Organisation par le biais du AWS Management Console, un utilisateur du compte de gestion Les fonctions des permettent d'effectuer les opérations suivantes :

1. Ouvrez la [console AWS Serverless Application Repository](#).
2. Choisissez Available Applications (Applications disponibles) dans le volet de navigation gauche.
3. Dans la tuile de l'application, choisir Unshare (Annuler le partage).
4. Dans la zone de message d'annulation du partage, confirmez que vous souhaitez annuler le partage de l'application en saisissant l'ID de l'organisation et le nom de l'application, puis en choisissant Save (Enregistrer).

Compte de gestion qui annule le partage d'une application depuis un AWS Organisation par le biais du AWS CLI

Pour annuler le partage d'une application d'un AWS Organisation, un utilisateur de la compte de gestion peut exécuter `aws serverlessrepo unshare-application` commande.

La commande suivante annule le partage d'une application d'un AWS Organisation, où *application* est l'Amazon Resource Name (ARN) de l'application, et *ID de l'organisation* est le AWSID de l'organisation :

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

Suppression d'une application

Vous pouvez supprimer des applications depuis AWS Serverless Application Repository à l'aide de l'AWS Management Console ou de l'interface de ligne de commande AWS SAM.

Suppression d'une application (console)

Pour supprimer une application publiée via l'AWS Management Console, procédez comme suit.

1. Ouvrez la [console AWS Serverless Application Repository](#).
2. Pour My Applications (Mes applications), choisissez l'application que vous souhaitez supprimer.
3. Sur la page des détails de l'application, choisissez Supprimer l'application.
4. Choisissez Supprimer l'application pour terminer la suppression.

Suppression d'une application (AWS CLI)

Pour supprimer une application publiée à l'aide de l'AWS CLI, exécutez la commande [aws serverlessrepo delete-application](#).

La commande suivante supprime une application, où *application-id* est le nom de ressource Amazon (ARN) de l'application :

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

Publication d'une nouvelle version d'une application existante

Cette section vous montre comment publier une nouvelle version d'une application existante sur AWS Serverless Application Repository en utilisant l'interface de ligne de commande AWS SAM ou AWS Management Console. Pour obtenir des instructions sur la publication d'une nouvelle application, veuillez consulter [Comment publier des applications](#).

Publication d'une nouvelle version d'une application existante (AWS CLI)

La façon la plus simple de publier une nouvelle version d'une application existante consiste à utiliser un ensemble de commandes d'interface de ligne de commande AWS SAM. Pour de plus amples informations, veuillez consulter [Publication d'une application à l'aide de l'interface de ligne de commande AWS SAM](#) dans le Guide du développeur AWS Serverless Application Model (AWS SAM).

Publication d'une nouvelle version d'une application existante (console)

Pour publier une nouvelle version d'une application que vous avez déjà publiée, procédez comme suit :

1. Ouvrez la [console AWS Serverless Application Repository](#).
2. Dans le panneau de navigation, choisissez Mes applications pour afficher la liste des applications que vous avez créées.
3. Choisissez l'application pour laquelle vous voulez publier une nouvelle version.
4. Choisissez Publish new version (Publier une nouvelle version).
5. Dans Versions, entrez les informations d'application suivantes :

Propriété	Obligatoire	Description
Version sémantique	TRUE	Version sémantique de l'application. Pour de plus amples informations, veuillez consulter le site web Contrôle de version sémantique . Vous devez fournir une valeur pour cette propriété afin de rendre votre application publique.
URL du code source	FALSE	Lien vers un référentiel public pour le code source de votre application.

Propriété	Obligatoire	Description
SAM template (Modèle SAM)	TRUE	Une validitéAWS Serverless Application Model(AWS SAM) qui définit le modèleAWSressources qui sont utilisées.

6. Choisissez Publier la version.

Badge Auteur vérifié

Auteurs vérifiés dans leAWS Serverless Application Repository sont ceux pour lesquelsAWS a procédé à un examen de bonne foi, en tant que fournisseur de services raisonnable et prudent, des informations fournies par le demandeur et a confirmé que l'identité du demandeur est telle que déclarée.

Les applications des auteurs vérifiés affichent un badge d'auteur vérifié, ainsi qu'un lien vers le profil public de l'auteur. Le badge Auteur vérifié s'affiche à la fois dans les résultats de recherche et sur la page détaillée de l'application.

Demande d'un badge Auteur vérifié

Vous pouvez demander à être approuvé en tant qu'auteur vérifié dans leAWS Serverless Application Repository en envoyant un e-mail à serverlessrepo-verified-author@amazon.com. Vous devez fournir les informations suivantes :

- Nom de l'auteur
- AWSID de compte
- Lien de profil accessible au public, tel que votreGitHubouLinkedInprofil

Après avoir soumis une demande de badge d'auteur vérifié, vous devez attendre à recevoir une réponse deAWS dans quelques jours. Vous serez parfois invité à fournir des informations supplémentaires avant que votre demande soit approuvée.

Une fois votre demande approuvée, le badge d'auteur vérifié est généralement affiché pour vos applications dans un délai d'un jour.

Note

Le badge d'auteur vérifié s'affiche pour toutes les applications qui correspondent à la fois au `AWSnom` du compte et de l'auteur. Étant donné que `AWS` Les comptes peuvent avoir plusieurs auteurs, les badges ne sont pas affichés sur les applications dont le nom d'auteur est différent. Pour que des badges d'auteur apparaissent sur les demandes portant des noms d'auteur différents, vous devez soumettre une autre demande pour cet auteur.

Partage de couches Lambda

Si vous avez implémenté des fonctionnalités dans une couche Lambda, vous pouvez partager cette dernière sans en héberger une instance globale. Le partage de couches de cette manière permet à d'autres personnes de déployer une instance de votre couche sur leur propre compte. Cela empêche les applications clientes de dépendre d'une instance globale de votre couche. Le `AWS Serverless Application Repository` vous permet de partager des couches Lambda de cette manière facilement.

Pour plus d'informations sur les couches Lambda, consultez [AWS Lambda Couches](#) dans le `AWS Lambda Manuel du développeur`.

Fonctionnement

Voici les étapes de partage de votre couche à l'aide de la `AWS Serverless Application Repository`. Cela permet de créer une copie de votre couche dans `AWS`.

1. Définissez une application sans serveur à l'aide d'`AWS SAM` qui inclut votre couche en tant que ressource, c'est-à-dire un `AWS::Serverless::LayerVersion` ou un `AWS::Lambda::LayerVersion` ressource.
2. Publiez votre application sur le `AWS Serverless Application Repository` et partagez-la (publiquement ou en privé).
3. Un client déploie votre application, ce qui crée une copie de votre couche dans son propre `AWS`. Le client peut désormais référencer le nom de ressource Amazon (ARN) de la couche dans son `AWS` dans leur application cliente.

Example (Exemple)

Voici un exemple de AWS SAM pour une application qui contient la couche Lambda que vous souhaitez partager :

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

Lorsqu'un client déploie votre application à partir du AWS Serverless Application Repository, une couche est créée dans leur AWS. L'ARN de la couche ressemble à ce qui suit :

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

Le client peut désormais référencer cet ARN dans sa propre application cliente, comme dans cet exemple :

```
Resources:
  MyFunction:
    Type: AWS::Serverless::Function
    Properties:
      Handler: index.handler
      Runtime: python3.7
      CodeUri: source/app-code/
      Layers:
        - arn:aws:lambda:us-east-1:012345678901:layer:shared-layer:1
```

Déploiement d'applications

Cette section vous apprend à rechercher et à déployer des applications sans serveur publiées sur AWS Serverless Application Repository. Vous pouvez rechercher des applications publiquement disponibles, sans avoir deAWS en visitant le [site public](#). Vous pouvez aussi rechercher des applications à partir de la console AWS Lambda.

Certaines applications ont un badge Auteur vérifié avec un lien vers le profil de l'auteur. Un auteur est considéré comme un auteur vérifié quand AWS a procédé à un examen de bonne foi, en tant que fournisseur de services raisonnable et prudent, des informations fournies par le demandeur et a confirmé que l'identité du demandeur est telle que déclarée.

Avant de déployer des applications à partir de AWS Serverless Application Repository, veuillez consulter les rubriques suivantes pour en savoir plus sur les autorisations de déploiement d'applications et les capacités d'application.

Rubriques

- [Autorisations de déploiement d'applications](#)
- [Fonctionnalités des applications : rôles IAM, stratégies de ressources et applications imbriquées](#)
- [Comment déployer des applications](#)

Autorisations de déploiement d'applications

Pour déployer une application dans le AWS Serverless Application Repository, vous devez avoir l'autorisation de le faire. Il existe trois catégories d'applications pour lesquelles vous disposez d'autorisations de déploiement :

- Privé— Applications créées avec le même compte et qui n'ont pas été partagées avec un autre compte. Vous avez l'autorisation de déployer des applications créées à l'aide de votre AWS.
- Partagé en privé— Applications que l'éditeur a explicitement partagées avec un ensemble spécifique de AWS comptes. Vous avez l'autorisation de déployer des applications qui ont été partagées avec votre AWS.
- Partagé publiquement— Applications que l'éditeur a partagées avec tout le monde. Vous avez l'autorisation de déployer n'importe quelle application partagée publiquement.

Vous pouvez uniquement rechercher et rechercher les applications pour lesquelles vous disposez d'autorisations. Il s'agit notamment des applications créées à l'aide de votre AWS, partagé en privé avec votre AWS et partagés publiquement. Toutes les autres applications ne sont pas affichées pour vous.

Important

Les applications qui contiennent des applications imbriquées héritent des restrictions de partage des applications imbriquées. Supposons par exemple qu'une application soit partagée publiquement, mais qu'elle contienne une application imbriquée qui n'est partagée qu'en privé avec le compte AWS qui a créé l'application parente. Dans ce cas, si votre AWS n'est pas autorisé à déployer l'application imbriquée, vous ne pouvez pas déployer l'application parente. Pour plus d'informations sur les applications imbriquées, consultez la rubrique relative aux [applications imbriquées](#) dans le Guide du développeur AWS Serverless Application Model.

Fonctionnalités des applications : rôles IAM, stratégies de ressources et applications imbriquées

Avant de déployer une application, il AWS Serverless Application Repository vérifie dans le modèle de l'application les rôles IAM, les politiques de ressources AWS et les applications imbriquées que le modèle indique qu'il doit créer. Les ressources IAM, telles qu'un rôle IAM disposant d'un accès complet, peuvent modifier n'importe quelle ressource de votre AWS compte. Par conséquent, nous vous recommandons de passer en revue les autorisations associées à l'application avant de poursuivre. Vous éviterez ainsi de créer par erreur des ressources disposant d'autorisations que vous ne souhaitez pas accorder. Pour ce faire, vous devez confirmer que l'application contient des capacités pour qu'AWS Serverless Application Repository puisse déployer l'application en votre nom.

Les applications peuvent contenir l'une des quatre capacités suivantes : `CAPABILITY_IAM`, `CAPABILITY_NAMED_IAM`, `CAPABILITY_RESOURCE_POLICY` et `CAPABILITY_AUTO_EXPAND`.

Les ressources suivantes nécessitent que vous spécifiiez `CAPABILITY_IAM` ou `CAPABILITY_NAMED_IAM` : [AWS::IAM::Group](#), [AWS::IAM::InstanceProfile](#), [AWS::IAM::Policy](#), et [AWS::IAM::Role](#). Si l'application contient des ressources IAM dotées de noms personnalisés, vous devez spécifier `CAPABILITY_NAMED_IAM`. Pour obtenir un exemple de spécification des capacités, veuillez consulter [Recherche et confirmation des capacités d'une application \(AWS CLI\)](#).

Les ressources suivantes nécessitent que vous spécifiez `CAPABILITY_RESOURCE_POLICY` : [AWS::Lambda::LayerVersionPermission](#), [AWS::Lambda::Permission](#), [AWS::Events::EventBusPolicy](#), [AWS::IAM::Policy](#), [AWS::ApplicationAutoScaling::ScalingPolicy](#), [AWS::S3::BucketPolicy](#), [AWS::SQS::QueuePolicy](#), et [AWS::SNS::TopicPolicy](#).

Les applications contenant une ou plusieurs applications imbriquées exigent de spécifier `CAPABILITY_AUTO_EXPAND`. Pour plus d'informations sur les applications imbriquées, consultez la rubrique relative aux [applications imbriquées](#) dans le Guide du développeur AWS Serverless Application Model.

Recherche et confirmation des capacités d'une application (console)

Vous pouvez trouver des applications disponibles AWS Serverless Application Repository sur le [AWS Serverless Application Repository site Web](#) ou via la [console Lambda \(sur la page Créer une fonction sous l'AWS Serverless Application Repository onglet\)](#).

Les applications qui exigent la confirmation des capacités pour créer des rôles IAM ou des stratégies de ressources personnalisés ne s'affichent pas dans les résultats de la recherche par défaut.

Pour rechercher les applications contenant ces capacités, vous devez cocher la case **Afficher les applications qui créent des rôles IAM ou des stratégies de ressources personnalisés**.

Vous pouvez passer en revue les capacités d'une application dans l'onglet **Permissions (Autorisations)** lorsque vous sélectionnez l'application. Pour déployer l'application, vous devez cocher la case **Je confirme que cette application crée des rôles IAM ou des stratégies de ressources personnalisés**. Si vous ne confirmez pas ces fonctionnalités, le message d'erreur suivant s'affiche : **Reconnaissance requise**. Pour déployer, cochez la case dans la section **Configurer les paramètres de l'application**.

Affichage des fonctionnalités des applications (AWS CLI)

Pour afficher les capacités d'une application à l'aide de AWS CLI, vous avez d'abord besoin de l'ARN (Amazon Resource Name) de l'application. Vous pouvez ensuite exécuter la commande suivante :

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

La propriété de réponse [requiredCapabilities](#) contient la liste des capacités d'application que vous devez confirmer pour pouvoir déployer l'application. Notez que si la propriété [requiredCapabilities](#) est vide, l'application n'a pas de capacités requises.

Comment déployer des applications

Cette section fournit des procédures de déploiement d'applications sans serveur à partir de la AWS Serverless Application Repository en utilisant le AWS Management Console ou l'AWS CLI.

Déploiement d'une nouvelle application (console)

Cette section vous montre comment déployer une nouvelle application à partir de AWS Serverless Application Repository en utilisant AWS Management Console. Pour obtenir des instructions sur le déploiement d'une nouvelle version d'une application existante, veuillez consulter [Mise à jour des applications](#).

Exploration, recherche et déploiement d'applications

Recherchez, configurez et déployez une application dans AWS Serverless Application Repository à l'aide de la procédure suivante.

Pour rechercher et configurer une application dans AWS Serverless Application Repository

1. Ouvrez la [page d'accueil publique AWS Serverless Application Repository](#), ou ouvrez la [console AWS Lambda](#). Choisissez Créer une fonction, puis choisissez Parcourir le référentiel d'applications sans serveur.
2. Recherchez une application.

Note

Pour afficher les applications contenant des rôles IAM ou des stratégies de ressources personnalisés, cochez la case Show apps that create custom IAM roles or resource policies (Afficher les applications qui créent des rôles IAM ou des stratégies de ressources personnalisés). Pour plus d'informations sur les rôles IAM et les stratégies de ressources personnalisés, consultez [Confirmation des capacités d'une application](#).

3. Choisissez une application pour afficher des détails tels que ses autorisations, ses fonctionnalités et le nombre de fois où elle a été déployée par AWS les clients.

Le nombre de déploiements s'affiche pour la AWS région dans laquelle vous essayez de déployer l'application.

4. Sur la page des détails de l'application, consultez les autorisations et les ressources de l'application en affichant le modèle AWS SAM, la licence et le fichier Lisez-moi. Sur cette

page, vous pouvez également rechercher le lien Source code URL (URL du code source) des applications publiquement partagées. Si l'application comprend des applications imbriquées, vous pouvez également afficher les détails de ces applications sur cette page.

5. Configurez l'application dans la section Application settings (Paramètres de l'application). Pour obtenir des instructions sur la configuration d'une application particulière, consultez le fichier Lisez-moi de l'application.

Par exemple, la configuration requise peut inclure la spécification du nom d'une ressource à laquelle vous voulez que l'application accède. Une telle ressource peut être une table Amazon DynamoDB, un compartiment Amazon S3 ou une API Amazon API Gateway.

6. Choisissez Deploy (Déployer). Vous accédez ainsi à la page Statut du déploiement.

Note

Si l'application dispose de fonctionnalités nécessitant un accusé de réception, vous devez cocher la case J'accuse réception que cette application crée des rôles IAM personnalisés ou des stratégies de ressources avant de déployer l'application. Si vous ne le faites pas, une erreur se produit. Pour plus d'informations sur les rôles IAM et les stratégies de ressources personnalisés, consultez [Confirmation des capacités d'une application](#).

7. Sur la page Deployment status (Statut du déploiement), vous pouvez consulter la progression de votre déploiement. En attendant la fin de votre déploiement, vous pouvez rechercher et parcourir d'autres applications, puis revenir à cette page via la console Lambda.

Une fois votre application déployée avec succès, vous pouvez consulter et gérer les ressources qui ont été créées à l'aide d'AWS des outils existants.

Déploiement d'une nouvelle application (AWS CLI)

Cette section vous montre comment déployer une nouvelle application à partir de l'AWS Serverless Application Repository en utilisant la AWS CLI. Pour obtenir des instructions sur le déploiement d'une nouvelle version d'une application existante, veuillez consulter [Mise à jour des applications](#).

Recherche et confirmation des capacités d'une application (AWS CLI)

Pour confirmer les capacités d'une application à l'aide de l'AWS CLI, procédez comme suit :

1. Passez en revue les fonctionnalités de l'application. Utilisez la AWS CLI commande suivante pour vérifier les fonctionnalités d'une application :

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

La propriété de réponse [requiredCapabilities](#) contient la liste des capacités d'application que vous devez confirmer pour pouvoir déployer l'application. Vous pouvez également utiliser l'[GetApplication API](#) des AWS SDK pour obtenir ces données.

2. Créez le changeset. Vous devez fournir l'ensemble des [fonctionnalités](#) requises lorsque vous créez le AWS CloudFormation changeset. Par exemple, utilisez la commande de l'AWS CLI suivante pour déployer une application en confirmant ses capacités :

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

L'ID de changeset est renvoyé lorsque cette commande est exécutée avec succès. Vous avez besoin de l'ID de changeset pour l'étape suivante. Vous pouvez également utiliser l'[CreateCloudFormationChangeSet API](#) des AWS SDK pour créer le changeset.

Par exemple, la AWS CLI commande suivante reconnaît une application qui contient une [AWS::IAM::Role](#) ressource avec un nom personnalisé et une ou plusieurs applications imbriquées :

```
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. Exécutez le changeset. L'exécution du changeset effectue réellement le déploiement. Indiquez l'ID de changeset qui a été renvoyé lorsque vous avez créé le changeset à l'étape précédente.

L'exemple de commande AWS CLI suivant exécute le changeset de l'application pour déployer cette dernière :

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

Vous pouvez également utiliser l'[ExecuteChangeSet API](#) des AWS SDK pour exécuter le `changeset`.

Suppression des piles d'applications

Pour supprimer une application précédemment déployée à l'aide de l'AWS Serverless Application Repository, suivez la même procédure que pour la suppression d'une pile AWS CloudFormation :

- AWS Management Console: Pour supprimer une application à l'aide de l'AWS Management Console, consultez la section [Supprimer une pile sur la AWS CloudFormation console](#) dans le Guide de l'utilisateur AWS CloudFormation.
- AWS CLI: Pour supprimer une application à l'aide de l'AWS CLI, consultez la section [Supprimer une pile](#) dans le Guide de l'utilisateur AWS CloudFormation.

Mise à jour des applications

Après avoir déployé une application à partir de l'AWS Serverless Application Repository, vous pouvez la mettre à jour. Par exemple, vous pouvez modifier un paramètre d'application ou mettre à jour l'application vers la dernière version publiée.

Les sections suivantes décrivent comment déployer une nouvelle version d'une application à l'aide de la AWS Management Console ou de l'AWS CLI.

Mise à jour des applications (console)

Pour mettre à jour une application que vous avez précédemment déployée, utilisez la même procédure que pour le déploiement d'une nouvelle application et fournissez le même nom d'application que celui avec lequel vous l'avez initialement déployée. Plus particulièrement, l'AWS Serverless Application Repository ajoute le préfixe `serverlessrepo-` au nom de votre application. Toutefois, pour déployer une nouvelle version de votre application, vous fournissez le nom d'application d'origine sans `serverlessrepo-` en préfixe.

Par exemple, si vous avez déployé une application portant le nom `MyApplication`, le nom de la pile est `serverlessrepo-MyApplication`. Pour mettre à jour cette application, vous devez fournir à `MyApplication` le nouveau nom. Ne spécifiez pas le nom complet de la pile `serverlessrepo-MyApplication`.

Pour tous les autres paramètres d'application, vous pouvez conserver les mêmes valeurs que le déploiement précédent ou fournir de nouvelles valeurs.

Mise à jour des applications (AWS CLI)

Pour mettre à jour une application que vous avez précédemment déployée, utilisez la même procédure que le déploiement d'une nouvelle application et fournissez la même `--stack-name` que celle avec laquelle vous l'avez initialement déployée. Plus particulièrement, AWS Serverless Application Repository ajoute le préfixe `serverlessrepo-` à votre nom de pile. Toutefois, pour déployer une nouvelle version de votre application, vous fournissez le nom de la pile d'origine sans ajouter le préfixe `serverlessrepo-`.

Par exemple, si vous avez déployé une application avec le nom de la pile `MyApplication`, le nom de la pile créée est `serverlessrepo-MyApplication`. Pour mettre à jour cette application, vous devez fournir à `MyApplication` nouveau le nom. Ne spécifiez pas le nom complet de la pile `deserverlessrepo-MyApplication`.

Sécurité dans le AWS Serverless Application Repository

Chez AWS, la sécurité dans le cloud est notre priorité numéro 1. En tant que client AWS, vous bénéficiez d'un centre de données et d'une architecture réseau conçus pour répondre aux exigences des organisations les plus pointilleuses en termes de sécurité.

La sécurité est une responsabilité partagée entre AWS et vous-même. Le [modèle de responsabilité partagée](#) décrit cette notion par les termes sécurité du cloud et sécurité dans le cloud :

- Sécurité du cloud – AWS est responsable de la protection de l'infrastructure qui exécute des services AWS dans le cloud AWS. AWS vous fournit également les services que vous pouvez utiliser en toute sécurité. Des auditeurs tiers testent et vérifient régulièrement l'efficacité de notre sécurité dans le cadre des [programmes de conformité AWS](#). Pour de plus amples informations sur les programmes de conformité qui s'appliquent à AWS Serverless Application Repository, veuillez consulter [Services AWS concernés par le programme de conformité](#).
- Sécurité dans le cloud : votre responsabilité est déterminée par le service AWS que vous utilisez. Vous êtes également responsable d'autres facteurs, y compris la sensibilité de vos données, les exigences de votre entreprise, la législation et la réglementation applicables.

Cette documentation vous aide à comprendre comment appliquer le modèle de responsabilité partagée lorsque vous utilisez AWS Serverless Application Repository. Les rubriques suivantes vous montrent comment configurer l'AWS Serverless Application Repository pour qu'elle réponde à vos objectifs de sécurité et de conformité. Vous apprendrez également à utiliser d'autres services AWS pour surveiller et sécuriser vos ressources AWS Serverless Application Repository.

Rubriques

- [Protection des données dans le AWS Serverless Application Repository](#)
- [Identity and Access Management pour AWS Serverless Application Repository](#)
- [Journalisation et surveillance dans AWS Serverless Application Repository](#)
- [Validation de la conformité pour l'AWS Serverless Application Repository](#)
- [Résilience dans la gestion de configuration d'AWS Serverless Application Repository](#)
- [Sécurité de l'infrastructure dans le AWS Serverless Application Repository](#)

Protection des données dans le AWS Serverless Application Repository

Le AWS modèle de [responsabilité partagée modèle](#) s'applique à la protection des données dans AWS Serverless Application Repository. Comme décrit dans ce modèle, AWS est chargé de protéger l'infrastructure mondiale qui gère tous les AWS Cloud. Vous êtes responsable du contrôle de votre contenu hébergé sur cette infrastructure. Vous êtes également responsable des tâches de configuration et de gestion de la sécurité pour Services AWS que vous utilisez. Pour plus d'informations sur la confidentialité des données, consultez la section [Confidentialité des données FAQ](#). Pour plus d'informations sur la protection des données en Europe, consultez le [AWS Modèle de responsabilité partagée et article de GDPR](#) blog sur le AWS Blog sur la sécurité.

Pour des raisons de protection des données, nous vous recommandons de protéger Compte AWS informations d'identification et configuration des utilisateurs individuels avec AWS IAM Identity Center or AWS Identity and Access Management (IAM). Ainsi, chaque utilisateur se voit attribuer uniquement les autorisations nécessaires pour exécuter ses tâches. Nous vous recommandons également de sécuriser vos données comme indiqué ci-dessous :

- Utilisez l'authentification multifactorielle (MFA) pour chaque compte.
- Utilisez SSL/TLS pour communiquer avec AWS ressources. Nous avons besoin de la TLS version 1.2 et recommandons la TLS version 1.3.
- Configuration API et enregistrement des activités des utilisateurs avec AWS CloudTrail. Pour plus d'informations sur l'utilisation CloudTrail des sentiers pour capturer AWS activités, voir [Travailler avec les CloudTrail sentiers](#) dans le AWS CloudTrail Guide de l'utilisateur.
- Utiliser AWS solutions de chiffrement, ainsi que tous les contrôles de sécurité par défaut intégrés Services AWS.
- Utilisez des services de sécurité gérés avancés tels qu'Amazon Macie, qui contribuent à la découverte et à la sécurisation des données sensibles stockées dans Amazon S3.
- Si vous avez besoin de FIPS 140 à 3 modules cryptographiques validés pour accéder AWS via une interface de ligne de commande ou un API, utilisez un FIPS point de terminaison. Pour plus d'informations sur les FIPS points de terminaison disponibles, voir [Federal Information Processing Standard \(FIPS\) 140-3](#).

Nous vous recommandons fortement de ne jamais placer d'informations confidentielles ou sensibles, telles que les adresses e-mail de vos clients, dans des balises ou des champs de texte libre tels

que le champ Name (Nom). Cela inclut lorsque vous travaillez avec AWS Serverless Application Repository ou autre Services AWS à l'aide de la consoleAPI, AWS CLI, ou AWS SDKs. Toutes les données que vous entrez dans des balises ou des champs de texte de forme libre utilisés pour les noms peuvent être utilisées à des fins de facturation ou dans les journaux de diagnostic. Si vous fournissez un URL à un serveur externe, nous vous recommandons vivement de ne pas y inclure d'informations d'identification URL pour valider votre demande auprès de ce serveur.

Chiffrement en transit

AWS Serverless Application Repository APIles points de terminaison ne prennent en charge que les connexions sécurisées. HTTPS Lorsque vous gérez AWS Serverless Application Repository ressources avec le AWS Management Console, AWS SDK, ou le AWS Serverless Application Repository API, toutes les communications sont cryptées avec Transport Layer Security (TLS).

Pour une liste complète des API points de terminaison, voir [AWS Régions et points de terminaison](#) dans le Références générales AWS.

Chiffrement au repos

Le AWS Serverless Application Repository chiffre les fichiers que vous téléchargez sur le AWS Serverless Application Repository, y compris les packages de déploiement et les archives de couches.

Identity and Access Management pour AWS Serverless Application Repository

AWS Identity and Access Management (IAM) est un Service AWS qui permet à un administrateur de contrôler en toute sécurité l'accès à AWS ressources. IAMles administrateurs contrôlent qui peut être authentifié (connecté) et autorisé (autorisé) à utiliser AWS Serverless Application Repository ressources. IAMest un Service AWS que vous pouvez utiliser sans frais supplémentaires.

Pour obtenir un aperçu du IAM fonctionnement, voir [Comprendre le IAM fonctionnement](#) dans le guide de l'IAMutilisateur.

Rubriques

- [Public ciblé](#)
- [Authentification avec des identités](#)

- [Gestion des accès à l'aide de politiques](#)
- [Comment AWS Serverless Application Repository fonctionne-t-il avec IAM](#)
- [Exemples de politiques AWS Serverless Application Repository basées sur l'identité](#)
- [AWS Serverless Application Repository Exemples de politiques d'application](#)
- [AWS Serverless Application Repository Autorisations d'API : Référence des actions et ressources](#)
- [Résolution des problèmes AWS Serverless Application Repository d'identité et d'accès](#)

Public ciblé

Comment utilisez-vous AWS Identity and Access Management (IAM) diffère en fonction du travail que vous effectuez dans AWS Serverless Application Repository.

Utilisateur du service — Si vous utilisez le AWS Serverless Application Repository service pour faire votre travail, puis votre administrateur vous fournit les informations d'identification et les autorisations dont vous avez besoin. Au fur et à mesure que vous en utilisez AWS Serverless Application Repository fonctionnalités pour effectuer votre travail, vous aurez peut-être besoin d'autorisations supplémentaires. En comprenant bien la gestion des accès, vous saurez demander les autorisations appropriées à votre administrateur. Si vous ne parvenez pas à accéder à une fonctionnalité dans AWS Serverless Application Repository, voir [Résolution des problèmes AWS Serverless Application Repository d'identité et d'accès](#).

Administrateur du service — Si vous êtes responsable de AWS Serverless Application Repository ressources de votre entreprise, vous avez probablement un accès complet à AWS Serverless Application Repository. C'est à vous de déterminer lequel AWS Serverless Application Repository fonctionnalités et ressources auxquelles les utilisateurs de vos services devraient avoir accès. Vous devez ensuite envoyer des demandes à votre IAM administrateur pour modifier les autorisations des utilisateurs de votre service. Consultez les informations de cette page pour comprendre les concepts de base de IAM. Pour en savoir plus sur la façon dont votre entreprise peut utiliser IAM AWS Serverless Application Repository, voir [Comment AWS Serverless Application Repository fonctionne-t-il avec IAM](#).

IAM administrateur — Si vous êtes IAM administrateur, vous souhaitez peut-être en savoir plus sur la manière dont vous pouvez rédiger des politiques pour gérer l'accès à AWS Serverless Application Repository. Pour voir un exemple AWS Serverless Application Repository politiques basées sur l'identité que vous pouvez utiliser dans IAM, voir. [Exemples de politiques AWS Serverless Application Repository basées sur l'identité](#)

Authentification avec des identités

L'authentification est la façon dont vous vous connectez à AWS en utilisant vos informations d'identification. Vous devez être authentifié (connecté) à AWS) en tant que Utilisateur racine d'un compte AWS, en tant qu'IAMutilisateur ou en assumant un IAM rôle.

Vous pouvez vous connecter à AWS en tant qu'identité fédérée en utilisant les informations d'identification fournies par le biais d'une source d'identité. AWS IAM Identity Center Les utilisateurs (IAMIdentity Center), l'authentification unique de votre entreprise et vos informations d'identification Google ou Facebook sont des exemples d'identités fédérées. Lorsque vous vous connectez en tant qu'identité fédérée, votre administrateur a préalablement configuré la fédération d'identité à l'aide de IAM rôles. Lorsque vous accédez AWS en utilisant la fédération, vous assumez indirectement un rôle.

Selon le type d'utilisateur que vous êtes, vous pouvez vous connecter au AWS Management Console ou le AWS portail d'accès. Pour plus d'informations sur la connexion à AWS, voir [Comment se connecter à votre Compte AWS](#) dans le .Connexion à AWS Guide de l'utilisateur.

Si vous accédez AWS programmatiquement, AWS fournit un kit de développement logiciel (SDK) et une interface de ligne de commande (CLI) pour signer cryptographiquement vos demandes à l'aide de vos informations d'identification. Si vous n'utilisez pas AWS outils, vous devez signer vous-même les demandes. Pour plus d'informations sur l'utilisation de la méthode recommandée pour signer vous-même les demandes, voir [Signature AWS API demandes](#) dans le guide de IAM l'utilisateur.

Quelle que soit la méthode d'authentification que vous utilisez, vous devrez peut-être fournir des informations de sécurité supplémentaires. Par exemple, AWS vous recommande d'utiliser l'authentification multifactorielle (MFA) pour renforcer la sécurité de votre compte. Pour en savoir plus, consultez la section [Authentification multifactorielle](#) dans le AWS IAM Identity Center Guide de l'utilisateur et [utilisation de l'authentification multifactorielle \(MFA\) dans AWS](#) dans le guide de l'utilisateur IAM.

Compte AWS utilisateur root

Lorsque vous créez un Compte AWS, vous commencez avec une seule identité de connexion qui donne un accès complet à tous Services AWS et les ressources du compte. Cette identité s'appelle Compte AWS utilisateur root et est accessible en vous connectant avec l'adresse e-mail et le mot de passe que vous avez utilisés pour créer le compte. Il est vivement recommandé de ne pas utiliser l'utilisateur racine pour vos tâches quotidiennes. Protégez vos informations d'identification d'utilisateur racine et utilisez-les pour effectuer les tâches que seul l'utilisateur racine peut effectuer. Pour obtenir la liste complète des tâches qui nécessitent que vous vous connectiez

en tant qu'utilisateur root, consultez la section [Tâches nécessitant des informations d'identification utilisateur root](#) dans le guide de IAM l'utilisateur.

Utilisateurs et groupes IAM

Un [IAMutilisateur](#) est une identité au sein de votre Compte AWS qui dispose d'autorisations spécifiques pour une seule personne ou une seule application. Dans la mesure du possible, nous vous recommandons de vous appuyer sur des informations d'identification temporaires plutôt que de créer des IAM utilisateurs dotés d'informations d'identification à long terme, telles que des mots de passe et des clés d'accès. Toutefois, si vous avez des cas d'utilisation spécifiques qui nécessitent des informations d'identification à long terme auprès des IAM utilisateurs, nous vous recommandons de faire pivoter les clés d'accès. Pour plus d'informations, voir [Rotation régulière des clés d'accès pour les cas d'utilisation nécessitant des informations d'identification à long terme](#) dans le Guide de IAM l'utilisateur.

Un [IAMgroupe](#) est une identité qui définit un ensemble d'IAMutilisateurs. Vous ne pouvez pas vous connecter en tant que groupe. Vous pouvez utiliser les groupes pour spécifier des autorisations pour plusieurs utilisateurs à la fois. Les groupes permettent de gérer plus facilement les autorisations pour de grands ensembles d'utilisateurs. Par exemple, vous pouvez nommer un groupe IAMAdminset lui donner les autorisations nécessaires pour administrer IAM des ressources.

Les utilisateurs sont différents des rôles. Un utilisateur est associé de manière unique à une personne ou une application, alors qu'un rôle est conçu pour être endossé par tout utilisateur qui en a besoin. Les utilisateurs disposent d'informations d'identification permanentes, mais les rôles fournissent des informations d'identification temporaires. Pour en savoir plus, voir [Quand créer un IAM utilisateur \(au lieu d'un rôle\)](#) dans le Guide de IAM l'utilisateur.

IAMRôles

Un [IAMrôle](#) est une identité au sein de votre Compte AWS qui dispose d'autorisations spécifiques. Il est similaire à un IAM utilisateur, mais n'est pas associé à une personne en particulier. Vous pouvez assumer temporairement un IAM rôle dans AWS Management Console en [changeant de rôle](#). Vous pouvez assumer un rôle en appelant un AWS CLI or AWS APIopération ou en utilisant une option personnaliséeURL. Pour plus d'informations sur les méthodes d'utilisation des rôles, consultez la section [Utilisation IAM des rôles](#) dans le Guide de IAM l'utilisateur.

IAMles rôles dotés d'informations d'identification temporaires sont utiles dans les situations suivantes :

- **Accès utilisateur fédéré** : pour attribuer des autorisations à une identité fédérée, vous créez un rôle et définissez des autorisations pour le rôle. Quand une identité externe s'authentifie, l'identité est associée au rôle et reçoit les autorisations qui sont définies par celui-ci. Pour plus d'informations sur les rôles pour la fédération, voir [Création d'un rôle pour un fournisseur d'identité tiers](#) dans le guide de IAM l'utilisateur. Si vous utilisez IAM Identity Center, vous configurez un ensemble d'autorisations. Pour contrôler les accès auxquels vos identités peuvent accéder après leur authentification, IAM Identity Center met en corrélation l'ensemble d'autorisations avec un rôle dans IAM. Pour plus d'informations sur les ensembles d'autorisations, voir [Ensembles d'autorisations](#) dans le AWS IAM Identity Center Guide de l'utilisateur.
- **Autorisations IAM utilisateur temporaires** : un IAM utilisateur ou un rôle peut assumer un IAM rôle afin d'obtenir temporairement différentes autorisations pour une tâche spécifique.
- **Accès entre comptes** : vous pouvez utiliser un IAM rôle pour autoriser une personne (un mandant fiable) d'un autre compte à accéder aux ressources de votre compte. Les rôles constituent le principal moyen d'accorder l'accès intercompte. Cependant, avec certains Services AWS, vous pouvez associer une politique directement à une ressource (au lieu d'utiliser un rôle comme proxy). Pour connaître la différence entre les rôles et les politiques basées sur les ressources pour l'accès entre comptes, voir [Accès aux ressources entre comptes IAM dans le guide](#) de l'IAM utilisateur.
- **Accès multiservices** — Certains Services AWS utilisent des fonctionnalités dans d'autres Services AWS. Par exemple, lorsque vous effectuez un appel dans un service, il est courant que ce service exécute des applications dans Amazon EC2 ou stocke des objets dans Amazon S3. Un service peut le faire en utilisant les autorisations d'appel du principal, un rôle de service ou un rôle lié au service.
- **Sessions d'accès transmises (FAS)** : lorsque vous utilisez un IAM utilisateur ou un rôle pour effectuer des actions dans AWS, vous êtes considéré comme un directeur. Lorsque vous utilisez certains services, vous pouvez effectuer une action qui initie une autre action dans un autre service. FAS utilise les autorisations du principal appelant un Service AWS, combiné à la demande Service AWS pour adresser des demandes aux services en aval. FAS les demandes ne sont effectuées que lorsqu'un service reçoit une demande nécessitant des interactions avec d'autres Services AWS ou des ressources à compléter. Dans ce cas, vous devez disposer d'autorisations nécessaires pour effectuer les deux actions. Pour plus de détails sur les politiques relatives FAS aux demandes, consultez la section [Transférer les sessions d'accès](#).
- **Rôle de service** — Un rôle de service est un [IAM rôle](#) qu'un service assume pour effectuer des actions en votre nom. Un IAM administrateur peut créer, modifier et supprimer un rôle de service de l'intérieur IAM. Pour plus d'informations, voir [Création d'un rôle pour déléguer des autorisations à un Service AWS](#) dans le guide de l'utilisateur IAM.

- **Rôle lié à un service** — Un rôle lié à un service est un type de rôle lié à un service Service AWS. Le service peut assumer le rôle d'effectuer une action en votre nom. Les rôles liés à un service apparaissent dans votre Compte AWS et appartiennent au service. Un IAM administrateur peut consulter, mais pas modifier les autorisations pour les rôles liés à un service.
- **Applications exécutées sur Amazon EC2** : vous pouvez utiliser un IAM rôle pour gérer les informations d'identification temporaires pour les applications qui s'exécutent sur une EC2 instance et qui créent AWS CLI or AWS API demandes. Cela est préférable au stockage des clés d'accès dans l'EC2 instance. Pour attribuer un AWS pour attribuer un rôle à une EC2 instance et le mettre à la disposition de toutes ses applications, vous créez un profil d'instance attaché à l'instance. Un profil d'instance contient le rôle et permet aux programmes exécutés sur l'EC2 instance d'obtenir des informations d'identification temporaires. Pour plus d'informations, consultez la section [Utilisation d'un IAM rôle pour accorder des autorisations aux applications exécutées sur des EC2 instances Amazon](#) dans le Guide de IAM l'utilisateur.

Pour savoir s'il faut utiliser IAM des rôles ou des IAM utilisateurs, voir [Quand créer un IAM rôle \(au lieu d'un utilisateur\)](#) dans le guide de IAM l'utilisateur.

Gestion des accès à l'aide de politiques

Vous contrôlez l'accès dans AWS en créant des politiques et en les associant à AWS identités ou ressources. Une politique est un objet dans AWS qui, lorsqu'elle est associée à une identité ou à une ressource, définit leurs autorisations. AWS évalue ces politiques lorsqu'un principal (utilisateur, utilisateur root ou session de rôle) fait une demande. Les autorisations dans les politiques déterminent si la demande est autorisée ou refusée. La plupart des politiques sont stockées dans AWS sous forme de JSON documents. Pour plus d'informations sur la structure et le contenu des documents de JSON politique, voir [Présentation des JSON politiques](#) dans le guide de IAM l'utilisateur.

Les administrateurs peuvent utiliser AWS JSON politiques pour spécifier qui a accès à quoi. C'est-à-dire, quel principal peut effectuer des actions sur quelles ressources et dans quelles conditions.

Par défaut, les utilisateurs et les rôles ne disposent d'aucune autorisation. Pour autoriser les utilisateurs à effectuer des actions sur les ressources dont ils ont besoin, un IAM administrateur peut créer des IAM politiques. L'administrateur peut ensuite ajouter les IAM politiques aux rôles, et les utilisateurs peuvent assumer les rôles.

IAM les politiques définissent les autorisations pour une action, quelle que soit la méthode que vous utilisez pour effectuer l'opération. Par exemple, supposons que vous disposiez d'une politique qui

autorise l'action `iam:GetRole`. Un utilisateur appliquant cette politique peut obtenir des informations sur le rôle auprès du AWS Management Console, le AWS CLI, ou le AWS API.

politiques basées sur l'identité

Les politiques basées sur l'identité sont JSON des documents de politique d'autorisation que vous pouvez joindre à une identité, telle qu'un IAM utilisateur, un groupe d'utilisateurs ou un rôle. Ces politiques contrôlent quel type d'actions des utilisateurs et des rôles peuvent exécuter, sur quelles ressources et dans quelles conditions. Pour savoir comment créer une politique basée sur l'identité, consultez la section [Création de IAM politiques](#) dans le Guide de l'IAMutilisateur.

Les politiques basées sur l'identité peuvent être classées comme des politiques en ligne ou des politiques gérées. Les politiques en ligne sont intégrées directement à un utilisateur, groupe ou rôle. Les politiques gérées sont des politiques autonomes que vous pouvez associer à plusieurs utilisateurs, groupes et rôles dans votre Compte AWS. Les politiques gérées incluent AWS politiques gérées et politiques gérées par le client. Pour savoir comment choisir entre une politique gérée ou une politique intégrée, voir [Choisir entre des politiques gérées et des politiques intégrées dans le Guide](#) de l'IAMutilisateur.

Politiques basées sur une ressource

Les politiques basées sur les ressources sont des documents JSON de stratégie que vous attachez à une ressource. Les politiques de confiance dans les IAM rôles et les politiques relatives aux compartiments Amazon S3 sont des exemples de politiques basées sur les ressources. Dans les services qui sont compatibles avec les politiques basées sur les ressources, les administrateurs de service peuvent les utiliser pour contrôler l'accès à une ressource spécifique. Pour la ressource dans laquelle se trouve la politique, cette dernière définit quel type d'actions un principal spécifié peut effectuer sur cette ressource et dans quelles conditions. Vous devez [spécifier un principal](#) dans une politique basée sur les ressources. Les principaux peuvent inclure des comptes, des utilisateurs, des rôles, des utilisateurs fédérés ou Services AWS.

Les politiques basées sur les ressources sont des politiques en ligne situées dans ce service. Vous ne pouvez pas utiliser AWS politiques gérées à partir IAM d'une stratégie basée sur les ressources.

Listes de contrôle d'accès (ACLs)

Les listes de contrôle d'accès (ACLs) contrôlent les principaux (membres du compte, utilisateurs ou rôles) autorisés à accéder à une ressource. ACLs sont similaires aux politiques basées sur les ressources, bien qu'elles n'utilisent pas le format du document JSON de stratégie.

Amazon S3, AWS WAF, et Amazon VPC sont des exemples de services qui prennent en charge ACLs. Pour en savoir plus sur ACLs, consultez la [présentation de la liste de contrôle d'accès \(ACL\)](#) dans le guide du développeur Amazon Simple Storage Service.

Autres types de politique

AWS prend en charge d'autres types de politiques moins courants. Ces types de politiques peuvent définir le nombre maximum d'autorisations qui vous sont accordées par des types de politiques plus courants.

- **Limites d'autorisations** — Une limite d'autorisations est une fonctionnalité avancée dans laquelle vous définissez le maximum d'autorisations qu'une politique basée sur l'identité peut accorder à une IAM entité (IAM utilisateur ou rôle). Vous pouvez définir une limite d'autorisations pour une entité. Les autorisations en résultant représentent la combinaison des politiques basées sur l'identité d'une entité et de ses limites d'autorisations. Les politiques basées sur les ressources qui spécifient l'utilisateur ou le rôle dans le champ `Principal` ne sont pas limitées par les limites d'autorisations. Un refus explicite dans l'une de ces politiques annule l'autorisation. Pour plus d'informations sur les limites d'autorisations, voir [Limites d'autorisations pour les IAM entités](#) dans le Guide de IAM l'utilisateur.
- **Politiques de contrôle des services (SCPs)** : SCPs JSON politiques qui spécifient les autorisations maximales pour une organisation ou une unité organisationnelle (UO) dans AWS Organizations. AWS Organizations est un service de regroupement et de gestion centralisée de plusieurs Comptes AWS que votre entreprise possède. Si vous activez toutes les fonctionnalités d'une organisation, vous pouvez appliquer des politiques de contrôle des services (SCPs) à l'un ou à l'ensemble de vos comptes. Les SCP limites d'autorisations pour les entités figurant dans les comptes des membres, y compris chaque Utilisateur racine d'un compte AWS. Pour plus d'informations sur les Organizations et SCPs consultez les [politiques de contrôle des services](#) dans le AWS Organizations Guide de l'utilisateur.
- **Politiques de séance** : les politiques de séance sont des politiques avancées que vous utilisez en tant que paramètre lorsque vous créez par programmation une séance temporaire pour un rôle ou un utilisateur fédéré. Les autorisations de séance en résultant sont une combinaison des politiques basées sur l'identité de l'utilisateur ou du rôle et des politiques de séance. Les autorisations peuvent également provenir d'une politique basée sur les ressources. Un refus explicite dans l'une de ces politiques annule l'autorisation. Pour plus d'informations, consultez la section [Politiques de session](#) dans le guide de IAM l'utilisateur.

Types de politique multiple

Lorsque plusieurs types de politiques s'appliquent à la requête, les autorisations en résultant sont plus compliquées à comprendre. Pour savoir comment AWS détermine s'il faut autoriser une demande lorsque plusieurs types de politiques sont impliqués, voir la [logique d'évaluation des politiques](#) dans le guide de IAM l'utilisateur.

Comment AWS Serverless Application Repository fonctionne-t-il avec IAM

Avant IAM de gérer l'accès au AWS Serverless Application Repository, vous devez connaître les IAM fonctionnalités disponibles avec le AWS Serverless Application Repository.

Pour obtenir un aperçu du IAM fonctionnement, consultez la section [Comprendre le IAM fonctionnement](#) dans le guide de IAM l'utilisateur. Pour obtenir une vue d'ensemble du fonctionnement de ces services AWS Serverless Application Repository et AWS des autres services IAM, consultez la section [AWS Services qui fonctionnent avec IAM](#) dans le guide de IAM l'utilisateur.

Rubriques

- [Politiques AWS Serverless Application Repository basées sur l'identité](#)
- [AWS Serverless Application Repository Politiques relatives aux applications](#)
- [Autorisation basée sur les balises AWS Serverless Application Repository](#)
- [AWS Serverless Application Repository IAM Rôles](#)

Politiques AWS Serverless Application Repository basées sur l'identité

Avec les politiques IAM basées sur l'identité, vous pouvez spécifier les actions et les ressources autorisées ou refusées, ainsi que les conditions dans lesquelles les actions sont autorisées ou refusées. La gestion de configuration d' AWS Serverless Application Repository prend en charge des actions, ressources et clés de condition spécifiques. Pour en savoir plus sur tous les éléments que vous utilisez dans une JSON politique, consultez la section [Référence des éléments de IAM JSON stratégie](#) dans le guide de IAM l'utilisateur.

Un exemple de politique d'autorisation est exposé ci-dessous.

```
{  
  
  "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"
  }
]
```

La politique possède deux énoncés:

- La première instruction accorde des autorisations pour l' AWS Serverless Application Repository action `serverlessrepo:CreateApplication` sur toutes les AWS Serverless Application Repository ressources, comme indiqué par le caractère générique (*) comme Resource valeur.
- La deuxième instruction autorise l' AWS Serverless Application Repository action `serverlessrepo:CreateApplicationVersion` sur une AWS ressource en utilisant le nom de ressource Amazon (ARN) pour une AWS Serverless Application Repository application. L'application est spécifiée par la valeur Resource.

La stratégie ne spécifie pas l'élément `Principal` car, dans une stratégie basée sur une identité, vous ne spécifiez pas le mandataire qui obtient l'autorisation. Quand vous attachez une politique à un utilisateur, l'utilisateur est le principal implicite. Lorsque vous associez une politique d'autorisation à un IAM rôle, le principal identifié dans la politique de confiance du rôle obtient les autorisations.

Pour un tableau présentant toutes les AWS Serverless Application Repository API opérations et les AWS ressources auxquelles elles s'appliquent, voir [AWS Serverless Application Repository Autorisations d'API : Référence des actions et ressources](#).

Actions

Les administrateurs peuvent utiliser AWS JSON des politiques pour spécifier qui a accès à quoi. C'est-à-dire, quel principal peut effectuer des actions sur quelles ressources et dans quelles conditions.

L'Action élément d'une JSON politique décrit les actions que vous pouvez utiliser pour autoriser ou refuser l'accès dans une politique. Les actions de stratégie portent généralement le même nom que l' AWS API opération associée. Il existe certaines exceptions, telles que les actions avec autorisation uniquement qui n'ont pas d'opération correspondante. API Certaines opérations nécessitent également plusieurs actions dans une politique. Ces actions supplémentaires sont nommées actions dépendantes.

Intégration d'actions dans une stratégie afin d'accorder l'autorisation d'exécuter les opérations associées.

Les actions de politique AWS Serverless Application Repository utilisent le préfixe suivant avant l'action : `serverlessrepo:`. Par exemple, pour autoriser quelqu'un à exécuter une AWS Serverless Application Repository instance avec l' AWS Serverless Application Repository `SearchApplicationsAPI` opération, vous devez inclure `serverlessrepo:SearchApplicationsaction` dans sa politique. Les déclarations de politique doivent inclure un élément `Action` ou `NotAction`. AWS Serverless Application Repository définit son propre ensemble d'actions décrivant les tâches que vous pouvez effectuer avec ce service.

Pour spécifier plusieurs actions dans une seule déclaration, séparez-les par des virgules comme suit :

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

Vous pouvez aussi spécifier plusieurs actions à l'aide de caractères génériques (*). Par exemple, pour spécifier toutes les actions qui commencent par le mot `List`, incluez l'action suivante :

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

Pour consulter la liste des AWS Serverless Application Repository actions, reportez-vous à la section [Actions définies par AWS Serverless Application Repository](#) dans le guide de IAM l'utilisateur.

Ressources

Les administrateurs peuvent utiliser AWS JSON des politiques pour spécifier qui a accès à quoi. C'est-à-dire, quel principal peut effectuer des actions sur quelles ressources et dans quelles conditions.

L'élément `Resource` JSON de stratégie indique le ou les objets auxquels s'applique l'action. Les instructions doivent inclure un élément `Resource` ou `NotResource`. Il est recommandé de spécifier une ressource en utilisant son [Amazon Resource Name \(ARN\)](#). Vous pouvez le faire pour des actions qui prennent en charge un type de ressource spécifique, connu sous la dénomination autorisations de niveau ressource.

Pour les actions qui ne sont pas compatibles avec les autorisations de niveau ressource, telles que les opérations de liste, utilisez un caractère générique (*) afin d'indiquer que l'instruction s'applique à toutes les ressources.

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

Dans le AWS Serverless Application Repository, la AWS ressource principale est une AWS Serverless Application Repository application. AWS Serverless Application Repository les applications sont associées à des noms de ressources Amazon uniques (ARNs), comme indiqué dans le tableau suivant.

AWS Type de ressource	Format du nom de ressource Amazon (ARN)
Application	arn: <i>partition</i> :serverlessrepo: <i>region</i> : <i>account-id</i> :applications/ <i>application-name</i>

Pour plus d'informations sur le format de ARNs, consultez [Amazon Resource Names \(ARNs\) et AWS Service Namespaces](#).

Voici un exemple de politique qui accorde des autorisations pour l'`serverlessrepo:ListApplications` action sur toutes les AWS ressources. Dans l'implémentation actuelle, AWS Serverless Application Repository il n'est pas possible d'identifier AWS des ressources spécifiques en utilisant la AWS ressource ARNs (également appelée autorisations au niveau de la API ressource) pour certaines actions. Dans ce cas, vous devez spécifier un caractère générique (*).

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

Pour un tableau présentant toutes les AWS Serverless Application Repository API actions et les AWS ressources auxquelles elles s'appliquent, voir [AWS Serverless Application Repository Autorisations d'API : Référence des actions et ressources](#).

Clés de condition

Il AWS Serverless Application Repository ne fournit aucune clé de condition spécifique au service, mais il prend en charge l'utilisation de certaines clés de condition globales. Pour voir toutes les clés de condition AWS globales, voir [Clés contextuelles de condition AWS globale](#) dans le guide de IAM l'utilisateur.

Exemples

Pour consulter des exemples de politiques AWS Serverless Application Repository basées sur l'identité, consultez. [Exemples de politiques AWS Serverless Application Repository basées sur l'identité](#)

AWS Serverless Application Repository Politiques relatives aux applications

Les politiques d'application déterminent les actions qu'un principal spécifié principalOrg peut ou peut effectuer sur une AWS Serverless Application Repository application.

Vous pouvez ajouter des autorisations à la politique associée à une AWS Serverless Application Repository application. Les politiques d'autorisation associées aux AWS Serverless Application Repository applications sont appelées politiques d'application. Les [politiques d'application](#) sont des extensions des politiques [IAM basées sur les ressources](#). La ressource principale est l' AWS

Serverless Application Repository application. Vous pouvez utiliser les politiques des AWS Serverless Application Repository applications pour gérer les autorisations de déploiement des applications.

AWS Serverless Application Repository les politiques relatives aux applications sont principalement utilisées par les éditeurs pour autoriser les consommateurs à déployer leurs applications, ainsi que pour effectuer des opérations connexes, telles que la recherche et l'affichage des détails de ces applications. Les éditeurs peuvent définir les autorisations d'application selon les trois catégories suivantes :

- Privé : applications créées avec le même compte et qui n'ont été partagées avec aucun autre compte. Vous êtes autorisé à déployer des applications créées à l'aide de votre AWS compte.
- Partage privé : applications que l'éditeur a explicitement partagées avec un ensemble spécifique de AWS comptes ou d' AWS Organisations. Vous êtes autorisé à déployer des applications partagées avec votre AWS compte ou votre AWS organisation.
- Partage public : applications que l'éditeur a partagées avec tout le monde. Vous avez l'autorisation de déployer n'importe quelle application partagée publiquement.

Vous pouvez accorder des autorisations en utilisant le AWS CLI AWS SDKs, le ou le AWS Management Console.

Exemples

Pour consulter des exemples de gestion des politiques AWS Serverless Application Repository d'application, consultez [AWS Serverless Application Repository Exemples de politiques d'application](#).

Autorisation basée sur les balises AWS Serverless Application Repository

Il AWS Serverless Application Repository ne prend pas en charge le contrôle de l'accès aux ressources ou aux actions en fonction des balises.

AWS Serverless Application Repository IAM Rôles

Un [IAM rôle](#) est une entité de votre AWS compte qui possède des autorisations spécifiques.

Utilisation d'informations d'identification temporaires avec AWS Serverless Application Repository

Vous pouvez utiliser des informations d'identification temporaires pour vous connecter à la fédération, pour assumer un IAM rôle ou pour assumer un rôle entre comptes. Vous obtenez des informations d'identification de sécurité temporaires en appelant AWS STS API des opérations telles que [AssumeRole](#) ou [GetFederationToken](#).

Les AWS Serverless Application Repository supports utilisant des informations d'identification temporaires.

Rôles liés à un service

Ne AWS Serverless Application Repository prend pas en charge les rôles liés à un service.

Rôles de service

Ne AWS Serverless Application Repository prend pas en charge les rôles de service.

Exemples de politiques AWS Serverless Application Repository basées sur l'identité

Par défaut, les utilisateurs et les rôles IAM ne sont pas autorisés à créer ou modifier les ressources AWS Serverless Application Repository. Ils ne peuvent pas non plus exécuter des tâches à l'aide de AWS Management Console, AWS CLI ou de l'API AWS. Un administrateur IAM doit créer des politiques IAM autorisant les utilisateurs et les rôles à exécuter des opérations d'API spécifiques sur les ressources spécifiées dont ils ont besoin. Il doit ensuite attacher ces politiques aux utilisateurs ou aux groupes IAM ayant besoin de ces autorisations.

Pour apprendre à créer une stratégie basée sur l'identité IAM à l'aide de ces exemples de documents de stratégie JSON, veuillez consulter [Création de stratégies dans l'onglet JSON](#) dans le IAM Guide de l'utilisateur.

Rubriques

- [Bonnes pratiques en matière de politiques](#)
- [Utilisation de la console AWS Serverless Application Repository](#)
- [Autoriser les utilisateurs à afficher leurs propres autorisations](#)
- [Exemples de politiques gérées par le client](#)

Bonnes pratiques en matière de politiques

Les politiques basées sur l'identité sont très puissantes. Elles déterminent si une personne peut créer, consulter ou supprimer des ressources AWS Serverless Application Repository dans votre compte. Ces actions peuvent entraîner des frais pour votre compte AWS. Lorsque vous créez ou modifiez des politiques basées sur l'identité, suivez ces instructions et recommandations :

- **Accorder le moindre privilège** : lorsque vous créez des politiques personnalisées, accordez uniquement les autorisations nécessaires à l'exécution d'une tâche. Commencez avec un ensemble d'autorisations minimum et accordez-en d'autres si nécessaire. Cette méthode est plus sûre que de commencer avec des autorisations trop permissives et d'essayer de les restreindre plus tard. Pour plus d'informations, consultez [Accorder le privilège le plus faible](#) dans le Guide de l'utilisateur IAM.
- **Activer la MFA pour les opérations confidentielles** : pour plus de sécurité, demandez aux utilisateurs IAM d'utiliser l'Authentification multifacteur (MFA) pour accéder à des ressources ou à des opérations d'API confidentielles. Pour de plus amples informations, veuillez consulter [Utilisation de l'Authentification multi-facteur \(MFA\) dans AWS](#) dans le guide de l'utilisateur IAM.
- **Utiliser des conditions de politique pour davantage de sécurité** : dans la mesure du possible, définissez les conditions dans lesquelles vos politiques basées sur l'identité autorisent l'accès à une ressource. Par exemple, vous pouvez rédiger les conditions pour spécifier une plage d'adresses IP autorisées d'où peut provenir une demande. Vous pouvez également écrire des conditions pour autoriser les requêtes uniquement à une date ou dans une plage de temps spécifiée, ou pour imposer l'utilisation de SSL ou de MFA. Pour plus d'informations, consultez [Éléments de politique JSON IAM : Condition](#) dans le Guide de l'utilisateur IAM.

Utilisation de la console AWS Serverless Application Repository

La console AWS Serverless Application Repository fournit un environnement intégré vous permettant de découvrir et de gérer les applications AWS Serverless Application Repository. La console offre de nombreuses fonctions et flux de travail qui exigent souvent des autorisations pour gérer une application AWS Serverless Application Repository en plus des autorisations propres à l'API documentées dans [AWS Serverless Application Repository Autorisations d'API : Référence des actions et ressources](#).

Pour de plus amples informations sur les autorisations qui sont nécessaires pour utiliser la console AWS Serverless Application Repository, veuillez consulter [Exemples de politiques gérées par le client](#).

Autoriser les utilisateurs à afficher leurs propres autorisations

Cet exemple montre comment créer une politique qui permet aux utilisateurs IAM d'afficher les politiques en ligne et gérées attachées à leur identité d'utilisateur. Cette politique inclut les autorisations nécessaires pour réaliser cette action sur la console ou par programmation à l'aide de l'AWS CLI ou de l'API AWS.

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

Exemples de politiques gérées par le client

Cette section fournit un ensemble d'exemples de stratégies que vous pouvez associer à un utilisateur. Si vous créez des stratégies pour la première fois, nous vous recommandons de commencer par créer un utilisateur IAM dans votre compte et de lui attacher les stratégies dans l'ordre. Vous pouvez également utiliser ces exemples pour créer une stratégie personnalisée unique qui inclut des autorisations pour effectuer plusieurs actions, puis l'attacher à l'utilisateur.

Pour plus d'informations sur la manière d'associer des politiques aux utilisateurs, consultez la section [Ajouter des autorisations à un utilisateur](#) dans le guide de l'utilisateur IAM.

Exemples

- [Exemple d'éditeur 1 : autoriser un éditeur à répertorier les applications](#)
- [Exemple d'éditeur 2 : autoriser un éditeur à afficher les détails d'une application ou d'une version de l'application](#)
- [Exemple d'éditeur 3 : autoriser un éditeur à créer une application ou une version de l'application](#)
- [Exemple d'éditeur 4 : autoriser un éditeur à créer une stratégie d'application pour partager des applications avec d'autres](#)
- [Exemple de consommateur 1 : autoriser un consommateur à rechercher des applications](#)
- [Exemple de consommateur 2 : autoriser un consommateur à afficher les détails d'une application](#)
- [Exemple de consommateur 3 : autoriser un consommateur à déployer une application](#)
- [Exemple 4 du consommateur : Refuser l'accès aux ressources de déploiement](#)
- [Exemple 5 du consommateur : empêcher un consommateur de rechercher et de déployer des applications publiques](#)

Exemple d'éditeur 1 : autoriser un éditeur à répertorier les applications

Un utilisateur IAM de votre compte doit disposer des autorisations pour l'opération `serverlessrepo:ListApplications` avant d'avoir accès au contenu de la console. Lorsque vous accordez ces autorisations, la console peut afficher la liste des AWS Serverless Application Repository applications du AWS compte créé dans la AWS région spécifique à laquelle appartient l'utilisateur.

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

```
}
```

Exemple d'éditeur 2 : autoriser un éditeur à afficher les détails d'une application ou d'une version de l'application

Un utilisateur peut sélectionner une application AWS Serverless Application Repository et en afficher les détails. Ces détails incluent l'auteur, la description, les versions et les autres informations de configuration. Pour ce faire, l'utilisateur a besoin d'autorisations pour les opérations d'API `serverlessrepo:GetApplication` et `serverlessrepo:ListApplicationVersions` pour AWS Serverless Application Repository.

Dans l'exemple suivant, ces autorisations sont accordées pour l'application spécifique dont l'ARN (Amazon Resource Name) est spécifié comme la valeur `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"
    }
  ]
}
```

Exemple d'éditeur 3 : autoriser un éditeur à créer une application ou une version de l'application

Si vous souhaitez autoriser un utilisateur à avoir des autorisations pour créer des applications AWS Serverless Application Repository, vous devez accorder des autorisations aux opérations `serverlessrepo:CreateApplication` et `serverlessrepo:CreateApplicationVersions`, comme illustré dans la stratégie suivante.

```
{
```

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

Exemple d'éditeur 4 : autoriser un éditeur à créer une stratégie d'application pour partager des applications avec d'autres

Pour que les utilisateurs partagent des applications avec d'autres, vous devez leur accorder des autorisations pour créer des stratégies d'application, comme illustré dans la stratégie suivante.

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

Exemple de consommateur 1 : autoriser un consommateur à rechercher des applications

Pour que les consommateurs recherchent des applications, vous devez leur accorder les autorisations suivantes.

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

Exemple de consommateur 2 : autoriser un consommateur à afficher les détails d'une application

Un utilisateur peut sélectionner une application AWS Serverless Application Repository et en afficher les détails, tels que l'auteur, la description, les versions et les autres informations de configuration. Pour ce faire, l'utilisateur doit disposer des autorisations pour les opérations AWS Serverless Application Repository suivantes.

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

Exemple de consommateur 3 : autoriser un consommateur à déployer une application

Pour que les clients déploient des applications, vous devez leur accorder des autorisations pour exécuter un certain nombre d'opérations. La stratégie suivante accorde aux clients les autorisations requises.

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

Note

Le déploiement d'une application peut nécessiter des autorisations pour utiliser des AWS ressources supplémentaires. Comme il AWS Serverless Application Repository utilise le même mécanisme de déploiement sous-jacent que AWS CloudFormation, voir [Contrôle de l'accès à l'aide de la AWS Identity and Access Management](#) pour plus d'informations. Vous pouvez également consulter [Dépannage : Autorisations IAM insuffisantes](#) pour obtenir de l'aide en cas de problèmes de déploiement liés aux autorisations.

Exemple 4 du consommateur : Refuser l'accès aux ressources de déploiement

Lorsqu'une application est partagée en privé avec un AWS compte, par défaut, tous les utilisateurs de ce compte peuvent accéder aux ressources de déploiement de tous les autres utilisateurs du même compte. La politique suivante empêche les utilisateurs d'un compte d'accéder aux ressources de

déploiement, qui sont stockées dans le compartiment Amazon S3 pour leAWS Serverless Application Repository.

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

Exemple 5 du consommateur : empêcher un consommateur de rechercher et de déployer des applications publiques

Vous pouvez empêcher des utilisateurs d'effectuer certaines actions sur les applications.

La stratégie suivante s'applique aux demandes publiques en spécifiant que `serverlessrepo:applicationType` est `public`. Elle empêche les utilisateurs d'effectuer un certain nombre d'actions en spécifiant qu' `Effect` est `Deny`. Pour plus d'informations sur les clés de condition disponibles pour AWS Serverless Application Repository, consultez [Actions, Ressources et Clés de condition pour 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

Cette déclaration de politique peut également être utilisée comme politique de contrôle des services et appliquée à une AWS organisation. Pour plus d'informations sur les politiques de contrôle des services, voir [Politiques de contrôle des services](#) dans le guide de AWS Organizations l'utilisateur.

AWS Serverless Application Repository Exemples de politiques d'application

Les politiques d'autorisation associées aux AWS Serverless Application Repository applications sont appelées politiques d'application. Les politiques d'application déterminent les actions qu'un principal ou PrincipalOrg spécifié peut effectuer sur une AWS Serverless Application Repository application.

Une AWS Serverless Application Repository application est la AWS ressource principale du AWS Serverless Application Repository. AWS Serverless Application Repository les politiques relatives aux applications sont principalement utilisées par les éditeurs pour autoriser les consommateurs à déployer leurs applications, ainsi que pour les opérations connexes, telles que la recherche et l'affichage des détails de ces applications.

Les éditeurs peuvent définir les autorisations d'application selon les trois catégories suivantes :

- Privé : applications qui ont été créées avec le même compte et qui n'ont été partagées avec aucun autre compte. Seuls les utilisateurs qui partagent votre AWS compte sont autorisés à déployer des applications privées.
- Partagées en privé : applications que l'éditeur a explicitement partagées avec un ensemble de AWS comptes spécifique ou avec AWS des comptes d'une AWS organisation. Les utilisateurs sont autorisés à déployer des applications qui ont été partagées avec leur AWS compte ou leur AWS

organisation. Pour plus d'informations sur AWS les organisations, consultez le [Guide de AWS Organizations l'utilisateur](#).

- Partagées publiquement : applications que l'éditeur a partagées avec tout le monde. Tous les consommateurs ont l'autorisation de déployer n'importe quelle application partagée publiquement.

Note

Pour les applications partagées en privé, AWS Serverless Application Repository seuls les AWS comptes principaux sont pris en charge. Les éditeurs peuvent accorder ou refuser l'accès à une AWS Serverless Application Repository application à tous les utilisateurs d'un AWS compte en tant que groupe unique. Les éditeurs ne peuvent pas autoriser ou refuser l'accès à une AWS Serverless Application Repository application à des utilisateurs individuels au sein d'un AWS compte.

Pour de plus amples informations sur la définition des autorisations d'application à l'aide de la AWS Management Console, veuillez consulter [Suppression d'une application](#).

Pour de plus amples informations sur la définition des autorisations d'application à l'aide de AWS CLI et des exemples, veuillez consulter les sections suivantes.

Autorisations des applications (AWS CLI et AWS SDK)

Lorsque vous utilisez le AWS CLI ou les AWS SDK pour définir les autorisations d'une AWS Serverless Application Repository application, vous pouvez spécifier les actions suivantes :

Action	Description
GetApplication	Accorde une autorisation pour afficher des informations sur l'application.
CreateCloudFormationChangeSet	Accorde une autorisation pour déployer l'application. Remarque : cette action n'accorde aucune autre autorisation que celle permettant de déployer.
CreateCloudFormationTemplate	Accorde l'autorisation de créer un modèle AWS CloudFormation pour l'application.

Action	Description
ListApplicationVersions	Accorde l'autorisation de répertorier les versions de l'application.
ListApplicationDependencies	Accorde l'autorisation de répertorier les applications imbriquées dans l'application qu'elles contiennent.
SearchApplications	Accorde une autorisation pour rechercher l'application.
Déploiement	Cette action active toutes les actions répertoriées précédemment dans le tableau. Autrement dit, elle accorde l'autorisation d'afficher l'application, de la déployer, de lister les versions et de la rechercher.

Exemples de stratégie de

Les exemples suivants montrent comment accorder des autorisations à l'aide de la AWS CLI. Pour de plus amples informations sur la façon d'accorder des autorisations à l'aide de l'AWS Management Console, veuillez consulter [Suppression d'une application](#).

Tous les exemples de cette section utilisent les commandes AWS CLI suivantes pour gérer les stratégies d'autorisations associées aux applications AWS Serverless Application Repository :

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

Rubriques

- [Exemple 1 : partager une application avec un autre compte](#)
- [Exemple 2 : partager publiquement une application](#)
- [Exemple 3 : rendre une application privée](#)
- [Exemple 4 : spécification de plusieurs comptes et autorisations](#)
- [Exemple 5 : partager une application avec tous les comptes d'uneAWS organisation](#)
- [Exemple 6 : Partage d'une application avec certains comptes d'uneAWS organisation](#)
- [Exemple 7 : récupérer une stratégie d'application](#)
- [Exemple 8 : autoriser l'imbrication de l'application par certains comptes](#)

Exemple 1 : partager une application avec un autre compte

Pour partager une application avec un autre compte spécifique, mais empêcher qu'elle soit partagée avec d'autres utilisateurs, vous devez spécifier l'identifiant du AWS compte avec lequel vous souhaitez partager en tant que compte principal. Ceci correspond à la définition de l'application sur partagé en privé. Pour ce faire, exécutez la commande AWS CLI suivante.

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

Note

Les applications partagées en privé ne peuvent être utilisées que dans la AWS région où l'application a été créée.

Exemple 2 : partager publiquement une application

Pour rendre une application publique, vous la partagez avec tous en spécifiant « * » comme mandataire, comme illustré dans l'exemple suivant. Les applications qui sont partagées publiquement sont disponibles dans toutes les régions.

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

Note

Pour partager publiquement une application, elle doit avoir à la fois définies les propriétés `LicenseUrl` et `SemanticVersion`.

Exemple 3 : rendre une application privée

Vous pouvez rendre une application privée afin qu'elle ne soit partagée avec personne et qu'elle ne puisse être déployée que par le AWS compte qui en est propriétaire. Pour ce faire, vous supprimez

les principes et les actions de la politique, ce qui supprime également les autorisations accordées aux autres comptes de votre AWS organisation pour déployer votre application.

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

Note

Les applications privées ne peuvent être utilisées que dans la AWS région où l'application a été créée.

Exemple 4 : spécification de plusieurs comptes et autorisations

Vous pouvez accorder plusieurs autorisations, et vous pouvez les accorder à plusieurs AWS comptes à la fois. Pour ce faire, vous spécifiez des listes en tant que mandataire et actions, comme indiqué dans l'exemple suivant.

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

Exemple 5 : partager une application avec tous les comptes d'une AWS organisation

Les autorisations peuvent être accordées à tous les utilisateurs d'une AWS organisation. Pour ce faire, spécifiez l'ID de votre organisation, comme dans l'exemple suivant.

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

Pour plus d'informations sur AWS les organisations, consultez le [Guide de AWS Organizations l'utilisateur](#).

Note

Vous pouvez uniquement spécifier l'AWS organisation dont votre AWS compte est membre. Si vous essayez de spécifier une AWS organisation dont vous n'êtes pas membre, une erreur se produira.

Pour partager votre application avec votre AWS organisation, vous devez inclure une autorisation pour l'UnshareApplication action, au cas où le partage doit être révoqué future.

Exemple 6 : Partage d'une application avec certains comptes d'une AWS organisation

Des autorisations peuvent être accordées à des comptes spécifiques au sein d'une AWS organisation. Pour ce faire, vous devez spécifier une liste de AWS comptes en tant que compte principal et l'identifiant de votre organisation, comme dans l'exemple suivant.

```
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

Vous pouvez uniquement spécifier l'AWS organisation dont votre AWS compte est membre. Si vous essayez de spécifier une AWS organisation dont vous n'êtes pas membre, une erreur se produira.

Pour partager votre application avec votre AWS organisation, vous devez inclure une autorisation pour l'UnshareApplication action, au cas où le partage doit être révoqué future.

Exemple 7 : récupérer une stratégie d'application

Pour afficher la stratégie en cours d'une application, par exemple pour voir si elle est actuellement partagée, vous utilisez la commande `get-application-policy`, tel qu'illustré dans l'exemple suivant.

```
aws serverlessrepo get-application-policy \  

```

```
--region region \  
--application-id application-arn
```

Exemple 8 : autoriser l'imbrication de l'application par certains comptes

Tout le monde peut imbriquer des applications publiques. Pour autoriser l'imbrication de votre application par certains comptes uniquement, vous devez définir les autorisations minimales suivantes, comme illustré à l'exemple suivant.

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

AWS Serverless Application Repository Autorisations d'API : Référence des actions et ressources

Lorsque vous configurez un [contrôle d'accès](#) et écrivez des politiques d'autorisations que vous pouvez attacher à une identité IAM (politiques basées sur une identité), vous pouvez utiliser la table ci-dessous comme référence. Le chaque AWS Serverless Application Repository Opération d'API, les actions correspondantes pour lesquelles vous pouvez accorder des autorisations d'exécution et AWS Ressource pour laquelle vous pouvez accorder les autorisations. Vous spécifiez les actions dans le champ `Action` de la politique ainsi que la valeur des ressources dans le champ `Resource` de la politique.

Pour indiquer une action, utilisez le préfixe `serverlessrepo:` suivi du nom de l'opération d'API (par exemple, `serverlessrepo:ListApplications`).

Opération	URI	Méthode	AWS Ressources (ARN)
fonctionne-ressource : ListApplications	/applications	GET	*
Autorisations nécessaires: dépôt sans serveur :ListAppl ications			

Opération	URI	Méthode	AWSRessources (ARN)
fonctionne-ressource : CreateApplication	/applications	POST	*
Autorisations nécessaires : dépôt sans serveur : CreateAp plication			
fonctionne-ressource : GetApplication	/applications/ <i>id-application</i>	GET	arn:aws:serverless repo: <i>région</i> : <i>id-compte</i> :applicat ions/ <i>nom-application</i>
Autorisations nécessaires : dépôt sans serveur : GetAppli cation			
fonctionne-ressource : DeleteApplication	/applications/ <i>id-application</i>	DELETE	arn:aws:serverless repo: <i>région</i> : <i>id-compte</i> :applicat ions/ <i>nom-application</i>
Autorisations nécessaires : dépôt sans serveur : DeleteAp plication			
fonctionne-ressource : UpdateApplication	/applications/ <i>id-application</i>	CORRECTIF	arn:aws:serverless repo: <i>région</i> : <i>id-compte</i> :applicat ions/ <i>nom-application</i>
Autorisations nécessaires : dépôt sans serveur : UpdateAp plication			

Opération	URI	Méthode	AWSResources (ARN)
fonctionne-ressource : CreateCloudFormationChangeDéfinir	/applications/ <i>id-application</i> / changesets	POST	arn:aws:serverless repo: <i>région</i> : <i>id-compte</i> :applicat ions/ <i>nom-applicat ion</i>
Autorisations nécessaires: dépôt sans serveur :CreateCloudFormationChange Définir			
fonctionne-ressource : GetApplicationStratégie	/applications/ <i>id-application</i> / policy	GET	arn:aws:serverless repo: <i>région</i> : <i>id-compte</i> :applicat ions/ <i>nom-applicat ion</i>
Autorisations nécessaires: dépôt sans serveur :GetApplicationStratégie			
fonctionne-ressource : PutApplicationStratégie	/applications/ <i>id-application</i> / policy	PUT	arn:aws:serverless repo: <i>région</i> : <i>id-compte</i> :applicat ions/ <i>nom-applicat ion</i>
Autorisations nécessaires: dépôt sans serveur :PutApplicationStratégie			

Opération	URI	Méthode	AWSResources (ARN)
fonctionne-ressource : ListApplicationVersions d' Autorisations nécessaires: dépôt sans serveur :ListApplicationVersions d'	/applications/ <i>id-application</i> /versions	GET	arn:aws:serverlessrepo: <i>région</i> : <i>id-compte</i> :applications/ <i>nom-application</i>
fonctionne-ressource : CreateApplicationVersion Autorisations nécessaires: dépôt sans serveur :CreateApplicationVersion	/applications/ <i>id-application</i> /versions/ <i>version-sémantique</i>	PUT	arn:aws:serverlessrepo: <i>région</i> : <i>id-compte</i> :applications/ <i>nom-application</i>
fonctionne-ressource : ListApplicationDependencies Autorisations nécessaires: dépôt sans serveur :ListApplicationDependencies	/applications/ <i>id-application</i> /dependencies	GET	arn:aws:serverlessrepo: <i>région</i> : <i>id-compte</i> :applications/ <i>nom-application</i>

Opération	URI	Méthode	AWSRessources (ARN)
fonctionne-ressource : SearchApplications	N/A	N/A	*
Autorisations nécessaires: dépôt sans serveur :SearchAp plications			

Résolution des problèmes AWS Serverless Application Repository d'identité et d'accès

Utilisez les informations suivantes pour vous aider à diagnostiquer et à résoudre les problèmes courants que vous pouvez rencontrer lors de l'utilisation du AWS Serverless Application Repository et IAM.

Rubriques

- [Je ne suis pas autorisé à effectuer une action dans AWS Serverless Application Repository](#)
- [Je ne suis pas autorisé à exécuter iam : PassRole](#)
- [Je suis administrateur et je souhaite autoriser d'autres personnes à accéder au AWS Serverless Application Repository](#)
- [Je souhaite autoriser des personnes extérieures à mon AWS compte à accéder à mes AWS Serverless Application Repository ressources](#)

Je ne suis pas autorisé à effectuer une action dans AWS Serverless Application Repository

S'il vous AWS Management Console indique que vous n'êtes pas autorisé à effectuer une action, vous devez contacter votre administrateur pour obtenir de l'aide. Votre administrateur est la personne qui vous a fourni votre nom d'utilisateur et votre mot de passe.

L'exemple d'erreur suivant se produit lorsque l'utilisateur IAM `mateojackson` essaie d'utiliser la console pour afficher les détails d'une application mais ne dispose pas des autorisations nécessaires : `serverlessrepo:GetApplication`.

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

Dans ce cas, Mateo demande à son administrateur de mettre à jour ses stratégies pour l'autoriser à accéder à la ressource `my-example-application` en utilisant l'opération `serverlessrepo:GetApplication`.

Je ne suis pas autorisé à exécuter `iam:PassRole`

Si vous recevez une erreur selon laquelle vous n'êtes pas autorisé à exécuter `iam:PassRole` l'action, vos stratégies doivent être mises à jour afin de vous permettre de transmettre un rôle à AWS Serverless Application Repository.

Certains services AWS permettent de transmettre un rôle existant à ce service au lieu de créer un nouveau rôle de service ou un rôle lié à un service. Pour ce faire, un utilisateur doit disposer des autorisations nécessaires pour transmettre le rôle au service.

L'exemple d'erreur suivant se produit lorsqu'un IAM utilisateur nommé `marymajor` essaie d'utiliser la console pour effectuer une action dans AWS Serverless Application Repository. Toutefois, l'action nécessite que le service ait des autorisations accordées par un rôle de service. Mary ne dispose pas des autorisations nécessaires pour transférer le rôle au service.

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

Dans ce cas, les politiques de Mary doivent être mises à jour pour lui permettre d'exécuter l'action `iam:PassRole`.

Si vous avez besoin d'aide, contactez votre AWS administrateur. Votre administrateur vous a fourni vos informations d'identification de connexion.

Je suis administrateur et je souhaite autoriser d'autres personnes à accéder au AWS Serverless Application Repository

Pour autoriser d'autres personnes à accéder à AWS Serverless Application Repository, vous devez accorder l'autorisation aux personnes ou aux applications qui ont besoin d'y accéder. Si vous

utilisez AWS IAM Identity Center pour gérer des personnes et des applications, vous attribuez des ensembles d'autorisations aux utilisateurs ou aux groupes afin de définir leur niveau d'accès. Les ensembles d'autorisations créent et attribuent automatiquement des IAM politiques aux IAM rôles associés à la personne ou à l'application. Pour plus d'informations, consultez la section [Ensembles d'autorisations](#) dans le guide de AWS IAM Identity Center l'utilisateur.

Si vous n'utilisez pas IAM Identity Center, vous devez créer des IAM entités (utilisateurs ou rôles) pour les personnes ou les applications qui ont besoin d'un accès. Vous devez ensuite associer une politique à l'entité qui leur accorde les autorisations appropriées dans AWS Serverless Application Repository. Une fois les autorisations accordées, fournissez les informations d'identification à l'utilisateur ou au développeur de l'application. Ils utiliseront ces informations d'identification pour y accéder AWS. Pour en savoir plus sur la création d'IAMutilisateurs, de groupes, de politiques et d'autorisations, consultez la section [IAMIdentités, politiques et autorisations IAM dans](#) le guide de IAM l'utilisateur.

Je souhaite autoriser des personnes extérieures à mon AWS compte à accéder à mes AWS Serverless Application Repository ressources

Vous pouvez créer un rôle que les utilisateurs provenant d'autres comptes ou les personnes extérieures à votre organisation pourront utiliser pour accéder à vos ressources. Vous pouvez spécifier qui est autorisé à assumer le rôle. Pour les services qui prennent en charge les politiques basées sur les ressources ou les listes de contrôle d'accès (ACLs), vous pouvez utiliser ces politiques pour autoriser les utilisateurs à accéder à vos ressources.

Pour en savoir plus, consultez les éléments suivants :

- Pour savoir si ces fonctionnalités sont prises AWS Serverless Application Repository en charge, consultez [Comment AWS Serverless Application Repository fonctionne-t-il avec IAM](#).
- Pour savoir comment donner accès à vos ressources sur un site Comptes AWS qui vous appartient, consultez la section [Fournir l'accès à un IAM utilisateur dans un autre site Compte AWS que vous possédez](#) dans le Guide de IAM l'utilisateur.
- Pour savoir comment fournir l'accès à vos ressources à des tiers Comptes AWS, consultez la section [Fournir un accès à des ressources Comptes AWS détenues par des tiers](#) dans le Guide de IAM l'utilisateur.
- Pour savoir comment fournir un accès via la fédération d'identité, consultez la section [Fournir un accès aux utilisateurs authentifiés de manière externe \(fédération d'identité\)](#) dans le guide de l'IAMutilisateur.

- Pour connaître la différence entre l'utilisation de rôles et l'utilisation de politiques basées sur les ressources pour l'accès entre comptes, voir Accès aux [ressources entre comptes IAM dans le guide](#) de l'IAMutilisateur.

Journalisation et surveillance dans AWS Serverless Application Repository

La surveillance est essentielle pour assurer la fiabilité, la disponibilité et les performances de vos solutions AWS. Vous devez recueillir les données de surveillance de toutes les parties de votre solution AWS de manière à pouvoir déboguer plus facilement une éventuelle défaillance à plusieurs points. AWS fournit plusieurs outils pour surveiller vos ressources AWS Serverless Application Repository et répondre aux incidents potentiels, comme les suivants :

Journaux AWS CloudTrail

LeAWS Serverless Application Repositoryest intégré àAWS CloudTrail, service qui enregistre les actions réalisées par un utilisateur, un rôle ou unAWSservice dans leAWS Serverless Application Repository. CloudTrailcapture les appels d'API pour leAWS Serverless Application Repositoryen tant qu'événements.

Rubriques

- [Journalisation des AWS Serverless Application Repository appels d'API avec AWS CloudTrail](#)

Journalisation des AWS Serverless Application Repository appels d'API avec AWS CloudTrail

AWS Serverless Application Repositoryest intégré àAWS CloudTrail, qui enregistre les actions effectuées par un utilisateur, un rôle ou unAWSservice dans leAWS Serverless Application Repository. CloudTrailcapture les appels d'API pour leAWS Serverless Application Repositoryen tant qu'événements. Les appels capturés incluent des appels de la console AWS Serverless Application Repository et les appels de code vers les opérations d'API AWS Serverless Application Repository.

Si vous créez un journal de suivi, vous pouvez diffuser en continu lesCloudTrailévénements vers un compartiment Amazon S3, y compris des événements pour leAWS Serverless Application Repository. Si vous ne configurez pas de journal d'activité, vous pouvez toujours afficher les événements les plus récents dans la console CloudTrail dans Historique des événements.

En utilisant les informations collectées par CloudTrail, vous pouvez déterminer quelle demande a été faite à AWS Serverless Application Repository. Vous pouvez aussi déterminer l'adresse IP à partir de laquelle la demande a été faite, qui a effectué la demande, quand elle a eu lieu et autres informations supplémentaires.

Pour en savoir plus sur CloudTrail, veuillez consulter le Guide de l'utilisateur [AWS CloudTrail](#).

Informations AWS Serverless Application Repository dans CloudTrail

CloudTrail est activé sur votre compte AWS lorsque vous créez le compte. Lorsqu'une activité a lieu dans AWS Serverless Application Repository, cette activité est enregistrée dans un événement CloudTrail avec d'autres événements de service AWS dans Historique des événements. Vous pouvez afficher, rechercher et télécharger les événements récents dans votre AWS compte. Pour plus d'informations, consultez [Affichage des événements avec l'historique des événements CloudTrail](#).

Pour un enregistrement continu des événements dans votre compte AWS, y compris les événements pour AWS Serverless Application Repository, créez un journal de suivi. UNsentierpermetCloudTrailpour diffuser des fichiers journaux vers un compartiment Amazon S3. Par défaut, lorsque vous créez un journal d'activité dans la console, il s'applique à toutes les régions AWS. Le journal d'activité consigne les événements de toutes les Régions dans la partition AWSAWS et transfère les fichiers journaux dans le compartiment Amazon S3 de votre choix. En outre, vous pouvez configurer d'autres services AWS pour analyser plus en profondeur les données d'événement collectées dans les journaux CloudTrail et agir sur celles-ci. Pour en savoir plus, consultez les ressources suivantes :

- [Présentation de la création d'un journal d'activité](#)
- [Intégrations et services supportés par CloudTrail](#)
- [Configuration des Notifications de Amazon SNS pour CloudTrail](#)
- [Réception de fichiers journaux CloudTrail de plusieurs régions](#) et [Réception de fichiers journaux CloudTrail de plusieurs comptes](#)

Toutes les actions AWS Serverless Application Repository sont consignées par CloudTrail et documentées sur la page [Ressources AWS Serverless Application Repository](#). À titre d'exemple, les appels vers les opérations `CreateApplication`, `UpdateApplications` et `ListApplications` et génèrent des entrées dans les fichiers journaux CloudTrail.

Chaque événement ou entrée du journal contient des informations sur la personne qui a généré la demande. Les informations relatives à l'identité permettent de déterminer les éléments suivants :

- Si la demande a été effectuée avec les informations d'identification utilisateur racine ou AWS Identity and Access Management (IAM).
- Si la demande a été effectuée avec les informations d'identification de sécurité temporaires d'un rôle ou d'un utilisateur fédéré.
- Si la requête a été effectuée par un autre service AWS.

Pour plus d'informations, consultez la section [Élément `userIdentity` `CloudTrail`](#).

Présentation des entrées des fichiers journaux AWS Serverless Application Repository

Un journal d'activité est une configuration qui permet d'envoyer les événements dans des fichiers journaux à un compartiment Amazon S3 que vous spécifiez. Les fichiers journaux `CloudTrail` contiennent une ou plusieurs entrées de journal. Un événement représente une demande unique provenant de n'importe quelle source et comprend des informations sur l'action demandée, la date et l'heure de l'action, les paramètres de la requête, etc. Les fichiers journaux `CloudTrail` ne constituent pas une série ordonnée retraçant les appels d'API publics. Ils ne suivent aucun ordre précis.

L'exemple suivant montre une entrée de journal `CloudTrail` qui illustre l'action `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"
}
```

Validation de la conformité pour l'AWS Serverless Application Repository

Les auditeurs tiers évaluent la sécurité et la conformité AWS Serverless Application Repository dans le cadre de plusieurs programmes de conformité AWS. Il s'agit notamment des certifications SOC, PCI, FedRAMP, et autres.

Pour obtenir la liste des éléments AWS services qui entrent dans le champ d'application de certains programmes de conformité, consultez [AWS Services concernés par le programme de conformité](#). Pour obtenir des informations générales, consultez [AWS Compliance Programs \(Programmes de conformité\)](#).

Vous pouvez télécharger les rapports d'audit tiers en utilisant AWS Artifact. Pour plus d'informations, consultez [Downloading Reports \(Téléchargement des rapports\) dans AWS Artifact](#).

Votre responsabilité en matière de conformité lorsque vous utilisez AWS Serverless Application Repository est déterminée par la sensibilité de vos données, les objectifs de conformité de votre entreprise, ainsi que la législation et la réglementation en vigueur. AWS fournit les ressources suivantes pour faciliter le respect de la conformité :

- [Guides de démarrage rapide pour la sécurité et la conformité](#)— ces guides de déploiement traitent des considérations architecturales et indiquent les étapes à suivre pour déployer des environnements de référence centrés sur la sécurité et la conformité sur AWS.
- [Ressources de conformité AWS](#) : cet ensemble de manuels et de guides peut s'appliquer à votre secteur et à votre emplacement.

- [AWS Config](#) : ce service AWS permet d'évaluer la conformité des configurations de vos ressources par rapport à des pratiques internes, réglementations et autres directives sectorielles.
- [AWS Security Hub](#) : ce service AWS fournit une vue complète de votre état de sécurité au sein d'AWS qui vous permet de vérifier votre conformité aux normes du secteur et aux bonnes pratiques de sécurité.

Résilience dans la gestion de configuration d'AWS Serverless Application Repository

L'infrastructure mondiale d'AWS repose sur les Régions AWS et les zones de disponibilité AWS. Les Régions fournissent plusieurs zones de disponibilité physiquement séparées et isolées, reliées par un réseau à latence faible, à haut débit et hautement redondant. Avec les zones de disponibilité, vous pouvez concevoir et exploiter des applications et des bases de données qui basculent automatiquement d'une zone de disponibilité à l'autre sans interruption. Les zones de disponibilité sont plus hautement disponibles, tolérantes aux pannes et évolutives que les infrastructures traditionnelles à un ou plusieurs centres de données.

Pour en savoir plus sur les régions AWS et zones de disponibilité , consultez [Infrastructure mondiale AWS](#).

Sécurité de l'infrastructure dans le AWS Serverless Application Repository

En tant que service géré, AWS Serverless Application Repository il est protégé par la sécurité du réseau AWS mondial. Pour plus d'informations sur les services AWS de sécurité et sur la manière dont AWS l'infrastructure est protégée, consultez la section [Sécurité du AWS cloud](#). Pour concevoir votre AWS environnement en utilisant les meilleures pratiques en matière de sécurité de l'infrastructure, consultez la section [Protection de l'infrastructure](#) dans le cadre AWS bien architecturé du pilier de sécurité.

Vous utilisez API les appels AWS publiés pour accéder AWS Serverless Application Repository via le réseau. Les clients doivent prendre en charge les éléments suivants :

- Sécurité de la couche de transport (TLS). Nous avons besoin de la TLS version 1.2 et recommandons la TLS version 1.3.

- Des suites de chiffrement parfaitement confidentielles (PFS) telles que (Ephemeral Diffie-Hellman) ou DHE ECDHE (Elliptic Curve Ephemeral Diffie-Hellman). La plupart des systèmes modernes tels que Java 7 et les versions ultérieures prennent en charge ces modes.

En outre, les demandes doivent être signées à l'aide d'un identifiant de clé d'accès et d'une clé d'accès secrète associés à un IAM principal. Vous pouvez également utiliser [AWS Security Token Service](#) (AWS STS) pour générer des informations d'identification de sécurité temporaires et signer les demandes.

Quotas AWS Serverless Application Repository

Le AWS Serverless Application Repository a un quota pour le nombre d'applications publiques qu'un AWS peut avoir dans chaque compte AWS Région. Ce quota s'applique par région et peut être augmenté. Pour demander une augmentation, utilisez la [console Support Center](#).

Ressource	Quota par défaut
Applications publiques (par AWS Par compte AWS Région)	100

Les quotas suivants s'appliquent au stockage disponible pour les packages de code et les stratégies d'application. Vous ne pouvez pas modifier ces quotas.

Ressource	Quota
Stockage Amazon S3 gratuit pour les packages de code (par AWS Par compte AWS Région)	5 Go
Longueur de la stratégie d'application	6 144 caractères

Dépannage de AWS Serverless Application Repository

Quand vous utilisez AWS Serverless Application Repository, vous pouvez rencontrer des problèmes lorsque vous créez, mettez à jour ou supprimez vos applications. Utilisez cette section pour vous aider à résoudre les problèmes courants que vous pourriez rencontrer. Vous pouvez également rechercher des réponses et publier des questions dans les [forums AWS Serverless Application Repository](#).

Note

Les applications dans AWS Serverless Application Repository sont déployées avec AWS CloudFormation. Pour plus d'informations sur le dépannage des problèmes liés à AWS CloudFormation, consultez le [Guide de dépannage AWS CloudFormation](#).

Rubriques

- [Impossible de rendre une application publique](#)
- [Un quota a été dépassé](#)
- [Un fichier Lisez-moi \(Readme\) mis à jour ne s'affiche pas immédiatement](#)
- [Vous ne pouvez pas déployer une application en raison d'autorisations IAM insuffisantes](#)
- [Vous ne pouvez pas déployer la même application deux fois](#)
- [Pourquoi mon application n'est-elle pas disponible publiquement](#)
- [Contacter Support](#)

Impossible de rendre une application publique

Si vous ne pouvez pas rendre votre application publique, il se peut qu'il manque à votre application un fichier de licence approuvé par l'OSI (Open Source Initiative).

Pour rendre votre application publique, vous avez besoin d'un fichier de licence approuvé par l'OSI, ainsi que d'une version publiée avec succès de l'application avec l'URL du code source de la version. Une fois que l'application a été créée, vous ne pouvez pas mettre à jour la licence d'une application.

Si vous ne pouvez pas rendre votre application publique parce qu'il manque à votre application un fichier de licence, supprimez l'application et créez-en une nouvelle avec le même nom. Assurez-vous

de lui fournir une ou plusieurs licences open source approuvées par l'organisation OSI (Open Source Initiative).

Un quota a été dépassé

Si vous recevez un message d'erreur indiquant qu'un quota a été dépassé, vérifiez si vous avez atteint un quota de ressources. Pour les quotas AWS Serverless Application Repository, veuillez consulter [Quotas AWS Serverless Application Repository](#).

Un fichier Lisez-moi (Readme) mis à jour ne s'affiche pas immédiatement

Quand vous rendez votre application publique, le contenu de votre application peut prendre jusqu'à 24 heures pour être mis à jour. Si vous rencontrez un retard supérieur à 24 heures, contactez AWS Support à l'aide. Pour plus d'informations, consultez les rubriques suivantes.

Vous ne pouvez pas déployer une application en raison d'autorisations IAM insuffisantes

Pour déployer une application AWS Serverless Application Repository, vous avez besoin des autorisations pour les ressources AWS Serverless Application Repository et les piles AWS CloudFormation. Il se peut également que vous ayez besoin d'une autorisation pour utiliser les services sous-jacents décrits dans l'application. Par exemple, si vous créez un compartiment Amazon S3 ou une table Amazon DynamoDB, vous devez être autorisé à accéder à Amazon S3 ou à DynamoDB.

Si vous rencontrez ce type de problème, examinez votre stratégie AWS Identity and Access Management (IAM) et vérifiez que vous disposez des autorisations nécessaires. Pour de plus amples informations, veuillez consulter [Contrôle de l'accès avec AWS Identity and Access Management](#).

Vous ne pouvez pas déployer la même application deux fois

Le nom d'application que vous avez fourni est utilisé comme nom de la pile AWS CloudFormation. Si vous rencontrez des problèmes lors du déploiement d'une application, assurez-vous que vous n'avez pas de pile AWS CloudFormation, existante portant le même nom. Si tel est le cas, fournissez un autre nom d'application ou supprimez la pile existante pour déployer l'application avec le même nom.

Pourquoi mon application n'est-elle pas disponible publiquement

Par défaut, les applications sont privées. Pour rendre votre application publique, suivez la procédure décrite [ici](#).

Contactez Support

Dans certains cas, vous risquez de ne pas trouver de solutions de dépannage dans cette section ou via les [forums AWS Serverless Application Repository](#). Si vous avez souscrité AWS Premium Support, vous pouvez créer une demande d'assistance technique à l'adresse [AWS Support](#).

Avant de contacter AWS Support, veillez à obtenir l'ARN (Amazon Resource Name) de l'application pour laquelle vous avez des questions. L'ARN de l'application est disponible dans la [console AWS Serverless Application Repository](#).

Opérations

L'API REST AWS Serverless Application Repository inclut les opérations suivantes.

- [CreateApplication](#)

Crée une application, en incluant éventuellement unAWSFichier SAM en vue de créer la première version de l'application au cours du même appel.

- [CreateApplicationVersion](#)

Crée une version de l'application.

- [CreateCloudFormationChangeSet](#)

Crée unAWS CloudFormationjeu de modifications pour l'application donnée.

- [CreateCloudFormationTemplate](#)

Crée unAWS CloudFormationmodèle.

- [DeleteApplication](#)

Supprime l'application spécifiée.

- [GetApplication](#)

Obtient l'application spécifiée.

- [GetApplicationPolicy](#)

Récupère la stratégie de l'application.

- [GetCloudFormationTemplate](#)

Obtient le paramètre spécifiéAWS CloudFormationmodèle.

- [ListApplicationDependencies](#)

Récupère la liste des applications imbriquées dans l'application qu'elles contiennent.

- [ListApplications](#)

Répertorie les applications dont le demandeur est propriétaire.

- [ListApplicationVersions](#)

Répertorie les versions de l'application spécifiée.

- [PutApplicationPolicy](#)

Définit la stratégie d'autorisation pour une application. Pour obtenir la liste complète des actions prises en charge pour cette opération, consultez [Autorisations d'application](#).

- [UnshareApplication](#)

Départage une application d'unAWSOrganisation.

Cette opération ne peut être appelée qu'à partir du compte de gestion de l'organisation.

- [UpdateApplication](#)

Met à jour l'application spécifiée.

Ressources

Le AWS Serverless Application Repository RESTAPI inclut les ressources suivantes.

Rubriques

- [Applications](#)
- [Demandes applicationId](#)
- [Ensembles de applicationId modifications pour les applications](#)
- [Applications applicationId Dependencies](#)
- [applicationId Politique relative aux applications](#)
- [Applications applicationId Templates](#)
- [Applications applicationId Templates templateId](#)
- [Applications applicationId Unshare](#)
- [applicationId Versions des applications](#)
- [applicationId Versions des applications semanticVersion](#)

Applications

URI

/applications

HTTP méthodes

GET

ID de l'opération : ListApplications

Répertorie les applications dont le demandeur est propriétaire.

Paramètres Query (Requête)

Nom	Type	Obligatoire	Description
maxItems	Chaîne	False	Le nombre total d'articles à retourner.

Nom	Type	Obligatoire	Description
nextToken	Chaîne	False	Jeton permettant de spécifier où commencer la pagination.

Réponses

Code d'état	Modèle de réponse	Description
200	ApplicationPage	Réussite
400	BadRequestException	L'un des paramètres dans la demande n'est pas valide.
403	ForbiddenException	Le client n'est pas authentifié.
404	NotFoundException	La ressource (par exemple, une déclaration de politique d'accès) spécifiée dans la demande n'existe pas.
500	InternalServerErrorException	Le AWS Serverless Application Repository le service a rencontré une erreur interne.

POST

ID de l'opération : `CreateApplication`

Crée une application, incluant éventuellement un AWS SAM fichier pour créer la première version de l'application dans le même appel.

Réponses

Code d'état	Modèle de réponse	Description
201	Application	Réussite

Code d'état	Modèle de réponse	Description
400	BadRequestException	L'un des paramètres dans la demande n'est pas valide.
403	ForbiddenException	Le client n'est pas authentifié.
409	ConflictException	La ressource existe déjà.
429	TooManyRequestsException	Le client envoie un nombre de demandes supérieur au nombre autorisé par unité de temps.
500	InternalServerErrorException	Le AWS Serverless Application Repository le service a rencontré une erreur interne.

OPTIONS

Réponses

Code d'état	Modèle de réponse	Description
200	Aucun	Réponse 200

Schémas

Corps de la demande

POST schéma

```
{
  "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"
}
```

Corps de la réponse

ApplicationPage schéma

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

Application schéma

```
{
  "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
}
```

```
}
```

BadRequestException schéma

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

ForbiddenException schéma

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

NotFoundException schéma

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

ConflictException schéma

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

TooManyRequestsException schéma

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

InternalServerErrorException schéma

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

Propriétés

Application

Informations sur l'application

applicationId

L'application Amazon Resource Name (ARN).

Type : chaîne

Obligatoire : True

name

Nom de l'application.

Longueur minimale = 1. Longueur maximale = 140

Modèle : [a-zA-Z0-9\-\+]

Type : chaîne

Obligatoire : True

description

Description de l'application.

Longueur minimale = 1. Longueur maximale = 256

Type : chaîne

Obligatoire : True

author

Nom de l'auteur qui publie l'application.

Longueur minimale = 1. Longueur maximale = 127.

Motif « `^[a-z0-9]([a-z0-9]|(? ! -) * [a-z0-9])? $` » ;

Type : chaîne

Obligatoire : True

isVerifiedAuthor

Spécifie si l'auteur de cette application a été vérifié. Cela signifie que AWS a examiné de bonne foi, en tant que fournisseur de services raisonnable et prudent, les informations fournies par le demandeur et a confirmé que l'identité du demandeur est celle revendiquée.

Type : valeur booléenne

Obligatoire : False

verifiedAuthorUrl

Vers URL le profil public d'un auteur vérifié. Ceci URL est soumis par l'auteur.

Type : chaîne

Obligatoire : False

spdxLicenseId

Un identifiant valide provenant de <https://spdx.org/licenses/>.

Type : chaîne

Obligatoire : False

licenseUrl

Lien vers un fichier de licence de l'application qui correspond à la valeur d'`spdxLicenseIdentifiant` de votre application.

Taille maximale 5 Mo

Type : chaîne

Obligatoire : False

readmeUrl

Un lien vers le fichier readme en langage Markdown qui contient une description plus détaillée de l'application et de son fonctionnement.

Taille maximale 5 Mo

Type : chaîne

Obligatoire : False

labels

Étiquettes pour améliorer la découverte des applications dans les résultats de recherche.

Longueur minimale = 1. Longueur maximale = 127. Nombre maximum d'étiquettes : 10

Modèle : `^[a-zA-Z0-9+\\-_:\\V@]+$`

Type : tableau de types string

Obligatoire : False

creationTime

Date et heure de création de cette ressource.

Type : chaîne

Obligatoire : False

homePageUrl

A URL avec plus d'informations sur l'application, par exemple l'emplacement de votre GitHub référentiel pour l'application.

Type : chaîne

Obligatoire : False

version

Informations sur la version de l'application.

Type : [Version](#)

Obligatoire : False

ApplicationPage

Une liste des détails de l'application.

applications

Un ensemble de résumés de candidatures.

Type : tableau de types [ApplicationSummary](#)

Obligatoire : True

nextToken

Le jeton permettant de demander la page de résultats suivante.

Type : chaîne

Obligatoire : False

ApplicationSummary

Résumé des détails de l'application.

applicationId

L'application Amazon Resource Name (ARN).

Type : chaîne

Obligatoire : True

name

Nom de l'application.

Longueur minimale = 1. Longueur maximale = 140

Modèle : [a-zA-Z0-9\-\+]

Type : chaîne

Obligatoire : True

description

Description de l'application.

Longueur minimale = 1. Longueur maximale = 256

Type : chaîne

Obligatoire : True

author

Nom de l'auteur qui publie l'application.

Longueur minimale = 1. Longueur maximale = 127.

Motif « ^ [a-z0-9] (([a-z0-9] | - (? ! -) * [a-z0-9]) ? \$ » ;

Type : chaîne

Obligatoire : True

spdxLicenseId

Un identifiant valide provenant de <https://spdx.org/licenses/>.

Type : chaîne

Obligatoire : False

labels

Étiquettes pour améliorer la découverte des applications dans les résultats de recherche.

Longueur minimale = 1. Longueur maximale = 127. Nombre maximum d'étiquettes : 10

Modèle : ^[a-zA-Z0-9+\-_:\V@]+\\$\

Type : tableau de types string

Obligatoire : False

creationTime

Date et heure de création de cette ressource.

Type : chaîne

Obligatoire : False

homePageUrl

A URL avec plus d'informations sur l'application, par exemple l'emplacement de votre GitHub référentiel pour l'application.

Type : chaîne

Obligatoire : False

BadRequestException

L'un des paramètres dans la demande n'est pas valide.

message

L'un des paramètres dans la demande n'est pas valide.

Type : chaîne

Obligatoire : False

errorCode

400

Type : chaîne

Obligatoire : False

Capability

Valeurs qui doivent être spécifiées pour déployer certaines applications.

CAPABILITY_IAM
CAPABILITY_NAMED_IAM
CAPABILITY_AUTO_EXPAND
CAPABILITY_RESOURCE_POLICY

ConflictException

La ressource existe déjà.

message

La ressource existe déjà.

Type : chaîne

Obligatoire : False

errorCode

409

Type : chaîne

Obligatoire : False

CreateApplicationInput

Créez une demande de candidature.

name

Nom de l'application que vous souhaitez publier.

Longueur minimale = 1. Longueur maximale = 140

Modèle : [a-zA-Z0-9\-\-]+

Type : chaîne

Obligatoire : True

description

Description de l'application.

Longueur minimale = 1. Longueur maximale = 256

Type : chaîne

Obligatoire : True

author

Nom de l'auteur qui publie l'application.

Longueur minimale = 1. Longueur maximale = 127.

Motif « `^[a-z0-9]([a-z0-9]|(? ! -) * [a-z0-9])? $` » ;

Type : chaîne

Obligatoire : True

spdxLicenseId

Un identifiant valide provenant de <https://spdx.org/licenses/>.

Type : chaîne

Obligatoire : False

licenseBody

Un fichier texte local contenant la licence de l'application correspondant à la valeur d'`spdxLicenseId` de votre application. Le fichier est au format `file://<path>/<filename>`.

Taille maximale 5 Mo

Vous ne pouvez spécifier que l'un `licenseBody` des deux `licenseUrl` ; dans le cas contraire, une erreur se produira.

Type : chaîne

Obligatoire : False

licenseUrl

Lien vers l'objet S3 contenant la licence de l'application correspondant à la valeur `spdxLicense ID` de votre application.

Taille maximale 5 Mo

Vous ne pouvez spécifier que l'un `licenseBody` des deux `licenseUrl` ; dans le cas contraire, une erreur se produira.

Type : chaîne

Obligatoire : False

readmeBody

Un fichier texte `readme` local en langage Markdown qui contient une description plus détaillée de l'application et de son fonctionnement. Le fichier est au format `file://<path>/<filename>`.

Taille maximale 5 Mo

Vous ne pouvez spécifier que l'un `readmeBody` des deux `readmeUrl` ; dans le cas contraire, une erreur se produira.

Type : chaîne

Obligatoire : False

readmeUrl

Lien vers l'objet S3 en langage Markdown qui contient une description plus détaillée de l'application et de son fonctionnement.

Taille maximale 5 Mo

Vous ne pouvez spécifier que l'un `readmeBody` des deux `readmeUrl` ; dans le cas contraire, une erreur se produira.

Type : chaîne

Obligatoire : False

labels

Étiquettes pour améliorer la découverte des applications dans les résultats de recherche.

Longueur minimale = 1. Longueur maximale = 127. Nombre maximum d'étiquettes : 10

Modèle : `^[a-zA-Z0-9+\\-_:\\V@]+$`

Type : tableau de types string

Obligatoire : False

homePageUrl

A URL avec plus d'informations sur l'application, par exemple l'emplacement de votre GitHub référentiel pour l'application.

Type : chaîne

Obligatoire : False

semanticVersion

Version sémantique de l'application :

<https://semver.org/>

Type : chaîne

Obligatoire : False

templateBody

Le cru local emballé AWS SAM fichier modèle de votre application. Le fichier est au format `file://<path>/<filename>`.

Vous ne pouvez spécifier que l'un des deux `templateBody` `templateUrl` ; dans le cas contraire, une erreur se produira.

Type : chaîne

Obligatoire : False

templateUrl

Un lien vers l'objet S3 contenant le package AWS SAM modèle de votre candidature.

Vous ne pouvez spécifier que l'un des deux `templateBody` `templateUrl` ; dans le cas contraire, une erreur se produira.

Type : chaîne

Obligatoire : False

sourceCodeUrl

Lien vers un dépôt public pour le code source de votre application, par exemple le code URL d'un GitHub commit spécifique.

Type : chaîne

Obligatoire : False

sourceCodeArchiveUrl

Lien vers l'objet S3 qui contient l'ZIPArchive du code source de cette version de votre application.

Taille maximale 50 Mo

Type : chaîne

Obligatoire : False

ForbiddenException

Le client n'est pas authentifié.

message

Le client n'est pas authentifié.

Type : chaîne

Obligatoire : False

errorCode

403

Type : chaîne

Obligatoire : False

InternalServerErrorException

Le AWS Serverless Application Repository le service a rencontré une erreur interne.

message

Le AWS Serverless Application Repository le service a rencontré une erreur interne.

Type : chaîne

Obligatoire : False

errorCode

500

Type : chaîne

Obligatoire : False

NotFoundException

La ressource (par exemple, une déclaration de politique d'accès) spécifiée dans la demande n'existe pas.

message

La ressource (par exemple, une déclaration de politique d'accès) spécifiée dans la demande n'existe pas.

Type : chaîne

Obligatoire : False

errorCode

404

Type : chaîne

Obligatoire : False

ParameterDefinition

Paramètres pris en charge par l'application.

name

Nom du paramètre.

Type : chaîne

Obligatoire : True

defaultValue

Valeur correspondant au type approprié de modèle à utiliser si aucune valeur n'est spécifiée lors de la création d'une pile. Si vous avez défini des contraintes pour le paramètre, vous devez spécifier une valeur conforme à ces contraintes.

Type : chaîne

Obligatoire : False

description

Chaîne de 4 000 caractères maximum qui décrit le paramètre.

Type : chaîne

Obligatoire : False

type

Type du paramètre.

Valeurs valides : `String` | `Number` | `List<Number>` | `CommaDelimitedList`

String: chaîne littérale.

Par exemple, les utilisateurs peuvent spécifier `"MyUserName"`.

Number: entier ou valeur flottante. AWS CloudFormation valide la valeur du paramètre sous forme de nombre. Toutefois, lorsque vous utilisez le paramètre ailleurs dans votre modèle (par exemple, en utilisant la fonction `Ref` intrinsèque), la valeur du paramètre devient une chaîne.

Par exemple, les utilisateurs peuvent spécifier `"8888"`.

List<Number>: tableau d'entiers ou de nombres flottants séparés par des virgules. AWS CloudFormation valide la valeur du paramètre sous forme de nombres. Toutefois, lorsque vous utilisez le paramètre ailleurs dans votre modèle (par exemple, en utilisant la fonction `Ref` intrinsèque), la valeur du paramètre devient une liste de chaînes.

Par exemple, les utilisateurs peuvent spécifier « 80,20 », puis `Ref` obtenir le résultat. `["80", "20"]`

CommaDelimitedList: tableau de chaînes littérales séparées par des virgules. Le nombre total de chaînes doit correspondre au nombre total de virgules, plus un. De plus, chaque chaîne de membres est découpée en espace.

Par exemple, les utilisateurs peuvent spécifier « test, dev, prod », puis `Ref` obtenir le résultat.

`["test", "dev", "prod"]`

Type : chaîne

Obligatoire : False

noEcho

S'il faut masquer la valeur du paramètre chaque fois que quelqu'un passe un appel décrivant la pile.

Si vous définissez la valeur sur `true`, la valeur du paramètre est masquée par des astérisques (`*****`).

Type : valeur booléenne

Obligatoire : False

allowedPattern

Expression régulière qui représente les modèles que vous voulez autoriser pour les types `String`.

Type : chaîne

Obligatoire : False

constraintDescription

Chaîne qui explique la contrainte lorsque celle-ci n'est pas respectée. Par exemple, sans description de contrainte, un paramètre dont le modèle `[A-Za-z0-9]+` est autorisé affiche le message d'erreur suivant lorsque l'utilisateur spécifie une valeur non valide :

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

En ajoutant une description de contrainte, telle que « ne doit contenir que des lettres majuscules et minuscules et des chiffres », vous pouvez afficher le message d'erreur personnalisé suivant :

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

Type : chaîne

Obligatoire : False

minValue

Valeur numérique qui détermine la plus petite valeur numérique que vous souhaitez autoriser pour `Number` les types.

Type : entier

Obligatoire : False

maxValue

Valeur numérique qui détermine la plus grande valeur numérique que vous souhaitez autoriser pour `Number` les types.

Type : entier

Obligatoire : False

minLength

Valeur entière qui détermine le plus petit nombre de caractères que vous souhaitez autoriser pour les `String` types.

Type : entier

Obligatoire : False

maxLength

Valeur entière qui détermine le plus grand nombre de caractères que vous souhaitez autoriser pour les String types.

Type : entier

Obligatoire : False

allowedValues

Tableau contenant la liste des valeurs autorisées pour le paramètre.

Type : tableau de types string

Obligatoire : False

referencedByResources

Une liste de AWS SAM ressources qui utilisent ce paramètre.

Type : tableau de types string

Obligatoire : True

TooManyRequestsException

Le client envoie un nombre de demandes supérieur au nombre autorisé par unité de temps.

message

Le client envoie un nombre de demandes supérieur au nombre autorisé par unité de temps.

Type : chaîne

Obligatoire : False

errorCode

429

Type : chaîne
Obligatoire : False

Version

Détails de la version de l'application

applicationId

L'application Amazon Resource Name (ARN).

Type : chaîne
Obligatoire : True

semanticVersion

Version sémantique de l'application :

<https://semver.org/>

Type : chaîne
Obligatoire : True

sourceCodeUrl

Lien vers un dépôt public pour le code source de votre application, par exemple le code URL d'un GitHub commit spécifique.

Type : chaîne
Obligatoire : False

sourceCodeArchiveUrl

Lien vers l'objet S3 qui contient l'ZIPArchive du code source de cette version de votre application.

Taille maximale 50 Mo

Type : chaîne
Obligatoire : False

templateUrl

Un lien vers le package AWS SAM modèle de votre candidature.

Type : chaîne

Obligatoire : True

creationTime

Date et heure de création de cette ressource.

Type : chaîne

Obligatoire : True

parameterDefinitions

Tableau de types de paramètres pris en charge par l'application.

Type : tableau de types [ParameterDefinition](#)

Obligatoire : True

requiredCapabilities

Liste de valeurs que vous devez spécifier avant de pouvoir déployer certaines applications. Certaines applications peuvent inclure des ressources susceptibles d'affecter les autorisations de votre AWS compte, par exemple, en créant un nouveau AWS Identity and Access Management (IAM) utilisateurs. Pour ces applications, vous devez explicitement reconnaître leurs capacités en spécifiant ce paramètre.

Les seules valeurs valides sont

CAPABILITY_IAM, CAPABILITY_NAMED_IAM, CAPABILITY_RESOURCE_POLICY, et CAPABILITY_AUTO_EXPAND.

Les ressources suivantes nécessitent que vous spécifiez CAPABILITY_IAM or

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

et [IAM : : AWS IAM: :Role](#). Si l'application contient IAM des ressources, vous pouvez spécifier CAPABILITY_IAM soit CAPABILITY_NAMED_IAM. Si l'application contient des IAM ressources avec des noms personnalisés, vous devez le spécifier CAPABILITY_NAMED_IAM.

Les ressources suivantes nécessitent que vous spécifiez

CAPABILITY_RESOURCE_POLICY : AWS: :Lambda : :Permission, AWS: :PolicyIAM, : : : AWS :, AWS:S3 ApplicationAutoScaling : ScalingPolicy : :, : : : :, et BucketPolicy AWS: : : SQS :. QueuePolicy AWS SNS TopicPolicy

Les applications contenant une ou plusieurs applications imbriquées exigent de spécifier CAPABILITY_AUTO_EXPAND.

Si votre modèle d'application contient l'une des ressources ci-dessus, nous vous recommandons de vérifier toutes les autorisations associées à l'application avant le déploiement. Si vous ne spécifiez pas ce paramètre pour une application nécessitant des fonctionnalités, l'appel échouera.

Type : tableau de types [Capacité](#)

Obligatoire : True

resourcesSupported

Si tous les AWS les ressources contenues dans cette application sont prises en charge dans la région dans laquelle elles sont récupérées.

Type : valeur booléenne

Obligatoire : True

Consultez aussi

Pour plus d'informations sur son utilisation API dans l'une des langues spécifiques AWS SDKset des références, voir ce qui suit :

ListApplications

- [Interface de ligne de commande AWS](#)
- [AWSSDKpour .NET](#)
- [AWSSDKpour C++](#)
- [AWSSDKpour Go v2](#)
- [AWSSDKpour Java V2](#)
- [AWSSDKpour JavaScript V3](#)
- [AWSSDKpour PHP V3](#)

- [AWSSDKpour Python](#)
- [AWSSDKpour Ruby V3](#)

CreateApplication

- [Interface de ligne de commande AWS](#)
- [AWSSDKpour .NET](#)
- [AWSSDKpour C++](#)
- [AWSSDKpour Go v2](#)
- [AWSSDKpour Java V2](#)
- [AWSSDKpour JavaScript V3](#)
- [AWSSDKpour PHP V3](#)
- [AWSSDKpour Python](#)
- [AWSSDKpour Ruby V3](#)

Demandes applicationId

URI

/applications/*applicationId*

HTTPméthodes

GET

ID de l'opération : GetApplication

Obtient l'application spécifiée.

Paramètres de chemin d'accès

Nom	Type	Obligatoire	Description
<i>applicationId</i>	Chaîne	True	Le nom de la ressource Amazon (ARN) de l'application.

Paramètres Query (Requête)

Nom	Type	Obligatoire	Description
<code>semanticVersion</code>	Chaîne	False	Version sémantique de l'application à obtenir.

Réponses

Code d'état	Modèle de réponse	Description
200	Application	Réussite
400	BadRequestException	L'un des paramètres dans la demande n'est pas valide.
403	ForbiddenException	Le client n'est pas authentifié.
404	NotFoundException	La ressource (par exemple, une déclaration de politique d'accès) spécifiée dans la demande n'existe pas.
429	TooManyRequestsException	Le client envoie un nombre de demandes supérieur au nombre autorisé par unité de temps.
500	InternalServerErrorException	Le AWS Serverless Application Repository le service a rencontré une erreur interne.

DELETE

ID de l'opération : `DeleteApplication`

Supprime l'application spécifiée.

Paramètres de chemin d'accès

Nom	Type	Obligatoire	Description
<i>applicationId</i>	Chaîne	True	Le nom de la ressource Amazon (ARN) de l'application.

Réponses

Code d'état	Modèle de réponse	Description
204	Aucun	Réussite
400	BadRequestException	L'un des paramètres dans la demande n'est pas valide.
403	ForbiddenException	Le client n'est pas authentifié.
404	NotFoundException	La ressource (par exemple, une déclaration de politique d'accès) spécifiée dans la demande n'existe pas.
409	ConflictException	La ressource existe déjà.
429	TooManyRequestsException	Le client envoie un nombre de demandes supérieur au nombre autorisé par unité de temps.
500	InternalServerErrorException	Le AWS Serverless Application Repository le service a rencontré une erreur interne.

OPTIONS

Paramètres de chemin d'accès

Nom	Type	Obligatoire	Description
<i>applicationId</i>	Chaîne	True	Le nom de la ressource Amazon (ARN) de l'application.

Réponses

Code d'état	Modèle de réponse	Description
200	Aucun	Réponse 200

PATCH

ID de l'opération : UpdateApplication

Met à jour l'application spécifiée.

Paramètres de chemin d'accès

Nom	Type	Obligatoire	Description
<i>applicationId</i>	Chaîne	True	Le nom de la ressource Amazon (ARN) de l'application.

Réponses

Code d'état	Modèle de réponse	Description
200	Application	Réussite
400	BadRequestException	L'un des paramètres dans la demande n'est pas valide.

Code d'état	Modèle de réponse	Description
403	ForbiddenException	Le client n'est pas authentifié.
404	NotFoundException	La ressource (par exemple, une déclaration de politique d'accès) spécifiée dans la demande n'existe pas.
409	ConflictException	La ressource existe déjà.
429	TooManyRequestsException	Le client envoie un nombre de demandes supérieur au nombre autorisé par unité de temps.
500	InternalServerErrorException	Le AWS Serverless Application Repository le service a rencontré une erreur interne.

Schémas

Corps de la demande

PATCH schéma

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

Corps de la réponse

Application schéma

```
{
  "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
  }
}
```

BadRequestException schéma

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

ForbiddenException schéma

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

NotFoundException schéma

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

ConflictException schéma

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

TooManyRequestsException schéma

```
{
```

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

InternalServerErrorException schéma

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

Propriétés

Application

Informations sur l'application

applicationId

L'application Amazon Resource Name (ARN).

Type : chaîne

Obligatoire : True

name

Nom de l'application.

Longueur minimale = 1. Longueur maximale = 140

Modèle : [a-zA-Z0-9\-\-]+

Type : chaîne

Obligatoire : True

description

Description de l'application.

Longueur minimale = 1. Longueur maximale = 256

Type : chaîne

Obligatoire : True

author

Nom de l'auteur qui publie l'application.

Longueur minimale = 1. Longueur maximale = 127.

Motif « `^[a-z0-9]([a-z0-9]|-(?!-)*[a-z0-9])?$` » ;

Type : chaîne

Obligatoire : True

isVerifiedAuthor

Spécifie si l'auteur de cette application a été vérifié. Cela signifie que AWS a examiné de bonne foi, en tant que fournisseur de services raisonnable et prudent, les informations fournies par le demandeur et a confirmé que l'identité du demandeur est celle revendiquée.

Type : valeur booléenne

Obligatoire : False

verifiedAuthorUrl

Vers URL le profil public d'un auteur vérifié. Ceci URL est soumis par l'auteur.

Type : chaîne

Obligatoire : False

spdxLicenseId

Un identifiant valide provenant de <https://spdx.org/licenses/>.

Type : chaîne

Obligatoire : False

licenseUrl

Lien vers un fichier de licence de l'application qui correspond à la valeur d'`spdxLicenseIdentifiant` de votre application.

Taille maximale 5 Mo

Type : chaîne

Obligatoire : False

readmeUrl

Un lien vers le fichier `readme` en langage Markdown qui contient une description plus détaillée de l'application et de son fonctionnement.

Taille maximale 5 Mo

Type : chaîne

Obligatoire : False

labels

Étiquettes pour améliorer la découverte des applications dans les résultats de recherche.

Longueur minimale = 1. Longueur maximale = 127. Nombre maximum d'étiquettes : 10

Modèle : `^[a-zA-Z0-9+\\-_:\\V@]+$`

Type : tableau de types string

Obligatoire : False

creationTime

Date et heure de création de cette ressource.

Type : chaîne

Obligatoire : False

homePageUrl

A URL avec plus d'informations sur l'application, par exemple l'emplacement de votre GitHub référentiel pour l'application.

Type : chaîne

Obligatoire : False

version

Informations sur la version de l'application.

Type : [Version](#)

Obligatoire : False

BadRequestException

L'un des paramètres dans la demande n'est pas valide.

message

L'un des paramètres dans la demande n'est pas valide.

Type : chaîne

Obligatoire : False

errorCode

400

Type : chaîne

Obligatoire : False

Capability

Valeurs qui doivent être spécifiées pour déployer certaines applications.

CAPABILITY_IAM

CAPABILITY_NAMED_IAM
CAPABILITY_AUTO_EXPAND
CAPABILITY_RESOURCE_POLICY

ConflictException

La ressource existe déjà.

message

La ressource existe déjà.

Type : chaîne
Obligatoire : False

errorCode

409

Type : chaîne
Obligatoire : False

ForbiddenException

Le client n'est pas authentifié.

message

Le client n'est pas authentifié.

Type : chaîne
Obligatoire : False

errorCode

403

Type : chaîne

Obligatoire : False

InternalServerErrorException

Le AWS Serverless Application Repository le service a rencontré une erreur interne.

message

Le AWS Serverless Application Repository le service a rencontré une erreur interne.

Type : chaîne

Obligatoire : False

errorCode

500

Type : chaîne

Obligatoire : False

NotFoundException

La ressource (par exemple, une déclaration de politique d'accès) spécifiée dans la demande n'existe pas.

message

La ressource (par exemple, une déclaration de politique d'accès) spécifiée dans la demande n'existe pas.

Type : chaîne

Obligatoire : False

errorCode

404

Type : chaîne

Obligatoire : False

ParameterDefinition

Paramètres pris en charge par l'application.

name

Nom du paramètre.

Type : chaîne

Obligatoire : True

defaultValue

Valeur correspondant au type approprié de modèle à utiliser si aucune valeur n'est spécifiée lors de la création d'une pile. Si vous avez défini des contraintes pour le paramètre, vous devez spécifier une valeur conforme à ces contraintes.

Type : chaîne

Obligatoire : False

description

Chaîne de 4 000 caractères maximum qui décrit le paramètre.

Type : chaîne

Obligatoire : False

type

Type du paramètre.

Valeurs valides : `String` | `Number` | `List<Number>` | `CommaDelimitedList`

`String`: chaîne littérale.

Par exemple, les utilisateurs peuvent spécifier `"MyUserName"`.

`Number`: entier ou valeur flottante. AWS CloudFormation valide la valeur du paramètre sous forme de nombre. Toutefois, lorsque vous utilisez le paramètre ailleurs dans votre modèle (par exemple, en utilisant la fonction `Ref` intrinsèque), la valeur du paramètre devient une chaîne.

Par exemple, les utilisateurs peuvent spécifier "8888".

`List<Number>`: tableau d'entiers ou de nombres flottants séparés par des virgules. AWS CloudFormation valide la valeur du paramètre sous forme de nombres. Toutefois, lorsque vous utilisez le paramètre ailleurs dans votre modèle (par exemple, en utilisant la fonction `Ref` intrinsèque), la valeur du paramètre devient une liste de chaînes.

Par exemple, les utilisateurs peuvent spécifier « 80,20 », puis `Ref` obtenir le résultat. `["80", "20"]`

`CommaDelimitedList`: tableau de chaînes littérales séparées par des virgules. Le nombre total de chaînes doit correspondre au nombre total de virgules, plus un. De plus, chaque chaîne de membres est découpée en espace.

Par exemple, les utilisateurs peuvent spécifier « test, dev, prod », puis `Ref` obtenir le résultat.

`["test", "dev", "prod"]`

Type : chaîne

Obligatoire : False

`noEcho`

S'il faut masquer la valeur du paramètre chaque fois que quelqu'un passe un appel décrivant la pile. Si vous définissez la valeur sur `true`, la valeur du paramètre est masquée par des astérisques (*****).

Type : valeur booléenne

Obligatoire : False

`allowedPattern`

Expression régulière qui représente les modèles que vous voulez autoriser pour les types `String`.

Type : chaîne

Obligatoire : False

`constraintDescription`

Chaîne qui explique la contrainte lorsque celle-ci n'est pas respectée. Par exemple, sans description de contrainte, un paramètre dont le modèle `[A-Za-z0-9]+` est autorisé affiche le message d'erreur suivant lorsque l'utilisateur spécifie une valeur non valide :

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

En ajoutant une description de contrainte, telle que « ne doit contenir que des lettres majuscules et minuscules et des chiffres », vous pouvez afficher le message d'erreur personnalisé suivant :

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

Type : chaîne

Obligatoire : False

minValue

Valeur numérique qui détermine la plus petite valeur numérique que vous souhaitez autoriser pour `Number` les types.

Type : entier

Obligatoire : False

maxValue

Valeur numérique qui détermine la plus grande valeur numérique que vous souhaitez autoriser pour `Number` les types.

Type : entier

Obligatoire : False

minLength

Valeur entière qui détermine le plus petit nombre de caractères que vous souhaitez autoriser pour les `String` types.

Type : entier

Obligatoire : False

maxLength

Valeur entière qui détermine le plus grand nombre de caractères que vous souhaitez autoriser pour les `String` types.

Type : entier

Obligatoire : False

allowedValues

Tableau contenant la liste des valeurs autorisées pour le paramètre.

Type : tableau de types string

Obligatoire : False

referencedByResources

Une liste de AWS SAM ressources qui utilisent ce paramètre.

Type : tableau de types string

Obligatoire : True

TooManyRequestsException

Le client envoie un nombre de demandes supérieur au nombre autorisé par unité de temps.

message

Le client envoie un nombre de demandes supérieur au nombre autorisé par unité de temps.

Type : chaîne

Obligatoire : False

errorCode

429

Type : chaîne

Obligatoire : False

UpdateApplicationInput

Mettez à jour la demande de candidature.

description

Description de l'application.

Longueur minimale = 1. Longueur maximale = 256

Type : chaîne

Obligatoire : False

author

Nom de l'auteur qui publie l'application.

Longueur minimale = 1. Longueur maximale = 127.

Motif « `^[a-z0-9]([a-z0-9]|(? ! -) * [a-z0-9])? $` » ;

Type : chaîne

Obligatoire : False

readmeBody

Un fichier texte readme en langage Markdown qui contient une description plus détaillée de l'application et de son fonctionnement.

Taille maximale 5 Mo

Type : chaîne

Obligatoire : False

readmeUrl

Un lien vers le fichier readme en langage Markdown qui contient une description plus détaillée de l'application et de son fonctionnement.

Taille maximale 5 Mo

Type : chaîne

Obligatoire : False

labels

Étiquettes pour améliorer la découverte des applications dans les résultats de recherche.

Longueur minimale = 1. Longueur maximale = 127. Nombre maximum d'étiquettes : 10

Modèle : `^[a-zA-Z0-9+\\-_:\\V@]+$`

Type : tableau de types string

Obligatoire : False

homePageUrl

A URL avec plus d'informations sur l'application, par exemple l'emplacement de votre GitHub référentiel pour l'application.

Type : chaîne

Obligatoire : False

Version

Détails de la version de l'application

applicationId

L'application Amazon Resource Name (ARN).

Type : chaîne

Obligatoire : True

semanticVersion

Version sémantique de l'application :

<https://semver.org/>

Type : chaîne

Obligatoire : True

sourceCodeUrl

Lien vers un dépôt public pour le code source de votre application, par exemple le code URL d'un GitHub commit spécifique.

Type : chaîne

Obligatoire : False

sourceCodeArchiveUrl

Lien vers l'objet S3 qui contient l'ZIPArchive du code source de cette version de votre application.

Taille maximale 50 Mo

Type : chaîne

Obligatoire : False

templateUrl

Un lien vers le package AWS SAM modèle de votre candidature.

Type : chaîne

Obligatoire : True

creationTime

Date et heure de création de cette ressource.

Type : chaîne

Obligatoire : True

parameterDefinitions

Tableau de types de paramètres pris en charge par l'application.

Type : tableau de types [ParameterDefinition](#)

Obligatoire : True

requiredCapabilities

Liste de valeurs que vous devez spécifier avant de pouvoir déployer certaines applications. Certaines applications peuvent inclure des ressources susceptibles d'affecter les autorisations de votre AWS compte, par exemple, en créant un nouveau AWS Identity and Access Management (IAM) utilisateurs. Pour ces applications, vous devez explicitement reconnaître leurs capacités en spécifiant ce paramètre.

Les seules valeurs valides sont

CAPABILITY_IAM, CAPABILITY_NAMED_IAM, CAPABILITY_RESOURCE_POLICY, et CAPABILITY_AUTO_EXPAND.

Les ressources suivantes nécessitent que vous spécifiez CAPABILITY_IAM or

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

[et IAM : : AWS IAM: : Role](#). Si l'application contient IAM des ressources, vous pouvez spécifier

CAPABILITY_IAM soit CAPABILITY_NAMED_IAM. Si l'application contient des IAM ressources avec des noms personnalisés, vous devez le spécifier CAPABILITY_NAMED_IAM.

[Les ressources suivantes nécessitent que vous spécifiez](#)

[CAPABILITY_RESOURCE_POLICY : AWS: : Lambda : : Permission, AWS: :: Policy IAM, : : : AWS :](#)

[AWS: S3 Application Auto Scaling : ScalingPolicy : ;, : : : ;, et BucketPolicy AWS: : : SQS :](#)

[QueuePolicy AWS SNS TopicPolicy](#)

Les applications contenant une ou plusieurs applications imbriquées exigent de spécifier CAPABILITY_AUTO_EXPAND.

Si votre modèle d'application contient l'une des ressources ci-dessus, nous vous recommandons de vérifier toutes les autorisations associées à l'application avant le déploiement. Si vous ne spécifiez pas ce paramètre pour une application nécessitant des fonctionnalités, l'appel échouera.

Type : tableau de types [Capacité](#)

Obligatoire : True

resourcesSupported

Si tous les AWS les ressources contenues dans cette application sont prises en charge dans la région dans laquelle elles sont récupérées.

Type : valeur booléenne

Obligatoire : True

Consultez aussi

Pour plus d'informations sur son utilisation API dans l'une des langues spécifiques AWS SDKset des références, voir ce qui suit :

GetApplication

- [Interface de ligne de commande AWS](#)
- [AWSSDKpour. NET](#)
- [AWSSDKpour C++](#)
- [AWSSDKpour Go v2](#)
- [AWSSDKpour Java V2](#)
- [AWSSDKpour JavaScript V3](#)
- [AWSSDKpour PHP V3](#)
- [AWSSDKpour Python](#)
- [AWSSDKpour Ruby V3](#)

DeleteApplication

- [Interface de ligne de commande AWS](#)
- [AWSSDKpour. NET](#)
- [AWSSDKpour C++](#)
- [AWSSDKpour Go v2](#)
- [AWSSDKpour Java V2](#)
- [AWSSDKpour JavaScript V3](#)
- [AWSSDKpour PHP V3](#)
- [AWSSDKpour Python](#)
- [AWSSDKpour Ruby V3](#)

UpdateApplication

- [Interface de ligne de commande AWS](#)
- [AWSSDKpour. NET](#)

- [AWSSDKpour C++](#)
- [AWSSDKpour Go v2](#)
- [AWSSDKpour Java V2](#)
- [AWSSDKpour JavaScript V3](#)
- [AWSSDKpour PHP V3](#)
- [AWSSDKpour Python](#)
- [AWSSDKpour Ruby V3](#)

Ensembles de applicationId modifications pour les applications

URI

/applications/*applicationId*/changesets

HTTPméthodes

POST

ID de l'opération : CreateCloudFormationChangeSet

Crée un AWS CloudFormation ensemble de modifications pour l'application donnée.

Paramètres de chemin d'accès

Nom	Type	Obligatoire	Description
<i>applicationId</i>	Chaîne	True	Le nom de la ressource Amazon (ARN) de l'application.

Réponses

Code d'état	Modèle de réponse	Description
201	ChangeSetDetails	Réussite
400	BadRequestException	L'un des paramètres dans la demande n'est pas valide.

Code d'état	Modèle de réponse	Description
403	ForbiddenException	Le client n'est pas authentifié.
429	TooManyRequestsException	Le client envoie un nombre de demandes supérieur au nombre autorisé par unité de temps.
500	InternalServerErrorException	Le AWS Serverless Application Repository le service a rencontré une erreur interne.

OPTIONS

Paramètres de chemin d'accès

Nom	Type	Obligatoire	Description
<i>applicationId</i>	Chaîne	True	Le nom de la ressource Amazon (ARN) de l'application.

Réponses

Code d'état	Modèle de réponse	Description
200	Aucun	Réponse 200

Schémas

Corps de la demande

POST schéma

```
{
  "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"
  }
]
}
```

Corps de la réponse

ChangeSetDetails schéma

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

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

BadRequestException schéma

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

ForbiddenException schéma

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

TooManyRequestsException schéma

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

InternalServerErrorException schéma

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

Propriétés

BadRequestException

L'un des paramètres dans la demande n'est pas valide.

message

L'un des paramètres dans la demande n'est pas valide.

Type : chaîne

Obligatoire : False

errorCode

400

Type : chaîne

Obligatoire : False

ChangeSetDetails

Détails de l'ensemble de modifications.

applicationId

L'application Amazon Resource Name (ARN).

Type : chaîne

Obligatoire : True

semanticVersion

Version sémantique de l'application :

<https://semver.org/>

Type : chaîne

Obligatoire : True

changeSetId

Le nom de la ressource Amazon (ARN) de l'ensemble de modifications.

Contraintes de longueur : longueur minimale de 1.

Modèle ARN : [-a-Za-Z0-9 :/] *

Type : chaîne
Obligatoire : True

stackId

L'identifiant unique de la pile.

Type : chaîne
Obligatoire : True

CreateCloudFormationChangeSetInput

Créez une demande d'ensemble de modifications d'application.

stackName

Cette propriété correspond au paramètre du même nom pour AWS CloudFormation [CreateChangeSet](#) API.

Type : chaîne
Obligatoire : True

semanticVersion

Version sémantique de l'application :

<https://semver.org/>

Type : chaîne
Obligatoire : False

templateId

Ils UUID sont revenus par CreateCloudFormationTemplate.

Schéma : [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}

Type : chaîne
Obligatoire : False

parameterOverrides

Liste des valeurs des paramètres de l'application.

Type : tableau de types [ParameterValue](#)

Obligatoire : False

capabilities

Liste de valeurs que vous devez spécifier avant de pouvoir déployer certaines applications. Certaines applications peuvent inclure des ressources susceptibles d'affecter les autorisations de votre AWS compte, par exemple, en créant un nouveau AWS Identity and Access Management (IAM) utilisateurs. Pour ces applications, vous devez explicitement reconnaître leurs capacités en spécifiant ce paramètre.

Les seules valeurs valides sont

CAPABILITY_IAM, CAPABILITY_NAMED_IAM, CAPABILITY_RESOURCE_POLICY, et CAPABILITY_AUTO_EXPAND.

Les ressources suivantes nécessitent que vous spécifiez CAPABILITY_IAM or

CAPABILITY_NAMED_IAM : [AWS: IAM : :Group](#), [AWS: : IAM :InstanceProfile](#), [AWS : : :Policy et IAM : : AWS IAM: :Role](#). Si l'application contient IAM des ressources, vous pouvez spécifier CAPABILITY_IAM soit CAPABILITY_NAMED_IAM. Si l'application contient des IAM ressources avec des noms personnalisés, vous devez le spécifier CAPABILITY_NAMED_IAM.

Les ressources suivantes nécessitent que vous spécifiez

[CAPABILITY_RESOURCE_POLICY : AWS: :Lambda : :Permission](#), [AWS: :PolicyIAM, : : : AWS:, AWS :S3 ApplicationAutoScaling : ScalingPolicy : :, : : :, et BucketPolicy AWS: : SQS :. QueuePolicy AWS SNS TopicPolicy](#)

Les applications contenant une ou plusieurs applications imbriquées exigent de spécifier CAPABILITY_AUTO_EXPAND.

Si votre modèle d'application contient l'une des ressources ci-dessus, nous vous recommandons de vérifier toutes les autorisations associées à l'application avant le déploiement. Si vous ne spécifiez pas ce paramètre pour une application nécessitant des fonctionnalités, l'appel échouera.

Type : tableau de types string

Obligatoire : False

changeSetName

Cette propriété correspond au paramètre du même nom pour AWS CloudFormation [CreateChangeSet](#) API.

Type : chaîne
Obligatoire : False

clientToken

Cette propriété correspond au paramètre du même nom pour AWS CloudFormation [CreateChangeSet](#) API.

Type : chaîne
Obligatoire : False

description

Cette propriété correspond au paramètre du même nom pour AWS CloudFormation [CreateChangeSet](#) API.

Type : chaîne
Obligatoire : False

notificationArns

Cette propriété correspond au paramètre du même nom pour AWS CloudFormation [CreateChangeSet](#) API.

Type : tableau de types string
Obligatoire : False

resourceTypes

Cette propriété correspond au paramètre du même nom pour AWS CloudFormation [CreateChangeSet](#) API.

Type : tableau de types string
Obligatoire : False

rollbackConfiguration

Cette propriété correspond au paramètre du même nom pour AWS CloudFormation [CreateChangeSet](#) API.

Type : [RollbackConfiguration](#)

Obligatoire : False

tags

Cette propriété correspond au paramètre du même nom pour AWS CloudFormation [CreateChangeSet](#) API.

Type : tableau de types [Tag](#)

Obligatoire : False

ForbiddenException

Le client n'est pas authentifié.

message

Le client n'est pas authentifié.

Type : chaîne

Obligatoire : False

errorCode

403

Type : chaîne

Obligatoire : False

InternalServerErrorException

Le AWS Serverless Application Repository le service a rencontré une erreur interne.

message

Le AWS Serverless Application Repository le service a rencontré une erreur interne.

Type : chaîne

Obligatoire : False

errorCode

500

Type : chaîne

Obligatoire : False

ParameterValue

Valeur du paramètre de l'application.

name

Clé associée au paramètre. Si vous ne spécifiez pas de clé ni de valeur pour un paramètre en particulier, AWS CloudFormation utilise la valeur par défaut spécifiée dans votre modèle.

Type : chaîne

Obligatoire : True

value

Valeur d'entrée associée au paramètre.

Type : chaîne

Obligatoire : True

RollbackConfiguration

Cette propriété correspond à AWS CloudFormation [RollbackConfiguration](#) Type de données.

rollbackTriggers

Cette propriété correspond au contenu du même nom pour AWS CloudFormation [RollbackConfiguration](#)Type de données.

Type : tableau de types [RollbackTrigger](#)

Obligatoire : False

monitoringTimeInMinutes

Cette propriété correspond au contenu du même nom pour AWS CloudFormation [RollbackConfiguration](#)Type de données.

Type : entier

Obligatoire : False

RollbackTrigger

Cette propriété correspond à AWS CloudFormation [RollbackTrigger](#)Type de données.

arn

Cette propriété correspond au contenu du même nom pour AWS CloudFormation [RollbackTrigger](#)Type de données.

Type : chaîne

Obligatoire : True

type

Cette propriété correspond au contenu du même nom pour AWS CloudFormation [RollbackTrigger](#)Type de données.

Type : chaîne

Obligatoire : True

Tag

Cette propriété correspond à AWS CloudFormation Type de données de [balise](#).

key

Cette propriété correspond au contenu du même nom pour AWS CloudFormation Type de données de [balise](#).

Type : chaîne

Obligatoire : True

value

Cette propriété correspond au contenu du même nom pour AWS CloudFormation Type de données de [balise](#).

Type : chaîne

Obligatoire : True

TooManyRequestsException

Le client envoie un nombre de demandes supérieur au nombre autorisé par unité de temps.

message

Le client envoie un nombre de demandes supérieur au nombre autorisé par unité de temps.

Type : chaîne

Obligatoire : False

errorCode

429

Type : chaîne

Obligatoire : False

Consultez aussi

Pour plus d'informations sur son utilisation API dans l'une des langues spécifiques AWS SDKset des références, voir ce qui suit :

CreateCloudFormationChangeSet

- [Interface de ligne de commande AWS](#)
- [AWSSDKpour .NET](#)
- [AWSSDKpour C++](#)
- [AWSSDKpour Go v2](#)
- [AWSSDKpour Java V2](#)
- [AWSSDKpour JavaScript V3](#)
- [AWSSDKpour PHP V3](#)
- [AWSSDKpour Python](#)
- [AWSSDKpour Ruby V3](#)

Applications applicationId Dependencies

URI

/applications/*applicationId*/dependencies

HTTPméthodes

GET

ID de l'opération : ListApplicationDependencies

Récupère la liste des applications imbriquées dans l'application conteneur.

Paramètres de chemin d'accès

Nom	Type	Obligatoire	Description
<i>applicationId</i>	Chaîne	True	Le nom de la ressource Amazon (ARN) de l'application.

Paramètres Query (Requête)

Nom	Type	Obligatoire	Description
nextToken	Chaîne	False	Jeton permettant de spécifier où commencer la pagination.
maxItems	Chaîne	False	Le nombre total d'articles à retourner.
semanticVersion	Chaîne	False	Version sémantique de l'application à obtenir.

Réponses

Code d'état	Modèle de réponse	Description
200	ApplicationDependencyPage	Réussite
400	BadRequestException	L'un des paramètres dans la demande n'est pas valide.
403	ForbiddenException	Le client n'est pas authentifié.
404	NotFoundException	La ressource (par exemple, une déclaration de politique d'accès) spécifiée dans la demande n'existe pas.
429	TooManyRequestsException	Le client envoie un nombre de demandes supérieur au nombre autorisé par unité de temps.

Code d'état	Modèle de réponse	Description
500	InternalServerException	Le AWS Serverless Application Repository le service a rencontré une erreur interne.

OPTIONS

Paramètres de chemin d'accès

Nom	Type	Obligatoire	Description
<i>applicationId</i>	Chaîne	True	Le nom de la ressource Amazon (ARN) de l'application.

Réponses

Code d'état	Modèle de réponse	Description
200	Aucun	Réponse 200

Schémas

Corps de la réponse

ApplicationDependencyPage schéma

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

BadRequestException schéma

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

ForbiddenException schéma

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

NotFoundException schéma

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

TooManyRequestsException schéma

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

InternalServerErrorException schéma

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

Propriétés

ApplicationDependencyPage

Liste des résumés des applications imbriqués dans l'application.

dependencies

Tableau de résumés d'applications imbriqués dans l'application.

Type : tableau de types [ApplicationDependencySummary](#)

Obligatoire : True

nextToken

Le jeton permettant de demander la page de résultats suivante.

Type : chaîne

Obligatoire : False

ApplicationDependencySummary

Un résumé de l'application imbriqué.

applicationId

Le nom de ressource Amazon (ARN) de l'application imbriquée.

Type : chaîne

Obligatoire : True

semanticVersion

Version sémantique de l'application imbriquée.

Type : chaîne

Obligatoire : True

BadRequestException

L'un des paramètres dans la demande n'est pas valide.

message

L'un des paramètres dans la demande n'est pas valide.

Type : chaîne

Obligatoire : False

errorCode

400

Type : chaîne

Obligatoire : False

ForbiddenException

Le client n'est pas authentifié.

message

Le client n'est pas authentifié.

Type : chaîne

Obligatoire : False

errorCode

403

Type : chaîne

Obligatoire : False

InternalServerErrorException

Le AWS Serverless Application Repository le service a rencontré une erreur interne.

message

Le AWS Serverless Application Repository le service a rencontré une erreur interne.

Type : chaîne

Obligatoire : False

errorCode

500

Type : chaîne

Obligatoire : False

NotFoundException

La ressource (par exemple, une déclaration de politique d'accès) spécifiée dans la demande n'existe pas.

message

La ressource (par exemple, une déclaration de politique d'accès) spécifiée dans la demande n'existe pas.

Type : chaîne

Obligatoire : False

errorCode

404

Type : chaîne

Obligatoire : False

TooManyRequestsException

Le client envoie un nombre de demandes supérieur au nombre autorisé par unité de temps.

message

Le client envoie un nombre de demandes supérieur au nombre autorisé par unité de temps.

Type : chaîne

Obligatoire : False

errorCode

429

Type : chaîne

Obligatoire : False

Consultez aussi

Pour plus d'informations sur son utilisation API dans l'une des langues spécifiques AWS SDKset des références, voir ce qui suit :

ListApplicationDependencies

- [Interface de ligne de commande AWS](#)
- [AWSSDKpour .NET](#)
- [AWSSDKpour C++](#)
- [AWSSDKpour Go v2](#)
- [AWSSDKpour Java V2](#)
- [AWSSDKpour JavaScript V3](#)
- [AWSSDKpour PHP V3](#)
- [AWSSDKpour Python](#)
- [AWSSDKpour Ruby V3](#)

applicationId Politique relative aux applications

URI

/applications/*applicationId*/policy

HTTPméthodes

GET

ID de l'opération : `GetApplicationPolicy`

Récupère la politique de l'application.

Paramètres de chemin d'accès

Nom	Type	Obligatoire	Description
<i>applicationId</i>	Chaîne	True	Le nom de la ressource Amazon (ARN) de l'application.

Réponses

Code d'état	Modèle de réponse	Description
200	ApplicationPolicy	Réussite
400	BadRequestException	L'un des paramètres dans la demande n'est pas valide.
403	ForbiddenException	Le client n'est pas authentifié.
404	NotFoundException	La ressource (par exemple, une déclaration de politique d'accès) spécifiée dans la demande n'existe pas.
429	TooManyRequestsException	Le client envoie un nombre de demandes supérieur au nombre autorisé par unité de temps.
500	InternalServerErrorException	Le AWS Serverless Application Repository le service a rencontré une erreur interne.

PUT

ID de l'opération : PutApplicationPolicy

Définit la politique d'autorisation pour une application. Pour la liste des actions prises en charge pour cette opération, consultez la section [Autorisations des applications](#).

Paramètres de chemin d'accès

Nom	Type	Obligatoire	Description
<i>applicationId</i>	Chaîne	True	Le nom de la ressource Amazon (ARN) de l'application.

Réponses

Code d'état	Modèle de réponse	Description
200	ApplicationPolicy	Réussite
400	BadRequestException	L'un des paramètres dans la demande n'est pas valide.
403	ForbiddenException	Le client n'est pas authentifié.
404	NotFoundException	La ressource (par exemple, une déclaration de politique d'accès) spécifiée dans la demande n'existe pas.
429	TooManyRequestsException	Le client envoie un nombre de demandes supérieur au nombre autorisé par unité de temps.
500	InternalServerErrorException	Le AWS Serverless Application Repository le service a rencontré une erreur interne.

OPTIONS

Paramètres de chemin d'accès

Nom	Type	Obligatoire	Description
<i>applicationId</i>	Chaîne	True	Le nom de la ressource Amazon (ARN) de l'application.

Réponses

Code d'état	Modèle de réponse	Description
200	Aucun	Réponse 200

Schémas

Corps de la demande

PUT schéma

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

Corps de la réponse

ApplicationPolicy schéma

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

BadRequestException schéma

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

ForbiddenException schéma

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

NotFoundException schéma

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

TooManyRequestsException schéma

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

InternalServerErrorException schéma

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

Propriétés

ApplicationPolicy

Déclarations de politique appliquées à l'application.

statements

Un ensemble de déclarations de politique appliquées à l'application.

Type : tableau de types [ApplicationPolicyStatement](#)

Obligatoire : True

ApplicationPolicyStatement

Déclaration de politique appliquée à l'application.

statementId

Un identifiant unique pour le relevé.

Type : chaîne

Obligatoire : False

principals

Une gamme de AWS compte avec lequel IDs partager l'application, ou * pour rendre l'application publique.

Type : tableau de types string

Obligatoire : True

actions

Pour la liste des actions prises en charge pour cette opération, consultez la section [Autorisations des applications](#).

Type : tableau de types string

Obligatoire : True

principalOrgIDs

Le AWS Organizations ID avec lequel partager l'application.

Type : tableau de types string

Obligatoire : False

BadRequestException

L'un des paramètres dans la demande n'est pas valide.

message

L'un des paramètres dans la demande n'est pas valide.

Type : chaîne

Obligatoire : False

errorCode

400

Type : chaîne

Obligatoire : False

ForbiddenException

Le client n'est pas authentifié.

message

Le client n'est pas authentifié.

Type : chaîne

Obligatoire : False

errorCode

403

Type : chaîne

Obligatoire : False

InternalServerErrorException

Le AWS Serverless Application Repository le service a rencontré une erreur interne.

message

Le AWS Serverless Application Repository le service a rencontré une erreur interne.

Type : chaîne

Obligatoire : False

errorCode

500

Type : chaîne

Obligatoire : False

NotFoundException

La ressource (par exemple, une déclaration de politique d'accès) spécifiée dans la demande n'existe pas.

message

La ressource (par exemple, une déclaration de politique d'accès) spécifiée dans la demande n'existe pas.

Type : chaîne

Obligatoire : False

errorCode

404

Type : chaîne

Obligatoire : False

TooManyRequestsException

Le client envoie un nombre de demandes supérieur au nombre autorisé par unité de temps.

message

Le client envoie un nombre de demandes supérieur au nombre autorisé par unité de temps.

Type : chaîne

Obligatoire : False

errorCode

429

Type : chaîne

Obligatoire : False

Consultez aussi

Pour plus d'informations sur son utilisation API dans l'une des langues spécifiques AWS SDKset des références, voir ce qui suit :

GetApplicationPolicy

- [Interface de ligne de commande AWS](#)
- [AWSSDKpour .NET](#)
- [AWSSDKpour C++](#)
- [AWSSDKpour Go v2](#)
- [AWSSDKpour Java V2](#)
- [AWSSDKpour JavaScript V3](#)
- [AWSSDKpour PHP V3](#)
- [AWSSDKpour Python](#)
- [AWSSDKpour Ruby V3](#)

PutApplicationPolicy

- [Interface de ligne de commande AWS](#)
- [AWSSDKpour .NET](#)
- [AWSSDKpour C++](#)
- [AWSSDKpour Go v2](#)
- [AWSSDKpour Java V2](#)
- [AWSSDKpour JavaScript V3](#)
- [AWSSDKpour PHP V3](#)
- [AWSSDKpour Python](#)
- [AWSSDKpour Ruby V3](#)

Applications applicationId Templates

URI

/applications/*applicationId*/templates

HTTPméthodes

POST

ID de l'opération : `CreateCloudFormationTemplate`

Crée un AWS CloudFormation modèle.

Paramètres de chemin d'accès

Nom	Type	Obligatoire	Description
<i>applicationId</i>	Chaîne	True	Le nom de la ressource Amazon (ARN) de l'application.

Réponses

Code d'état	Modèle de réponse	Description
201	TemplateDetails	Réussite
400	BadRequestException	L'un des paramètres dans la demande n'est pas valide.
403	ForbiddenException	Le client n'est pas authentifié.
404	NotFoundException	La ressource (par exemple, une déclaration de politique d'accès) spécifiée dans la demande n'existe pas.
429	TooManyRequestsException	Le client envoie un nombre de demandes supérieur au nombre autorisé par unité de temps.
500	InternalServerErrorException	Le AWS Serverless Application Repository le service a rencontré une erreur interne.

OPTIONS

Paramètres de chemin d'accès

Nom	Type	Obligatoire	Description
<i>applicationId</i>	Chaîne	True	Le nom de la ressource Amazon (ARN) de l'application.

Réponses

Code d'état	Modèle de réponse	Description
200	Aucun	Réponse 200

Schémas

Corps de la demande

POST schéma

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

Corps de la réponse

TemplateDetails schéma

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

```
}
```

BadRequestException schéma

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

ForbiddenException schéma

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

NotFoundException schéma

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

TooManyRequestsException schéma

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

InternalServerErrorException schéma

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

Propriétés

BadRequestException

L'un des paramètres dans la demande n'est pas valide.

message

L'un des paramètres dans la demande n'est pas valide.

Type : chaîne

Obligatoire : False

errorCode

400

Type : chaîne

Obligatoire : False

CreateCloudFormationTemplateInput

Créez un modèle de demande.

semanticVersion

Version sémantique de l'application :

<https://semver.org/>

Type : chaîne

Obligatoire : False

ForbiddenException

Le client n'est pas authentifié.

message

Le client n'est pas authentifié.

Type : chaîne

Obligatoire : False

errorCode

403

Type : chaîne

Obligatoire : False

InternalServerErrorException

Le AWS Serverless Application Repository le service a rencontré une erreur interne.

message

Le AWS Serverless Application Repository le service a rencontré une erreur interne.

Type : chaîne

Obligatoire : False

errorCode

500

Type : chaîne

Obligatoire : False

NotFoundException

La ressource (par exemple, une déclaration de politique d'accès) spécifiée dans la demande n'existe pas.

message

La ressource (par exemple, une déclaration de politique d'accès) spécifiée dans la demande n'existe pas.

Type : chaîne

Obligatoire : False

errorCode

404

Type : chaîne

Obligatoire : False

TemplateDetails

Détails du modèle.

templateId

Ils UUID sont revenus par CreateCloudFormationTemplate.

Schéma : [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}

Type : chaîne

Obligatoire : True

templateUrl

Lien vers le modèle qui peut être utilisé pour déployer l'application à l'aide de AWS CloudFormation.

Type : chaîne

Obligatoire : True

applicationId

L'application Amazon Resource Name (ARN).

Type : chaîne

Obligatoire : True

semanticVersion

Version sémantique de l'application :

<https://semver.org/>

Type : chaîne
Obligatoire : True

status

État du flux de travail de création de modèles.

Valeurs possibles : PREPARING | ACTIVE | EXPIRED

Type : chaîne
Obligatoire : True
Valeurs : PREPARING | ACTIVE | EXPIRED

creationTime

Date et heure de création de cette ressource.

Type : chaîne
Obligatoire : True

expirationTime

Date et heure d'expiration de ce modèle. Les modèles expirent 1 heure après leur création.

Type : chaîne
Obligatoire : True

TooManyRequestsException

Le client envoie un nombre de demandes supérieur au nombre autorisé par unité de temps.

message

Le client envoie un nombre de demandes supérieur au nombre autorisé par unité de temps.

Type : chaîne
Obligatoire : False

errorCode

429

Type : chaîne

Obligatoire : False

Consultez aussi

Pour plus d'informations sur son utilisation API dans l'une des langues spécifiques AWS SDKset des références, voir ce qui suit :

CreateCloudFormationTemplate

- [Interface de ligne de commande AWS](#)
- [AWSSDKpour .NET](#)
- [AWSSDKpour C++](#)
- [AWSSDKpour Go v2](#)
- [AWSSDKpour Java V2](#)
- [AWSSDKpour JavaScript V3](#)
- [AWSSDKpour PHP V3](#)
- [AWSSDKpour Python](#)
- [AWSSDKpour Ruby V3](#)

Applications applicationId Templates templateId

URI

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

HTTPméthodes

GET

ID de l'opération : GetCloudFormationTemplate

Obtient le spécifié AWS CloudFormation modèle.

Paramètres de chemin d'accès

Nom	Type	Obligatoire	Description
<i>applicationId</i>	Chaîne	True	Le nom de la ressource Amazon (ARN) de l'application.
<i>templateId</i>	Chaîne	True	<p>Ils UUID sont revenus par CreateCloudFormationTemplate.</p> <p>Schéma : [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}</p>

Réponses

Code d'état	Modèle de réponse	Description
200	TemplateDetails	Réussite
400	BadRequestException	L'un des paramètres dans la demande n'est pas valide.
403	ForbiddenException	Le client n'est pas authentifié.
404	NotFoundException	La ressource (par exemple, une déclaration de politique d'accès) spécifiée dans la demande n'existe pas.
429	TooManyRequestsException	Le client envoie un nombre de demandes supérieur au nombre autorisé par unité de temps.

Code d'état	Modèle de réponse	Description
500	InternalServerErrorException	Le AWS Serverless Application Repository le service a rencontré une erreur interne.

OPTIONS

Paramètres de chemin d'accès

Nom	Type	Obligatoire	Description
<i>applicationId</i>	Chaîne	True	Le nom de la ressource Amazon (ARN) de l'application.
<i>templateId</i>	Chaîne	True	Ils UUID sont revenus par CreateCloudFormationTemplate. Schéma : [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}

Réponses

Code d'état	Modèle de réponse	Description
200	Aucun	Réponse 200

Schémas

Corps de la réponse

TemplateDetails schéma

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

BadRequestException schéma

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

ForbiddenException schéma

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

NotFoundException schéma

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

TooManyRequestsException schéma

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

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

InternalServerErrorException schéma

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

Propriétés

BadRequestException

L'un des paramètres dans la demande n'est pas valide.

message

L'un des paramètres dans la demande n'est pas valide.

Type : chaîne

Obligatoire : False

errorCode

400

Type : chaîne

Obligatoire : False

ForbiddenException

Le client n'est pas authentifié.

message

Le client n'est pas authentifié.

Type : chaîne

Obligatoire : False

errorCode

403

Type : chaîne

Obligatoire : False

InternalServerErrorException

Le AWS Serverless Application Repository le service a rencontré une erreur interne.

message

Le AWS Serverless Application Repository le service a rencontré une erreur interne.

Type : chaîne

Obligatoire : False

errorCode

500

Type : chaîne

Obligatoire : False

NotFoundException

La ressource (par exemple, une déclaration de politique d'accès) spécifiée dans la demande n'existe pas.

message

La ressource (par exemple, une déclaration de politique d'accès) spécifiée dans la demande n'existe pas.

Type : chaîne

Obligatoire : False

errorCode

404

Type : chaîne

Obligatoire : False

TemplateDetails

Détails du modèle.

templateId

Ils UUID sont revenus par CreateCloudFormationTemplate.

Schéma : [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}

Type : chaîne

Obligatoire : True

templateUrl

Lien vers le modèle qui peut être utilisé pour déployer l'application à l'aide de AWS CloudFormation.

Type : chaîne

Obligatoire : True

applicationId

L'application Amazon Resource Name (ARN).

Type : chaîne

Obligatoire : True

semanticVersion

Version sémantique de l'application :

<https://semver.org/>

Type : chaîne

Obligatoire : True

status

État du flux de travail de création de modèles.

Valeurs possibles : PREPARING | ACTIVE | EXPIRED

Type : chaîne

Obligatoire : True

Valeurs : PREPARING | ACTIVE | EXPIRED

creationTime

Date et heure de création de cette ressource.

Type : chaîne

Obligatoire : True

expirationTime

Date et heure d'expiration de ce modèle. Les modèles expirent 1 heure après leur création.

Type : chaîne

Obligatoire : True

TooManyRequestsException

Le client envoie un nombre de demandes supérieur au nombre autorisé par unité de temps.

message

Le client envoie un nombre de demandes supérieur au nombre autorisé par unité de temps.

Type : chaîne

Obligatoire : False

errorCode

429

Type : chaîne

Obligatoire : False

Consultez aussi

Pour plus d'informations sur son utilisation API dans l'une des langues spécifiques AWS SDKset des références, voir ce qui suit :

GetCloudFormationTemplate

- [Interface de ligne de commande AWS](#)
- [AWSSDKpour .NET](#)
- [AWSSDKpour C++](#)
- [AWSSDKpour Go v2](#)
- [AWSSDKpour Java V2](#)
- [AWSSDKpour JavaScript V3](#)
- [AWSSDKpour PHP V3](#)
- [AWSSDKpour Python](#)
- [AWSSDKpour Ruby V3](#)

Applications applicationId Unshare

URI

/applications/*applicationId*/unshare

HTTPméthodes

POST

ID de l'opération : UnshareApplication

Annule le partage d'une application d'un AWS Organisation.

Cette opération ne peut être appelée qu'à partir du compte de gestion de l'organisation.

Paramètres de chemin d'accès

Nom	Type	Obligatoire	Description
<i>applicationId</i>	Chaîne	True	Le nom de la ressource Amazon (ARN) de l'application.

Réponses

Code d'état	Modèle de réponse	Description
204	Aucun	Réussite
400	BadRequestException	L'un des paramètres dans la demande n'est pas valide.
403	ForbiddenException	Le client n'est pas authentifié.
404	NotFoundException	La ressource (par exemple, une déclaration de politique d'accès) spécifiée dans la demande n'existe pas.
429	TooManyRequestsException	Le client envoie un nombre de demandes supérieur au nombre autorisé par unité de temps.
500	InternalServerErrorException	Le AWS Serverless Application Repository le service a rencontré une erreur interne.

OPTIONS

Paramètres de chemin d'accès

Nom	Type	Obligatoire	Description
<i>applicationId</i>	Chaîne	True	Le nom de la ressource Amazon (ARN) de l'application.

Réponses

Code d'état	Modèle de réponse	Description
200	Aucun	Réponse 200

Schémas

Corps de la demande

POST schéma

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

Corps de la réponse

BadRequestException schéma

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

ForbiddenException schéma

```
{
```

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

NotFoundException schéma

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

TooManyRequestsException schéma

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

InternalServerErrorException schéma

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

Propriétés

BadRequestException

L'un des paramètres dans la demande n'est pas valide.

message

L'un des paramètres dans la demande n'est pas valide.

Type : chaîne

Obligatoire : False

errorCode

400

Type : chaîne

Obligatoire : False

ForbiddenException

Le client n'est pas authentifié.

message

Le client n'est pas authentifié.

Type : chaîne

Obligatoire : False

errorCode

403

Type : chaîne

Obligatoire : False

InternalServerErrorException

Le AWS Serverless Application Repository le service a rencontré une erreur interne.

message

Le AWS Serverless Application Repository le service a rencontré une erreur interne.

Type : chaîne

Obligatoire : False

errorCode

500

Type : chaîne
Obligatoire : False

NotFoundException

La ressource (par exemple, une déclaration de politique d'accès) spécifiée dans la demande n'existe pas.

message

La ressource (par exemple, une déclaration de politique d'accès) spécifiée dans la demande n'existe pas.

Type : chaîne
Obligatoire : False

errorCode

404

Type : chaîne
Obligatoire : False

TooManyRequestsException

Le client envoie un nombre de demandes supérieur au nombre autorisé par unité de temps.

message

Le client envoie un nombre de demandes supérieur au nombre autorisé par unité de temps.

Type : chaîne
Obligatoire : False

errorCode

429

Type : chaîne

Obligatoire : False

UnshareApplicationInput

Annuler le partage de la demande d'application.

organizationId

Le AWS Organizations ID à partir duquel annuler le partage de l'application.

Type : chaîne

Obligatoire : True

Consultez aussi

Pour plus d'informations sur son utilisation API dans l'une des langues spécifiques AWS SDKset des références, voir ce qui suit :

UnshareApplication

- [Interface de ligne de commande AWS](#)
- [AWSSDKpour .NET](#)
- [AWSSDKpour C++](#)
- [AWSSDKpour Go v2](#)
- [AWSSDKpour Java V2](#)
- [AWSSDKpour JavaScript V3](#)
- [AWSSDKpour PHP V3](#)
- [AWSSDKpour Python](#)
- [AWSSDKpour Ruby V3](#)

applicationId Versions des applications

URI

/applications/*applicationId*/versions

HTTPméthodes

GET

ID de l'opération : `ListApplicationVersions`

Répertorie les versions de l'application spécifiée.

Paramètres de chemin d'accès

Nom	Type	Obligatoire	Description
<i><code>applicationId</code></i>	Chaîne	True	Le nom de la ressource Amazon (ARN) de l'application.

Paramètres Query (Requête)

Nom	Type	Obligatoire	Description
<code>maxItems</code>	Chaîne	False	Le nombre total d'articles à retourner.
<code>nextToken</code>	Chaîne	False	Jeton permettant de spécifier où commencer la pagination.

Réponses

Code d'état	Modèle de réponse	Description
200	ApplicationVersionPage	Réussite
400	BadRequestException	L'un des paramètres dans la demande n'est pas valide.
403	ForbiddenException	Le client n'est pas authentifié.

Code d'état	Modèle de réponse	Description
404	NotFoundException	La ressource (par exemple, une déclaration de politique d'accès) spécifiée dans la demande n'existe pas.
429	TooManyRequestsException	Le client envoie un nombre de demandes supérieur au nombre autorisé par unité de temps.
500	InternalServerErrorException	Le AWS Serverless Application Repository le service a rencontré une erreur interne.

OPTIONS

Paramètres de chemin d'accès

Nom	Type	Obligatoire	Description
<i>applicationId</i>	Chaîne	True	Le nom de la ressource Amazon (ARN) de l'application.

Réponses

Code d'état	Modèle de réponse	Description
200	Aucun	Réponse 200

Schémas

Corps de la réponse

ApplicationVersionPage schéma

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

BadRequestException schéma

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

ForbiddenException schéma

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

NotFoundException schéma

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

TooManyRequestsException schéma

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

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

InternalServerErrorException schéma

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

Propriétés

ApplicationVersionPage

Liste des résumés des versions de l'application.

versions

Tableau de résumés des versions de l'application.

Type : tableau de types [VersionSummary](#)

Obligatoire : True

nextToken

Le jeton permettant de demander la page de résultats suivante.

Type : chaîne

Obligatoire : False

BadRequestException

L'un des paramètres dans la demande n'est pas valide.

message

L'un des paramètres dans la demande n'est pas valide.

Type : chaîne

Obligatoire : False

errorCode

400

Type : chaîne

Obligatoire : False

ForbiddenException

Le client n'est pas authentifié.

message

Le client n'est pas authentifié.

Type : chaîne

Obligatoire : False

errorCode

403

Type : chaîne

Obligatoire : False

InternalServerErrorException

Le AWS Serverless Application Repository le service a rencontré une erreur interne.

message

Le AWS Serverless Application Repository le service a rencontré une erreur interne.

Type : chaîne

Obligatoire : False

errorCode

500

Type : chaîne
Obligatoire : False

NotFoundException

La ressource (par exemple, une déclaration de politique d'accès) spécifiée dans la demande n'existe pas.

message

La ressource (par exemple, une déclaration de politique d'accès) spécifiée dans la demande n'existe pas.

Type : chaîne
Obligatoire : False

errorCode

404

Type : chaîne
Obligatoire : False

TooManyRequestsException

Le client envoie un nombre de demandes supérieur au nombre autorisé par unité de temps.

message

Le client envoie un nombre de demandes supérieur au nombre autorisé par unité de temps.

Type : chaîne
Obligatoire : False

errorCode

429

Type : chaîne
Obligatoire : False

VersionSummary

Un résumé de la version de l'application.

applicationId

L'application Amazon Resource Name (ARN).

Type : chaîne

Obligatoire : True

semanticVersion

Version sémantique de l'application :

<https://semver.org/>

Type : chaîne

Obligatoire : True

sourceCodeUrl

Lien vers un dépôt public pour le code source de votre application, par exemple le code URL d'un GitHub commit spécifique.

Type : chaîne

Obligatoire : False

creationTime

Date et heure de création de cette ressource.

Type : chaîne

Obligatoire : True

Consultez aussi

Pour plus d'informations sur son utilisation API dans l'une des langues spécifiques AWS SDKset des références, voir ce qui suit :

ListApplicationVersions

- [Interface de ligne de commande AWS](#)
- [AWSSDKpour .NET](#)
- [AWSSDKpour C++](#)
- [AWSSDKpour Go v2](#)
- [AWSSDKpour Java V2](#)
- [AWSSDKpour JavaScript V3](#)
- [AWSSDKpour PHP V3](#)
- [AWSSDKpour Python](#)
- [AWSSDKpour Ruby V3](#)

applicationId Versions des applications semanticVersion

URI

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

HTTPméthodes

PUT

ID de l'opération : CreateApplicationVersion

Crée une version de l'application.

Paramètres de chemin d'accès

Nom	Type	Obligatoire	Description
<i>applicationId</i>	Chaîne	True	Le nom de la ressource Amazon (ARN) de l'application.
<i>semanticVersion</i>	Chaîne	True	La version sémantique de la nouvelle version.

Réponses

Code d'état	Modèle de réponse	Description
201	Version	Réussite
400	BadRequestException	L'un des paramètres dans la demande n'est pas valide.
403	ForbiddenException	Le client n'est pas authentifié.
409	ConflictException	La ressource existe déjà.
429	TooManyRequestsException	Le client envoie un nombre de demandes supérieur au nombre autorisé par unité de temps.
500	InternalServerErrorException	Le AWS Serverless Application Repository le service a rencontré une erreur interne.

OPTIONS

Paramètres de chemin d'accès

Nom	Type	Obligatoire	Description
<i>applicationId</i>	Chaîne	True	Le nom de la ressource Amazon (ARN) de l'application.
<i>semanticVersion</i>	Chaîne	True	La version sémantique de la nouvelle version.

Réponses

Code d'état	Modèle de réponse	Description
200	Aucun	Réponse 200

Schémas

Corps de la demande

PUT schéma

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

Corps de la réponse

Version schéma

```
{
  "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
}
```

BadRequestException schéma

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

ForbiddenException schéma

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

ConflictException schéma

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

TooManyRequestsException schéma

```
{
```

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

InternalServerErrorException schéma

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

Propriétés

BadRequestException

L'un des paramètres dans la demande n'est pas valide.

message

L'un des paramètres dans la demande n'est pas valide.

Type : chaîne

Obligatoire : False

errorCode

400

Type : chaîne

Obligatoire : False

Capability

Valeurs qui doivent être spécifiées pour déployer certaines applications.

CAPABILITY_IAM

CAPABILITY_NAMED_IAM

CAPABILITY_AUTO_EXPAND

CAPABILITY_RESOURCE_POLICY

ConflictException

La ressource existe déjà.

message

La ressource existe déjà.

Type : chaîne

Obligatoire : False

errorCode

409

Type : chaîne

Obligatoire : False

CreateApplicationVersionInput

Créez une demande de version.

templateBody

Le brut emballé AWS SAM modèle de votre candidature.

Type : chaîne

Obligatoire : False

templateUrl

Un lien vers le package AWS SAM modèle de votre candidature.

Type : chaîne

Obligatoire : False

sourceCodeUrl

Lien vers un dépôt public pour le code source de votre application, par exemple le code URL d'un GitHub commit spécifique.

Type : chaîne

Obligatoire : False

sourceCodeArchiveUrl

Lien vers l'objet S3 qui contient l'ZIParchive du code source de cette version de votre application.

Taille maximale 50 Mo

Type : chaîne

Obligatoire : False

ForbiddenException

Le client n'est pas authentifié.

message

Le client n'est pas authentifié.

Type : chaîne

Obligatoire : False

errorCode

403

Type : chaîne

Obligatoire : False

InternalServerErrorException

Le AWS Serverless Application Repository le service a rencontré une erreur interne.

message

Le AWS Serverless Application Repository le service a rencontré une erreur interne.

Type : chaîne

Obligatoire : False

errorCode

500

Type : chaîne

Obligatoire : False

ParameterDefinition

Paramètres pris en charge par l'application.

name

Nom du paramètre.

Type : chaîne

Obligatoire : True

defaultValue

Valeur correspondant au type approprié de modèle à utiliser si aucune valeur n'est spécifiée lors de la création d'une pile. Si vous avez défini des contraintes pour le paramètre, vous devez spécifier une valeur conforme à ces contraintes.

Type : chaîne

Obligatoire : False

description

Chaîne de 4 000 caractères maximum qui décrit le paramètre.

Type : chaîne

Obligatoire : False

type

Type du paramètre.

Valeurs valides : `String` | `Number` | `List<Number>` | `CommaDelimitedList`

String: chaîne littérale.

Par exemple, les utilisateurs peuvent spécifier `"MyUserName"`.

Number: entier ou valeur flottante. AWS CloudFormation valide la valeur du paramètre sous forme de nombre. Toutefois, lorsque vous utilisez le paramètre ailleurs dans votre modèle (par exemple, en utilisant la fonction `Ref` intrinsèque), la valeur du paramètre devient une chaîne.

Par exemple, les utilisateurs peuvent spécifier `"8888"`.

List<Number>: tableau d'entiers ou de nombres flottants séparés par des virgules. AWS CloudFormation valide la valeur du paramètre sous forme de nombres. Toutefois, lorsque vous utilisez le paramètre ailleurs dans votre modèle (par exemple, en utilisant la fonction `Ref` intrinsèque), la valeur du paramètre devient une liste de chaînes.

Par exemple, les utilisateurs peuvent spécifier « 80,20 », puis `Ref` obtenir le résultat. `["80", "20"]`

CommaDelimitedList: tableau de chaînes littérales séparées par des virgules. Le nombre total de chaînes doit correspondre au nombre total de virgules, plus un. De plus, chaque chaîne de membres est découpée en espace.

Par exemple, les utilisateurs peuvent spécifier « test, dev, prod », puis `Ref` obtenir le résultat. `["test", "dev", "prod"]`

Type : chaîne
Obligatoire : False

noEcho

S'il faut masquer la valeur du paramètre chaque fois que quelqu'un passe un appel décrivant la pile. Si vous définissez la valeur sur `true`, la valeur du paramètre est masquée par des astérisques (`*****`).

Type : valeur booléenne
Obligatoire : False

allowedPattern

Expression régulière qui représente les modèles que vous voulez autoriser pour les types `String`.

Type : chaîne
Obligatoire : False

constraintDescription

Chaîne qui explique la contrainte lorsque celle-ci n'est pas respectée. Par exemple, sans description de contrainte, un paramètre dont le modèle `[A-Za-z0-9]+` est autorisé affiche le message d'erreur suivant lorsque l'utilisateur spécifie une valeur non valide :

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

En ajoutant une description de contrainte, telle que « ne doit contenir que des lettres majuscules et minuscules et des chiffres », vous pouvez afficher le message d'erreur personnalisé suivant :

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

Type : chaîne

Obligatoire : False

minValue

Valeur numérique qui détermine la plus petite valeur numérique que vous souhaitez autoriser pour `Number` les types.

Type : entier

Obligatoire : False

maxValue

Valeur numérique qui détermine la plus grande valeur numérique que vous souhaitez autoriser pour `Number` les types.

Type : entier

Obligatoire : False

minLength

Valeur entière qui détermine le plus petit nombre de caractères que vous souhaitez autoriser pour les `String` types.

Type : entier

Obligatoire : False

maxLength

Valeur entière qui détermine le plus grand nombre de caractères que vous souhaitez autoriser pour les `String` types.

Type : entier

Obligatoire : False

allowedValues

Tableau contenant la liste des valeurs autorisées pour le paramètre.

Type : tableau de types string

Obligatoire : False

referencedByResources

Une liste de AWS SAM ressources qui utilisent ce paramètre.

Type : tableau de types string

Obligatoire : True

TooManyRequestsException

Le client envoie un nombre de demandes supérieur au nombre autorisé par unité de temps.

message

Le client envoie un nombre de demandes supérieur au nombre autorisé par unité de temps.

Type : chaîne

Obligatoire : False

errorCode

429

Type : chaîne

Obligatoire : False

Version

Détails de la version de l'application

applicationId

L'application Amazon Resource Name (ARN).

Type : chaîne

Obligatoire : True

semanticVersion

Version sémantique de l'application :

<https://semver.org/>

Type : chaîne

Obligatoire : True

sourceCodeUrl

Lien vers un dépôt public pour le code source de votre application, par exemple le code URL d'un GitHub commit spécifique.

Type : chaîne

Obligatoire : False

sourceCodeArchiveUrl

Lien vers l'objet S3 qui contient l'ZIPArchive du code source de cette version de votre application.

Taille maximale 50 Mo

Type : chaîne

Obligatoire : False

templateUrl

Un lien vers le package AWS SAM modèle de votre candidature.

Les applications contenant une ou plusieurs applications imbriquées exigent de spécifier `CAPABILITY_AUTO_EXPAND`.

Si votre modèle d'application contient l'une des ressources ci-dessus, nous vous recommandons de vérifier toutes les autorisations associées à l'application avant le déploiement. Si vous ne spécifiez pas ce paramètre pour une application nécessitant des fonctionnalités, l'appel échouera.

Type : tableau de types [Capacité](#)

Obligatoire : True

resourcesSupported

Si tous les AWS les ressources contenues dans cette application sont prises en charge dans la région dans laquelle elles sont récupérées.

Type : valeur booléenne

Obligatoire : True

Consultez aussi

Pour plus d'informations sur son utilisation API dans l'une des langues spécifiques AWS SDKset des références, voir ce qui suit :

CreateApplicationVersion

- [Interface de ligne de commande AWS](#)
- [AWSSDKpour .NET](#)
- [AWSSDKpour C++](#)
- [AWSSDKpour Go v2](#)
- [AWSSDKpour Java V2](#)
- [AWSSDKpour JavaScript V3](#)
- [AWSSDKpour PHP V3](#)
- [AWSSDKpour Python](#)
- [AWSSDKpour Ruby V3](#)

Historique du document

- Version de l'API : dernière en date
- Dernière mise à jour de la documentation : 10 mars 2020

Le tableau suivant décrit les modifications importantes dans chaque édition du Guide du développeur AWS Serverless Application Repository. Pour recevoir les notifications des mises à jour de cette documentation, abonnez-vous à un flux RSS.

Modification	Description	Date
Mises à jour concernant le partage et la restriction de l'accès aux applications	Ajout de la prise en charge du partage d'applications sur les comptes d'une AWS organisation et de la restriction de l'accès aux applications publiques pour les AWS comptes et AWS Organisations. Pour plus d'exemples de partage d'applications avec les utilisateurs d'une organisation, consultez la section Exemples de politiques d'AWS Serverless Application Repository yapplication . Pour obtenir des exemples de restriction de l'accès aux applications publiques, consultez Exemples de stratégies AWS Serverless Application Repository basées sur l'identité .	10 mars 2020
Nouvelles ressources prises en charge	Ajout de la prise en charge d'un certain nombre de ressources supplémentaires. Pour obtenir la liste complète	17 janvier 2020

des ressources, veuillez consulter [la liste AWS des ressources](#)

[???](#)

AWS Serverless Application Repository est désormais disponible dans les régions Chine et de Ningxia en Chine. Pour de plus amples informations sur les régions et points de terminaison AWS Serverless Application Repository, veuillez consulter [Régions et points de terminaison](#) dans le Références générales AWS.

15 janvier 2020

[Section de sécurité mise à jour pour des raisons de cohérence avec les autres AWS services.](#)

Pour de plus amples informations, veuillez consulter [Sécurité](#).

2 janvier 2020

[Processus simplifié de publication des applications](#)

La nouvelle commande `publish` de l'interface de la CLI AWS SAM simplifie le processus de publication des applications sans serveur dans le AWS Serverless Application Repository. Pour consulter un end-to-end didacticiel sur le téléchargement et la publication d'un exemple d'application, voir [Démarrage rapide : publication d'applications](#). Pour obtenir des instructions sur la publication d'une application que vous avez déjà développée et testée dans le AWS cloud, consultez [Publier une application via la AWS SAM CLI](#).

21 décembre 2018

[Prise en charge des applications et des couches imbriquées](#)

Ajout de la prise en charge des applications imbriquées et des couches. Cela inclut les mises à jour des [AWS ressources prises en charge](#) et la [reconnaissance des capacités des applications](#).

29 novembre 2018

Publication d'applications avec des rôles IAM et des politiques de ressources personnalisés	Ajout de la prise en charge de la publication d'applications avec des rôles IAM et des stratégies de ressources personnalisés. Cela inclut des mises à jour des flux de travail des applications de consommation et des applications de publication , ainsi que des mises à jour AWSdes ressources prises en charge et des références d'API dans le guide duAWS Serverless Application Repository développeur.	16 novembre 2018
mises à jour du modèle de politique	Mises à jour des modèles de politique pris en charge dans le guide duAWS Serverless Application Repository développeur.	26 septembre 2018
mises à jour de documentation	Ajout d'une rubrique sur l'authentification et le contrôle d'accès au guide duAWS Serverless Application Repository développeur.	2 juillet 2018

Publication publique

Version publique duAWS Serverless Application Repository, qui est désormais disponible dans 14AWS régions. Pour plus d'informations sur lesAWS régions dans lesquelles leAWS Serverless Application Repository est disponible et lesAWS Serverless Application Repository points de terminaison, voir [Régions et points de terminaison](#) dans le Référence s générales AWS.

20 février 2018

Nouveau guide

Il s'agit de la première version préliminaire du Guide duAWS Serverless Application Repository développeur.

30 novembre 2017

Glossaire AWS

Pour connaître la terminologie la plus récente d'AWS, consultez le [Glossaire AWS](#) dans la Référence Glossaire AWS.

Les traductions sont fournies par des outils de traduction automatique. En cas de conflit entre le contenu d'une traduction et celui de la version originale en anglais, la version anglaise prévaudra.