



Entwicklerhandbuch

AWS Serverless Application Repository



AWS Serverless Application Repository: Entwicklerhandbuch

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

Die Handelsmarken und Handelsaufmachung von Amazon dürfen nicht in einer Weise in Verbindung mit nicht von Amazon stammenden Produkten oder Services verwendet werden, durch die Kunden irregeführt werden könnten oder Amazon in schlechtem Licht dargestellt oder diskreditiert werden könnte. Alle anderen Marken, die nicht im Besitz von Amazon sind, gehören den jeweiligen Besitzern, die möglicherweise mit Amazon verbunden sind oder von Amazon gesponsert werden.

Table of Contents

Was ist das AWS Serverless Application Repository?	1
Nächste Schritte	1
Quick Start: Veröffentlichen von Anwendungen	2
Übersicht	2
Hello World-Anwendung	2
Bevor Sie beginnen	3
Schritt 1: Initialisieren der Anwendung	3
Schritt 2: Lokales Testen der Anwendung	4
Schritt 3: Verpacken der Anwendung	5
Schritt 4: Veröffentlichen der Anwendung	7
Nächste Schritte	7
Weitere Informationen	8
Veröffentlichen von Anwendungen	9
Verwenden von AWS SAM mit der AWS Serverless Application Repository	10
UnterstützteAWSRessourcen imAWS Serverless Application Repository	10
Richtlinienvorlagen	11
Liste der in unterstütztenAWSRessourcen	11
So veröffentlichen Sie Anwendungen	18
Veröffentlichen einer Anwendung (AWS CLI)	19
Veröffentlichen einer neuen Anwendung (Konsole)	19
Freigeben einer Anwendung	26
Aufheben der Freigabe einer Anwendung	28
Löschen einer Anwendung	30
Veröffentlichen von neuen Anwendungsversionen	31
Verifiziertes Autoren-Badge	32
Anfordern eines verifizierten Autoren-Badge	32
Freigeben von Lambda-Ebenen	33
So funktioniert es	33
Beispiel	34
Bereitstellen von Anwendungen	35
Anwendungsbereitstellungsberechtigungen	35
Anwendungsfunktionen	36
Suchen und Bestätigen von Anwendungsfunktionen (Konsole)	37
Anzeigen von Anwendungsfunktionen (AWS CLI)	37

So stellen Sie Anwendungen bereit	38
Bereitstellen einer neuen Anwendungsversion (Konsole)	38
Bereitstellen einer neuen Anwendungsversion (AWS CLI)	40
Löschen von Anwendungs-Stacks	41
Aktualisieren von Anwendungen	41
Sicherheit	43
Datenschutz	44
Verschlüsselung während der Übertragung	45
Verschlüsselung im Ruhezustand	45
Identitäts- und Zugriffsverwaltung	45
Zielgruppe	46
Authentifizieren mit Identitäten	47
Verwalten des Zugriffs mit Richtlinien	50
Wie das mit IAMAWS Serverless Application Repository funktioniert	53
Beispiele für identitätsbasierte Richtlinien	59
Beispiele für Benutzerrichtlinien	68
Referenztable für AWS Serverless Application Repository-API-Berechtigungen	74
Fehlerbehebung	78
Protokollieren und überwachen	81
Protokollieren von AWS Serverless Application Repository-API-Aufrufen mit AWS CloudTrail	81
Compliance-Validierung	85
Ausfallsicherheit	86
Sicherheit der Infrastruktur	86
Kontingente	88
Fehlerbehebung	89
Sie können eine Anwendung nicht veröffentlichen	89
Ein Kontingent wurde überschritten	90
Eine aktualisierte Readme-Datei erscheint nicht sofort	90
Sie können eine Anwendung infolge von unzureichenden IAM-Berechtigungen nicht bereitstellen	90
Sie können eine Anwendung nicht zweimal bereitstellen	90
Warum ist meine Anwendung nicht öffentlich verfügbar	91
Den -Support kontaktieren	91
Operationen	92
Ressourcen	94

Applications	94
URI	94
HTTP-Methoden	94
Schemata	96
Eigenschaften	100
Weitere Informationen finden Sie auch unter	118
applicationId	119
URI	119
HTTP-Methoden	119
Schemata	123
Eigenschaften	126
Weitere Informationen finden Sie auch unter	140
Application-ID-Changesets für Anwendungen	141
URI	141
HTTP-Methoden	141
Schemata	143
Eigenschaften	145
Weitere Informationen finden Sie auch unter	153
Applications applicationId Dependencies	153
URI	153
HTTP-Methoden	153
Schemata	155
Eigenschaften	157
Weitere Informationen finden Sie auch unter	160
ApplicationID-Richtlinie für Anwendungen	161
URI	161
HTTP-Methoden	161
Schemata	163
Eigenschaften	165
Weitere Informationen finden Sie auch unter	169
Applications applicationId Templates	170
URI	170
HTTP-Methoden	170
Schemata	171
Eigenschaften	173
Weitere Informationen finden Sie auch unter	177

Applications applicationId Templates templateId	178
URI	178
HTTP-Methoden	178
Schemata	180
Eigenschaften	181
Weitere Informationen finden Sie auch unter	185
Applications applicationId Unshare	186
URI	186
HTTP-Methoden	186
Schemata	187
Eigenschaften	189
Weitere Informationen finden Sie auch unter	191
ApplicationID-Versionen von Anwendungen	192
URI	192
HTTP-Methoden	192
Schemata	194
Eigenschaften	195
Weitere Informationen finden Sie auch unter	199
Anwendungen ApplicationID-Versionen SemanticVersion	200
URI	200
HTTP-Methoden	200
Schemata	201
Eigenschaften	204
Weitere Informationen finden Sie auch unter	213
Dokumentverlauf	215
AWS-Glossar	220
.....	ccxxi

Was ist das AWS Serverless Application Repository?

Das AWS Serverless Application Repository erleichtert es Entwicklern und Unternehmen, serverlose Anwendungen schnell und einfach in der AWS Cloud. Weitere Informationen zu serverlosen Anwendungen finden Sie unter [Serverloses Computing und Anwendungen](#) auf der AWS Website.

Sie können Anwendungen einfach veröffentlichen, indem Sie sie für die ganze Community oder privat für Ihr Team oder in Ihrer Organisation freigeben. Zur Veröffentlichung einer serverlosen Anwendung (oder App) können Sie die AWS Management Console, die AWS SAM-Befehlszeilenschnittstelle (AWS SAM CLI) oder AWS SDKs zum Hochladen Ihres Codes. Zusammen mit Ihrem Code laden Sie eine einfache Manifestdatei hoch, die auch AWS Serverless Application Model (AWS SAM)-Vorlage genannt wird. Weitere Informationen über AWS SAM finden Sie im [AWS Serverless Application Model-Entwicklerleitfaden](#).

Das AWS Serverless Application Repository ist hochgradig mit der AWS Lambda-Konsole integriert. Diese Integration bedeutet, dass Entwickler aller Stufen mit serverlosem Computing beginnen können, ohne etwas Neues erlernen zu müssen. Sie können mithilfe von Kategorie-Schlüsselwörtern nach Anwendungen wie mobilen und Web-Backends, Anwendungen zur Datenverarbeitung oder Chatbots suchen. Außerdem können Sie über Name, Herausgeber oder Ereignisquelle nach Anwendungen suchen. Wenn Sie eine Anwendung verwenden möchten, wählen Sie sie einfach aus, konfigurieren Sie die erforderlichen Felder und stellen Sie sie mit ein paar Klicks bereit.

In diesem Leitfaden erfahren Sie mehr zu den beiden Möglichkeiten zur Arbeit mit dem AWS Serverless Application Repository:

- [Veröffentlichen von Anwendungen](#)— Sie können Anwendungen konfigurieren und hochladen, um sie anderen Entwicklern verfügbar zu machen, und neue Anwendungsversionen veröffentlichen.
- [Bereitstellen von Anwendungen](#)- Suchen Sie nach Anwendungen und zeigen Sie Informationen, wie Quellcode und Readme-Dateien, dazu an. Außerdem können Sie Anwendungen Ihrer Wahl installieren, konfigurieren und bereitstellen.

Nächste Schritte

- Für ein Tutorial zum Veröffentlichen einer Beispielanwendung im AWS Serverless Application Repository, siehe [Quick Start: Veröffentlichen von Anwendungen](#) aus.
- Anweisungen zum Bereitstellen von Anwendungen aus dem AWS Serverless Application Repository, siehe [So stellen Sie Anwendungen bereit](#) aus.

Quick Start: Veröffentlichen von Anwendungen

Dieser Leitfaden führt Sie durch die Schritte zum Herunterladen, Erstellen, Testen und Veröffentlichen einer serverlosen Beispielanwendung im AWS Serverless Application Repository mithilfe der AWS SAM-CLI. Sie können diese Beispielanwendung als Ausgangspunkt für die Entwicklung und Veröffentlichung Ihrer eigenen serverlosen Anwendung verwenden.

Übersicht

In den folgenden Schritten wird beschrieben, wie Sie eine serverlose Beispielanwendung herunterladen, erstellen und veröffentlichen:

1. Initialisieren. Laden Sie eine Beispielanwendung aus der Vorlage mithilfe von `sam init` herunter.
2. Ausführen lokaler Tests. Testen Sie die Anwendung lokal mit `sam local invoke` und/oder `sam local start-api`. Beachten Sie, dass Ihre Lambda-Funktion mit diesen Befehlen zwar lokal aufgerufen wird, aber aus AWS Ressourcen in der AWS Cloud liest und in sie schreibt.
3. Paket. Wenn Sie mit Ihrer Lambda-Funktion zufrieden sind, bündeln Sie die Lambda-Funktion, die AWS SAM Vorlage und alle Abhängigkeiten in einem AWS CloudFormation Bereitstellungspaket mit `sam package`. In diesem Schritt fügen Sie auch Informationen über die Anwendung hinzu, die in das AWS Serverless Application Repository hochgeladen wird.
4. Veröffentlichen. Veröffentlichen Sie die Anwendung im AWS Serverless Application Repository mithilfe von `sam publish`. Am Ende dieses Schritts können Sie Ihre Anwendung in der AWS Cloud anzeigen AWS Serverless Application Repository und sie mithilfe von verwenden AWS Serverless Application Repository.

Das Beispiel [Hello World-Anwendung](#) im nächsten Abschnitt führt Sie durch diese Schritte beim Erstellen und Veröffentlichen einer serverlosen Anwendung.

Hello World-Anwendung

In dieser Übung laden Sie eine serverlose Hello World-Anwendung, die ein einfaches API-Backend darstellt, herunter und testen Sie diese. Es hat einen Amazon-API-Gateway-Endpunkt, der eine GET-Operation und eine Lambda-Funktion unterstützt. Das heißt, wenn eine GET-Anfrage an den Endpunkt gesendet wird, ruft API Gateway die Lambda-Funktion auf. Dann führt AWS Lambda die Funktion aus, die einfach eine `hello world`-Nachricht zurückgibt.

Die Anwendung besteht aus folgenden Komponenten:

- Eine AWS SAM Vorlage, die zwei AWS Ressourcen für die Hello World-Anwendung definiert: einen API-Gateway-Dienst mit einer GET-Operation und eine Lambda-Funktion. Die Vorlage definiert auch die Zuordnung zwischen dem API-Gateway-GET-Vorgang und der Lambda-Funktion.
- Anwendungscode, der in Python geschrieben ist.

Bevor Sie beginnen

Stellen Sie sicher, dass Sie über die erforderliche Einrichtung für diese Übung verfügen:

- Sie müssen ein AWS Konto bei einem IAM-Benutzer Administratorberechtigungen haben Administratorberechtigungen besitzen. Siehe [AWS Konto einrichten](#).
- Sie müssen die AWS SAM-CLI (Command Line Interface, Befehlszeilenschnittstelle) installiert haben. Siehe [Installieren der AWS SAM-CLI](#).
- Sie müssen Version 1.16.77 oder höher der AWS CLI installiert haben. Siehe [Installieren der AWS Command Line Interface](#).

Schritt 1: Initialisieren der Anwendung

In diesem Abschnitt laden Sie die Beispielanwendung herunter, die aus einer AWS SAM-Vorlage und einem Anwendungscode besteht.

So initialisieren Sie die Anwendung

1. Führen Sie den folgenden Befehl bei einer AWS SAM-CLI-Eingabeaufforderung aus.

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

2. Überprüfen Sie den Inhalt des Verzeichnisses, das der Befehl erstellt hat (sam-app/):
 - `template.yaml`— Definiert zwei AWS Ressourcen, die die Hello World-Anwendung benötigt: eine Lambda-Funktion und einen API-Gateway-Endpunkt, der eine GET-Operation unterstützt. Die Vorlage definiert auch die Zuweisung zwischen den beiden Ressourcen.
 - Inhalt im Zusammenhang mit dem Hello World-Anwendungscode:
 - `hello_world/`Verzeichnis — Enthält den Anwendungscode, der zurückgegeben wird, `hello world` wenn Sie ihn ausführen.

Note

Für diese Übung wird der Anwendungscode in Python geschrieben und Sie geben die Laufzeit im Befehl `init` an. AWS Lambda unterstützt zusätzliche Sprachen zum Erstellen von Anwendungscode. Wenn Sie eine andere unterstützte Laufzeit angeben, enthält der `init`-Befehl den Hello World-Code in der angegebenen Sprache und eine `README.md`-Datei, die Sie für diese Sprache befolgen können. Informationen zu unterstützten Laufzeiten finden Sie unter [Lambda-Ausführungsumgebung und verfügbare Bibliotheken](#).

Schritt 2: Lokales Testen der Anwendung

Nachdem Sie die AWS SAM-Anwendung nun auf Ihrem lokalen Computer installiert haben, führen Sie die folgenden Schritte aus, um sie lokal zu testen.

So testen Sie die Anwendung lokal

1. Starten Sie den API-Gateway-Endpunkt lokal. Sie müssen den folgenden Befehl aus dem Verzeichnis ausführen, das die `template.yaml`-Datei enthält.

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

Der Befehl gibt einen API-Gateway-Endpunkt zurück, an den Sie Anfragen für lokale Tests senden können.

2. Testen Sie die Anwendung. Kopieren Sie die API Gateway Gateway-Endpunkt-URL, fügen Sie sie in den Browser ein und wählen Sie Enter. Ein Beispiel für eine API-Gateway-Endpunkt-URL lautet `http://127.0.0.1:3000/hello`.

API Gateway ruft lokal die Lambda-Funktion auf, der der Endpunkt zugeordnet ist. Die Lambda-Funktion wird im lokalen Docker-Container ausgeführt und kehrt zurück `hello world`. API Gateway gibt eine Antwort an den Browser zurück, die den Text enthält.

Übung: Ändern der Nachrichtenzeichenfolge

Nachdem Sie die Beispielanwendung erfolgreich getestet haben, können Sie mit einer einfachen Änderung experimentieren: Ändern Sie die zurückgegebene Nachrichtenzeichenfolge.

1. Bearbeiten Sie die `/hello_world/app.py`-Datei, um die Nachrichtenzeichenfolge von `'hello world'` in `'Hello World!'` zu ändern.
2. Laden Sie die Test-URL in Ihrem Browser neu und beachten Sie die neue Zeichenfolge.

Sie werden feststellen, dass Ihr neuer Code dynamisch geladen wird, ohne dass Sie den Prozess `sam local` neu starten müssen.

Schritt 3: Verpacken der Anwendung

Nachdem Sie Ihre Anwendung lokal getestet haben, verwenden Sie die AWS SAM-CLI, um ein Bereitstellungspaket und eine verpackte AWS SAM-Vorlage zu erstellen.

Note

In den folgenden Schritten erstellen Sie eine ZIP-Datei für den Inhalt des `hello_world/`-Verzeichnisses, das den Anwendungscode enthält. Diese ZIP-Datei ist das Bereitstellungspaket für Ihre serverlose Anwendung. Weitere Informationen finden Sie unter [Erstellen eines Deployment-Pakets \(Python\)](#) im AWS LambdaEntwicklerhandbuch.

So erstellen Sie ein Lambda-Bereitstellungspaket

1. Fügen Sie Ihrer AWS SAM-Vorlagendatei einen `Metadata`-Abschnitt hinzu, der die erforderlichen Anwendungsinformationen enthält. Weitere Informationen zum Abschnitt `Metadata` der AWS SAM-Vorlagen finden Sie unter [AWS SAM Template Metadata Section Properties](#) im AWS Serverless Application Model-Entwicklerhandbuch.

Hier ist ein `Metadata`-Beispielabschnitt:

```
Metadata:
  AWS::ServerlessRepo::Application:
    Name: my-app
    Description: hello world
    Author: user1
    SdxLicenseId: 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
```

Die `ReadmeUrl` Eigenschaften `LicenseUrl` und können entweder Verweise auf lokale Dateien sein (wie im obigen Beispiel), oder sie können Links zu Amazon S3 S3-Buckets sein, die diese Artefakte bereits hosten.

- Erstellen Sie einen S3-Bucket an dem Speicherort, an dem Sie den verpackten Code speichern möchten. Wenn Sie einen vorhandenen S3-Bucket verwenden möchten, überspringen Sie diesen Schritt.

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

- Erstellen Sie das Bereitstellungspaket für Lambda-Funktionen, indem Sie den folgenden `packageAWS SAM CLI`-Befehl ausführen.

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

Der Befehl hat folgende Auswirkungen:

- Zippt den Inhalt des `saws-sam/hello_world/` Verzeichnisses und lädt ihn auf Amazon S3 hoch.
- Lädt das Bereitstellungspaket, die README-Datei und die LICENSE-Datei in den Amazon S3 S3-Bucket hoch, der durch die `--s3-bucket` Option angegeben wurde.
- Gibt eine neue Vorlagendatei mit dem Namen `packaged.yaml` aus, die Sie im nächsten Schritt zum Veröffentlichen der Anwendung im AWS Serverless Application Repository verwenden. Die `packaged.yaml` Vorlagendatei ähnelt der ursprünglichen Vorlagendatei (`template.yaml`), weist jedoch einen wesentlichen Unterschied auf: Die Eigenschaften `CodeUri`, `LicenseUrl`, und die `ReadmeUrl` Eigenschaften verweisen auf den Amazon S3 S3-Bucket und die Objekte, die die entsprechenden Artefakte enthalten. Der folgende Ausschnitt aus einer `packaged.yaml`-Beispielvorlagendatei zeigt die `CodeUri`-Eigenschaft:

```
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/fbd77a3647a4f47a352fc0bjectGUID
```

```
  ...
```

Schritt 4: Veröffentlichen der Anwendung

Nachdem Sie das Bereitstellungspaket erstellt haben, verwenden Sie es, um die Anwendung im AWS Serverless Application Repository zu veröffentlichen.

So veröffentlichen Sie die serverlose Anwendung auf dem AWS Serverless Application Repository

- Führen Sie den folgenden Befehl aus, um die neue Anwendung mit der ersten Version in AWS Serverless Application Repository zu veröffentlichen, die als 0.0.1 erstellt wurde.

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

Note

Die Anwendung wird standardmäßig als privat erstellt. Sie müssen die Anwendung teilen, bevor andere AWS-Konten Ihre Anwendung anzeigen und bereitstellen können. Weitere Informationen zum Freigeben Ihrer Anwendung finden Sie nachstehend in den nächsten Schritten.

Nächste Schritte

Nachdem Sie Ihre Beispielanwendung veröffentlicht haben, folgen einige Dinge, die Sie möglicherweise damit tun möchten.

- Ihre Bewerbung anzeigen in AWS Serverless Application Repository — Die Ausgabe des `aws sam publish` Befehls enthält einen Link, der AWS Serverless Application Repository direkt zur Detailseite Ihrer Anwendung führt. Sie können auch zur AWS Serverless Application Repository-Landingpage gehen und nach Ihrer Anwendung suchen.
- Teilen Sie Ihre Anwendung — Da Ihre Anwendung standardmäßig auf `private` gesetzt ist, ist sie für andere AWS Konten nicht sichtbar. Um Ihre Bewerbung mit anderen zu teilen, müssen Sie sie entweder veröffentlichen oder einer bestimmten Liste von AWS Konten die Erlaubnis erteilen. Weitere Informationen zum Freigeben Ihrer Anwendung mithilfe der AWS CLI finden Sie unter [AWS Serverless Application Repository Beispiele für Anwendungsrichtlinien](#). Informationen zum Freigeben Ihrer Anwendung über die Konsole finden Sie unter [Freigeben einer Anwendung](#).

Weitere Informationen

Weitere Informationen zum Abschnitt `Metadata` der AWS SAM-Vorlagen, die Befehle `aws sam package` und `aws sam publish` der AWS SAM-CLI finden Sie unter [Veröffentlichen von Anwendungen mithilfe der AWS SAM-CLI](#) im AWS Serverless Application Model-Entwicklerhandbuch.

Veröffentlichen von Anwendungen

Wenn Sie eine serverlose Anwendung im AWS Serverless Application Repository veröffentlichen, ermöglichen Sie anderen, diese zu suchen und bereitzustellen.

Zuerst definieren Sie Ihre Anwendung mit einer AWS Serverless Application Model (AWS SAM)-Vorlage. Wenn Sie Ihre Anwendung definieren, müssen Sie überlegen, ob Benutzer Ihrer Anwendung die Funktionen der Anwendung bestätigen müssen. Weitere Informationen zum Verwenden von AWS SAM und Bestätigen von Funktionen finden Sie unter [Verwenden von AWS SAM mit der AWS Serverless Application Repository](#).

Sie können serverlose Anwendungen über die AWS Management Console, die AWS SAM-Befehlszeilenschnittstelle (AWS SAM CLI) oder ein AWS-SDK. Weitere Informationen zu den Verfahren zum Veröffentlichen von Anwendungen im AWS Serverless Application Repository finden Sie unter [So veröffentlichen Sie Anwendungen](#).

Wenn Sie Ihre Anwendung veröffentlichen, ist sie zunächst auf `Privat`, was bedeutet, dass es nur für die AWS-Konto, das es erstellt hat. Um Ihre Bewerbung mit anderen zu teilen, müssen Sie sie entweder auf `Privat freigegeben` (freigegeben nur für einen spezifischen Satz von AWS-Konten) oder `Öffentlich freigegeben` (mit allen geteilt).

Wenn Sie eine Anwendung im AWS Serverless Application Repository veröffentlichen und auf „öffentlich“ festlegen, stellt der Dienst die Anwendung den Verbrauchern in allen Regionen zur Verfügung. Wenn ein Verbraucher eine öffentliche Anwendung in einer anderen Region als der Region bereitstellt, in der die Anwendung erstmals veröffentlicht wurde, kopiert AWS Serverless Application Repository die Bereitstellungsartefakte der Anwendung in einen Amazon S3 S3-Bucket in der Zielregion. Es aktualisiert alle Ressourcen in der AWS SAM-Vorlage, die diese Artefakte verwenden, um stattdessen auf die Dateien im Amazon S3 S3-Bucket für die Zielregion zu verweisen. Bereitstellungsartefakte können Lambda-Funktionscode, API-Definitionsdateien usw. umfassen.

Note

`Privat` und `Privat freigegeben` Anwendungen sind nur in der AWS-Region, in der sie geschaffen wurden. `Öffentlich freigegeben` Anwendungen sind in allen verfügbaren AWS-Regionen. Weitere Informationen zum Freigeben von Anwendungen finden Sie unter [AWS Serverless Application Repository Beispiele für Anwendungsrichtlinien](#).

Themen

- [Verwenden von AWS SAM mit der AWS Serverless Application Repository](#)
- [So veröffentlichen Sie Anwendungen](#)
- [Verifiziertes Autoren-Badge](#)
- [Freigeben von Lambda-Ebenen](#)

Verwenden von AWS SAM mit der AWS Serverless Application Repository

Die AWS Serverless Application Model (AWS SAM) ist ein Open-Source-Framework, mit dem Sie erstellen können [serverlose Anwendungen](#) auf AWS aus. Weitere Informationen zur Verwendung von AWS SAM zum Erstellen Ihrer serverlosen Anwendung finden Sie im [AWS Serverless Application Model-Entwicklerhandbuch](#).

Beim Erstellen von Anwendungen, die im AWS Serverless Application Repository, müssen Sie das Set von unterstützten AWS verfügbaren Ressourcen und Richtlinienvorlagen aus. In den folgenden Abschnitten werden diese Themen ausführlicher beschrieben.

Unterstützte AWS Ressourcen im AWS Serverless Application Repository

Das AWS Serverless Application Repository unterstützt serverlose Anwendungen, die aus vielen AWS SAM- und AWS CloudFormation-Ressourcen bestehen. So sehen Sie die vollständige Liste von AWS Ressourcen, die unterstützt werden von AWS Serverless Application Repository, finden Sie unter [Liste der in unterstützten AWS Ressourcen](#) aus.

Wenn Sie Support für eine zusätzliche AWS Ressource, kontaktieren Sie [AWS-Support](#) aus.

Important

Wenn Ihre Anwendungsvorlage eine der folgenden benutzerdefinierten IAM-Rollen oder Ressourcenrichtlinien enthält, wird die Anwendung standardmäßig nicht in den Suchergebnissen aufgeführt. Darüber hinaus müssen Kunden die benutzerdefinierten IAM-Rollen und Ressourcenrichtlinien der Anwendung bestätigen, ehe sie die Anwendung bereitstellen können. Weitere Informationen finden Sie unter [Bestätigen von Anwendungsfunktionen](#).

Die Liste der Ressourcen, für die dieses gilt, sieht wie folgt aus:

- IAM-Rollen: [AWS::IAM::Group](#), [AWS::IAM::InstanceProfile](#), [AWS::IAM::Policy](#), und [AWS::IAM::Role](#) aus.
- Ressourcenrichtlinien: [AWS::Lambda::LayerVersionBerechtigung](#), [AWS::Lambda::Permission](#), [AWS::Events::EventBus--Richtlinie](#), [AWS::IAM::Policy](#), [AWS::ApplicationAutoScaling::ScalingPolicy](#), [AWS::S3::BucketPolicy](#), [AWS::SQS::QueuePolicy](#), und [AWS::SNS::TopicPolicy](#) aus.

Wenn Ihre Anwendung die Ressource [AWS::Serverless::Application](#) enthält, müssen Kunden bestätigen, dass die Anwendung eine verschachtelte Anwendung enthält, ehe sie sie bereitstellen können. Weitere Informationen über verschachtelte Anwendungen finden Sie im Thema [Verschachtelte Anwendungen](#) im AWS Serverless Application Model-Entwicklerhandbuch. Weitere Informationen zur Funktionsbestätigung finden Sie unter [Bestätigen von Anwendungsfunktionen](#).

Richtlinienvorlagen

AWS SAM bietet Ihnen eine Liste von Richtlinienvorlagen, um den Umfang der Berechtigungen Ihrer Lambda-Funktionen für die von der Anwendung verwendeten Ressourcen festzulegen. Für die Verwendung von Richtlinienvorlagen sind keine zusätzlichen Kundenbestätigungen erforderlich, um die Anwendung zu durchsuchen oder bereitzustellen.

Für die Liste der Standard AWS SAM Richtlinienvorlagen siehe [AWS SAM-Richtlinienvorlagen](#) im [AWS Serverless Application Model-Entwicklerhandbuch](#) aus.

Liste der unterstützten AWS Ressourcen

Dies ist die vollständige Liste AWS Ressourcen, die von der AWS Serverless Application Repository aus.

- `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::VPCElasticNetworkInterfaceAttachment`
- `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`

So veröffentlichen Sie Anwendungen

In diesem Abschnitt finden Sie Verfahren zum Veröffentlichen der serverlosen Anwendung im AWS Serverless Application Repository mithilfe der AWS SAM-CLI oder der AWS Management Console.

Außerdem wird gezeigt, wie Sie Ihre Anwendung freigeben, damit andere Benutzer sie bereitstellen können, und aus dem AWS Serverless Application Repository löschen.

Important

Die Informationen, die Sie beim Veröffentlichen einer Anwendung eingeben, sind nicht verschlüsselt. Diese Informationen umfassen Daten wie den Namen des Autors. Wenn Sie nicht möchten, dass personenbezogene Daten gespeichert oder veröffentlicht werden, empfehlen wir, solche Informationen bei der Veröffentlichung Ihrer Anwendung nicht einzugeben.

Veröffentlichen einer Anwendung (AWS CLI)

Die einfachste Möglichkeit, eine Anwendung im AWS Serverless Application Repository zu veröffentlichen und freizugeben, besteht darin, eine Reihe von AWS SAM-CLI-Befehlen zu verwenden. Weitere Informationen finden Sie unter [Veröffentlichen einer Anwendung mithilfe der AWS SAM-CLI](#) im AWS Serverless Application Model (AWS SAM)-Entwicklerhandbuch.

Veröffentlichen einer neuen Anwendung (Konsole)

In diesem Abschnitt erfahren Sie, wie Sie die AWS Management Console verwenden können, um eine neue Anwendung im AWS Serverless Application Repository zu veröffentlichen. Anweisungen zum Veröffentlichen einer neuen Version einer vorhandenen Anwendung finden Sie unter [Veröffentlichen der neuen Version einer vorhandenen Anwendung](#).

Voraussetzungen

Vor der Veröffentlichung einer Anwendung im AWS Serverless Application Repository benötigen Sie Folgendes:

- Eine gültige AWS-Konto.
- Eine gültige AWS Serverless Application Model (AWS SAM) -Vorlage, die die AWS-Ressourcen, die verwendet werden. Weitere Informationen zu AWS SAM-Vorlagen finden Sie unter [AWS SAM-Vorlagengrundlagen](#).
- Ein Paket für Ihre Anwendung, das Sie mit dem AWS CloudFormation-Befehl `aws cloudformation package` für die AWS CLI erstellt haben. Mit diesem Befehl werden lokale Artefakte (lokale Pfade) verpackt

werden, auf die Ihre AWS SAM-Vorlage verweist. Weitere Details finden Sie unter [Paket](#) in der AWS CloudFormation-Dokumentation.

- Eine URL, die auf den Quellcode Ihrer Anwendung verweist, falls Sie Ihre Anwendung öffentlich veröffentlichen möchten.
- Eine readme.txt-Datei. In dieser Datei sollte beschrieben werden, wie Kunden Ihre Anwendung verwenden können und wie sie konfiguriert wird, bevor sie selbst bereitgestellt werden AWS-Konten.
- Eine license.txt-Datei oder eine gültige Lizenzkennung von der [SPDX-Website](#). Beachten Sie, dass eine Lizenz nur erforderlich ist, wenn Sie Ihre Anwendung öffentlich freigeben möchten. Wenn Sie Ihre Anwendung privat halten oder nur privat freigeben möchten, müssen Sie keine Lizenz angeben.
- Eine gültige Amazon S3-Bucket-Richtlinie, die dem Service Leseberechtigungen für Artefakte erteilt, die beim Verpacken Ihrer Anwendung zu Amazon S3 hochgeladen wurden. Gehen Sie folgendermaßen vor, um diese Richtlinie festzulegen:
 1. Öffnen Sie die Amazon-S3-Konsole unter <https://console.aws.amazon.com/s3/>.
 2. Wählen Sie den Amazon S3 S3-Bucket aus, den Sie zum Verpacken Ihrer Anwendung verwendet haben.
 3. Wählen Sie die Registerkarte Permissions (Berechtigungen).
 4. Wählen Sie Bucket Policy (Bucket-Richtlinie) aus.
 5. Fügen Sie die folgende Richtlinienanweisung in den Bucket-Richtlinien-Editor ein. Stellen Sie sicher, dass Sie Ihren Bucket-Namen in der `ResourceElement`, und dein AWS-Konto-ID in der `Conditionelement`. Der Ausdruck im `Conditionelement` sicherstellen AWS Serverless Application Repository hat nur die Berechtigung, auf Anwendungen von den angegebenen AWS-Konto. Weitere Informationen zu Richtlinien-Erklärungen finden Sie unter [IAM-JSON-Richtlinienelementreferenz](#) im IAM User Guide aus.

```
{
  "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"
        }
    }
}
]
}

```

- Wählen Sie die Schaltfläche Save (Speichern) aus.

Verfahren

Erstellen Sie eine neue Anwendung mithilfe des folgenden Vorgangs im AWS Serverless Application Repository.

So erstellen Sie eine neue Anwendung im AWS Serverless Application Repository

- Öffnen Sie die [AWS Serverless Application Repository-Konsole](#) und wählen Sie Publish applications (Anwendungen veröffentlichen) aus.
- Geben Sie auf der Seite Publish an application (Anwendung veröffentlichen) die folgenden Anwendungsinformationen ein und wählen Sie dann Publish application (Anwendung veröffentlichen) aus:

Property (Eigenschaft)	Erforderlich	Beschreibung
Anwendungsname	TRUE	Der Name der Anwendung. Minimale Länge = 1. Maximale Länge = 140. Pattern: "[a-zA-Z0-9\-\-]+";
Autor	TRUE	Der Name des Autors, der die Anwendung veröffentlicht. Minimale Länge = 1. Maximale Länge = 127.

Property (Eigenschaft)	Erforderlich	Beschreibung
		Pattern: <code>"^[a-z0-9]([a-z0-9]-(?!-))*[a-z0-9]?\$"</code> ;
Homepage	FALSE	Eine URL mit weiteren Informationen über die Anwendung — z. B. GitHub Repository für die Anwendung.
Beschreibung	TRUE	Die Beschreibung der Anwendung. Minimale Länge = 1. Maximale Länge = 256.
Bezeichnungen	FALSE	Die Beschriftungen, die die Erkennung von Anwendungen in Suchergebnissen verbessern. Minimale Länge = 1. Maximale Länge = 127. Maximale Anzahl der Etiketten: 10. Pattern: <code>"^[a-zA-Z0-9+\\-._:\\@]+ \$"</code> ;

Property (Eigenschaft)	Erforderlich	Beschreibung
Spdx-Lizenz (Dropdown-Liste)	FALSE	<p>Wählen Sie eine gültige Lizenzkennung aus der Dropdown-Liste aus, die Lizenzen enthält, die auf der SPDX-Website verfügbar sind. Wenn Sie ein Element in der Dropdown-Liste auswählen, wird das Textfeld Lizenz darunter aufgefüllt. Hinweis: Wenn Sie eine Lizenz im Dropdown-Liste auswählen, wird der Inhalt der LicenseTextfeld und verwirft alle manuellen Änderungen, die Sie vorgenommen haben.</p>

Property (Eigenschaft)	Erforderlich	Beschreibung
License	FALSE	<p>Laden Sie eine TXT-Lizenzdatei hoch oder wählen Sie eine Lizenz aus dem Dropdown-Menü <code>SPDX license (SPDX-Lizenz)</code>, das in der vorherigen Zeile beschrieben ist. Wenn Sie eine Lizenz aus dem Dropdown-Menü <code>SPDX license (SPDX-Lizenz)</code> auswählen, wird automatisch das Textfeld <code>Lizenz</code> ausgefüllt. Sie können den Inhalt dieses Textfelds manuell bearbeiten, nachdem Sie eine Lizenzdatei hochgeladen haben oder eine aus dem Dropdown-Menü <code>Spdx-Lizenz</code> auswählen. Wenn jedoch eine andere <code>Spdx-Lizenz</code> aus dem Dropdown-Menü ausgewählt wird, werden alle manuellen Änderungen, die Sie vorgenommen haben, verworfen.</p> <p>Dies ist ein optionales Feld, aber Sie müssen eine Lizenz bereitstellen, um die Anwendung öffentlich freigeben zu können.</p>

Property (Eigenschaft)	Erforderlich	Beschreibung
Readme	FALSE	Laden Sie den Inhalt der Readme-Datei hoch, die im Text- oder Markdown-Format vorliegen kann. Diese Inhalte werden auf der Detailseite der Anwendung im AWS Serverless Application Repository angezeigt. Sie können den Inhalt dieses Textfelds nach dem Hochladen einer Datei manuell bearbeiten.
Semantic version (Semantische Version)	FALSE	Die semantische Version der Anwendung. Weitere Informationen finden Sie auf der Semantic Versioning-Website . Sie müssen einen Wert für diese Eigenschaft angeben, um Ihre Anwendung öffentlich zu machen.
Quellcode-URL	FALSE	Ein Link zu einem öffentlichen Repository für den Quellcode Ihrer Anwendung.
SAM-Vorlage	TRUE	Eine gültige AWS Serverless Application Model (AWS SAM) -Vorlage, die die AWS Ressourcen, die verwendet werden.

Freigeben einer Anwendung

Für veröffentlichte Anwendungen können Berechtigungen in einer der drei folgenden Kategorien festgelegt werden:

- **Privat (Standardeinstellung)**— Anwendungen, die mit demselben Konto erstellt wurden und für kein anderes freigegeben wurdenAWSKonto. Nur Verbraucher, die Ihre teilenAWS-Konto hat die Berechtigung, private Anwendungen bereitzustellen.
- **Privat freigegeben**— Anwendungen, die der Herausgeber explizit für einen bestimmten Satz vonAWSKonten oder mitAWSKonten in einemAWSOrganisation. Verbraucher haben die Berechtigung, Anwendungen bereitzustellen, die für ihre freigegeben wurdenAWSkonto oderAWSOrganisation. Weitere Informationen zu AWS Organizations finden Sie im [AWS Organizations Benutzerhandbuch](#).
- **Öffentlich freigegeben**— Anwendungen, die der Herausgeber für alle freigegeben hat. Alle Verbraucher haben die Berechtigung, alle öffentlich freigegebenen Anwendungen bereitzustellen.

Nachdem Sie eine Anwendung im AWS Serverless Application Repository veröffentlicht haben, ist sie standardmäßig auf privat festgelegt. In diesem Abschnitt wird beschrieben, wie Sie eine Anwendung privat für bestimmte freigegebenAWSKonten oder einAWSOrganisation oder teilen Sie es öffentlich mit allen.


Freigeben einer Anwendung über die Konsole

Es gibt zwei Möglichkeiten, Ihre Anwendung für andere freizugeben: 1) Teilen Sie es mit bestimmtenAWSKonten oder dasAWSKonten in IhremAWSOrganisation oder 2) Teilen Sie es öffentlich mit allen. Weitere Informationen zu AWS Organizations finden Sie im [AWS Organizations Benutzerhandbuch](#).

Option 1: So teilen Sie Ihre Bewerbung mit bestimmtenAWSKonten oder Konten in IhremAWSOrganisation

1. Öffnen Sie die [AWS Serverless Application Repository-Konsole](#).
2. Wählen Sie im Navigationsbereich Published Applications (Veröffentlichte Anwendungen) aus, um eine Liste der von Ihnen erstellten Anwendungen anzuzeigen.
3. Wählen Sie die Anwendung aus, die Sie freigeben möchten.
4. Wählen Sie die Registerkarte Sharing (Freigabe).

5. Wählen Sie im Abschnitt Application policy statements (Anwendungsrichtlinienanweisungen) die Schaltfläche Create statement (Anweisung erstellen).
6. Füllen Sie im Fenster Statement Configuration (Anweisungskonfiguration) die Felder aus, die darauf basieren, wie Sie Ihre Anwendung freigeben möchten.

 Note


Wenn Sie sie für eine Organisation freigeben, können Sie nur die Organisation angeben, die AWSaccount ist Mitglied von. Wenn Sie versuchen, eine AWSOrganisation, in der Sie kein Mitglied sind, tritt ein Fehler auf.

So teilen Sie Ihre Bewerbung mit Ihrer AWSOrganisation müssen Sie anerkennen, dass die UnshareApplicationDie -Aktion wird Ihrer Richtlinienerklärung hinzugefügt, falls die Freigabe in future widerrufen werden muss.

7. Wählen Sie die Schaltfläche Save (Speichern) aus.

Option 2: So geben Sie Ihre Bewerbung öffentlich für alle frei

1. Öffnen Sie die [AWS Serverless Application Repository-Konsole](#).
2. Wählen Sie im Navigationsbereich Published Applications (Veröffentlichte Anwendungen) aus, um eine Liste der von Ihnen erstellten Anwendungen anzuzeigen.
3. Wählen Sie die Anwendung aus, die Sie freigeben möchten.
4. Wählen Sie die Registerkarte Sharing (Freigabe).
5. Wählen Sie im Abschnitt Public Sharing (Öffentliche Freigabe) die Schaltfläche Edit (Bearbeiten).
6. Wählen Sie unter Public sharing (Öffentliche Freigabe) das Optionsfeld Enabled (Aktiviert) aus.
7. Geben Sie im Textfeld den Namen Ihrer Anwendung ein und wählen Sie dann die Schaltfläche Save (Speichern) aus.

 Note

Um eine Anwendung öffentlich freigeben zu können, müssen die Eigenschaften SemanticVersion und LicenseUrl festgelegt sein.

Freigeben einer Anwendung über die AWS CLI

So geben Sie eine Anwendung mithilfe der AWS CLI. Sie erteilen Berechtigungen mit dem [put-application-policy](#) Befehl zur Angabe des AWS-Kontos, die Sie als Prinzipale freigeben möchten.

Weitere Informationen zum Teilen Ihrer Anwendung mithilfe der AWS CLI, siehe [AWS Serverless Application Repository Beispiele für Anwendungsrichtlinien](#) aus.

Aufheben der Freigabe einer Anwendung

Es gibt zwei Möglichkeiten, die Freigabe einer Anwendung von einer AWS-Organisation:

1. Der Herausgeber der Anwendung kann Berechtigungen mit dem [put-application-policy](#)-Befehl entfernen.
2. Ein Benutzer aus der Verwaltungskonto von einer AWS-Organisation kann eine [Anwendung aufgehoben](#) Betrieb für alle Anwendungen, die für die Organisation freigegeben sind, selbst wenn die Anwendung von einem Benutzer aus einem anderen Konto veröffentlicht wurde.

Note

Wenn eine Anwendung von einem nicht geteilt wird AWS-Organisation mit der Operation zur Aufhebung einer Freigabe aufgehoben werden kann AWS-Organisation erneut.

Weitere Informationen zu AWS Organizations finden Sie im [AWS Organizations Benutzerhandbuch](#).

Entfernen von Berechtigungen durch den Herausgeber

Entfernen von Berechtigungen durch den Herausgeber mithilfe der Konsole

So heben Sie die Freigabe einer Anwendung über die AWS Management Console entfernen, entfernen Sie die Richtlinianweisung, die sie für andere teilt AWS-Konten. Führen Sie dazu die folgenden Schritte aus:

1. Öffnen Sie die [AWS Serverless Application Repository-Konsole](#).
2. Wählen Sie im linken Navigationsbereich die Option Available Applications (Verfügbare Anwendungen).

3. Wählen Sie die Anwendung aus, deren Freigabe Sie aufheben möchten.
4. Wählen Sie die Registerkarte Sharing (Freigabe).
5. Wählen Sie im Abschnitt Application policy statements (Anwendungsrichtlinienanweisungen) die Richtlinienanweisung aus, die die Anwendung für die Konten freigibt, für die Sie die Freigabe aufheben möchten.
6. Wählen Sie Delete (Löschen).
7. Es wird eine Bestätigungsmeldung angezeigt. Klicken Sie erneut auf Delete (Löschen).

Aufheben von Berechtigungen durch den Herausgeber über die AWS CLI

So heben Sie die Freigabe einer Anwendung über die AWS CLI kann der Herausgeber Berechtigungen mit dem [put-application-policy](#) Befehl, um die Anwendung privat zu machen oder mit einem anderen Satz von AWS Konten.

Weitere Informationen zum Ändern von Berechtigungen mithilfe der AWS CLI, siehe [AWS Serverless Application Repository Beispiele für Anwendungsrichtlinien](#) aus.

Verwaltungskonto die Freigabe einer Anwendung aufheben

Verwaltungskonto trennt die Freigabe einer Anwendung von einem AWS Organisation über die - Konsole

So heben Sie die Freigabe einer Anwendung von einem AWS Organisation durch das AWS Management Console, ein Benutzer aus der Verwaltungskonto kann Folgendes tun:

1. Öffnen Sie die [AWS Serverless Application Repository-Konsole](#).
2. Wählen Sie im linken Navigationsbereich die Option Available Applications (Verfügbare Anwendungen).
3. Wählen Sie in der Kachel der Anwendung die Option Unshare (Freigabe aufheben).
4. Bestätigen Sie im Meldungsfeld zum Aufheben der Freigabe, dass Sie die Freigabe der Anwendung aufheben möchten, indem Sie die Organisations-ID und den Anwendungsnamen eingeben und dann Save (Speichern) auswählen.

Verwaltungskonto trennt die Freigabe einer Anwendung von einem AWS Organisation durch das AWS CLI

So heben Sie die Freigabe einer Anwendung von einem AWS Organisation, ein Benutzer aus dem Verwaltungskonto kann die `aws serverlessrepo unshare-application` Befehl.

Der folgende Befehl trennt die Freigabe einer Anwendung von einem AWS Organisation, wo *Anwendungs-ID* ist der Amazon-Ressourcenname (ARN) der Anwendung und *Organisations-ID* ist der AWS Organisations-ID:

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

Löschen einer Anwendung

Sie können Anwendungen aus dem löschen, AWS Serverless Application Repository, indem Sie entweder die AWS Management Console oder die AWS SAM-CLI verwenden.

Löschen einer Anwendung (Konsole)

Um eine Anwendung über die AWS Management Console zu veröffentlichen, gehen Sie wie folgt vor.

1. Öffnen Sie die [AWS Serverless Application Repository-Konsole](#).
2. Wählen Sie für My Applications (Eigene Anwendungen) die Anwendung aus, die Sie löschen möchten.
3. Wählen Sie auf der Detailseite der Anwendung Delete application (Anwendung löschen) aus.
4. Wählen Sie Delete application (Anwendung löschen) aus, um den Löschvorgang abzuschließen.

Löschen einer Anwendung (AWS CLI)

Um eine veröffentlichte Anwendung mithilfe der AWS CLI zu löschen, führen Sie den Befehl [aws serverlessrepo delete-application](#) aus.

Der folgende Befehl löscht eine Anwendung, wobei *application-id* der Amazon-Ressourcenname (ARN) der Anwendung ist:

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

Veröffentlichen der neuen Version einer vorhandenen Anwendung

In diesem Abschnitt erfahren Sie, wie Sie eine neue Version einer vorhandenen Anwendung mithilfe der AWS SAM-CLI oder der AWS Management Console im AWS Serverless Application Repository veröffentlichen. Anweisungen zum Veröffentlichen einer neuen Anwendung finden Sie unter [So veröffentlichen Sie Anwendungen](#).

Veröffentlichen der neuen Version einer vorhandenen Anwendung (AWS CLI)

Die einfachste Möglichkeit, eine neue Version einer vorhandenen Anwendung zu veröffentlichen, besteht darin, eine Reihe von AWS SAM-CLI-Befehlen zu verwenden. Weitere Informationen finden Sie unter [Veröffentlichen einer Anwendung mithilfe der AWS SAM-CLI](#) im AWS Serverless Application Model (AWS SAM)-Entwicklerhandbuch.

Veröffentlichen der neuen Version einer vorhandenen Anwendung (Konsole)

Gehen Sie folgendermaßen vor, um eine neue Version einer Anwendung zu veröffentlichen, die Sie zuvor veröffentlicht haben:

1. Öffnen Sie die [AWS Serverless Application Repository-Konsole](#).
2. Wählen Sie im Navigationsbereich My Applications (Meine Anwendungen) aus, um eine Liste der von Ihnen erstellten Anwendungen anzuzeigen.
3. Wählen Sie die Anwendung aus, für die Sie eine neue Version veröffentlichen möchten.
4. Wählen Sie Publish new version (Neue Version veröffentlichen) aus.
5. Geben Sie unter Versions (Versionen) die folgenden Anwendungsinformationen ein:

Property (Eigenschaft)	Erforderlich	Beschreibung
Semantic version (Semantische Version)	TRUE	Die semantische Version der Anwendung. Weitere Informationen finden Sie auf der Semantic Versioning-Website . Sie müssen einen Wert für diese Eigenschaft angeben, um Ihre Anwendung öffentlich zu machen.

Property (Eigenschaft)	Erforderlich	Beschreibung
Quellcode-URL	FALSE	Ein Link zu einem öffentlichen Repository für den Quellcode Ihrer Anwendung.
SAM-Vorlage	TRUE	Eine gültige AWS Serverless Application Model (AWS SAM) -Vorlage, die die AWS Ressourcen, die verwendet werden.

6. Wählen Sie Publish version (Version veröffentlichen) aus.

Verifiziertes Autoren-Badge

Verifizierte Autoren im AWS Serverless Application Repository sind diejenigen, für die AWS hat als vernünftiger und umsichtiger Dienstleister die vom Anfragenden bereitgestellten Informationen in gutem Glauben überprüft und bestätigt, dass die Identität des Anforderers wie behauptet ist.

Anwendungen verifizierter Autoren zeigen ein verifiziertes Autoren-Badge sowie einen Link zum öffentlichen Profil des Autors an. Das verifizierte Autoren-Badge wird sowohl in den Suchergebnissen als auch auf der Anwendungsdetailseite angezeigt.

Anfordern eines verifizierten Autoren-Badge

Sie können beantragen, als verifizierter Autor im AWS Serverless Application Repository indem Sie eine E-Mail an [senden `serverlessrepo-verified-author@amazon.com`](mailto:serverlessrepo-verified-author@amazon.com) aus. Sie müssen die folgenden Informationen angeben:

- Name des Autors
- AWS-Konto-ID
- Öffentlich zugänglicher Profillink, z. B. GitHub oder LinkedIn Profil

Nachdem Sie eine Anfrage für ein verifiziertes Autoren-Badge eingereicht haben, können Sie eine Antwort von AWS innerhalb weniger Tage. Möglicherweise werden Sie nach zusätzlichen Informationen gefragt, bevor Ihre Anfrage genehmigt wird.

Nachdem Ihre Anfrage genehmigt wurde, können Sie erwarten, dass das verifizierte Autoren-Badge für Ihre Anwendungen innerhalb eines Tages angezeigt wird.

Note

Das verifizierte Autoren-Badge wird für alle Anwendungen angezeigt, die sowohl mit AWS Konto- und Autorennamen. Da es sich bei AWS Konten können mehrere Autoren haben, Badges werden für Anwendungen mit einem anderen Autorennamen nicht angezeigt. Damit Autoren-Badges in Anwendungen mit unterschiedlichen Autorennamen angezeigt werden, müssen Sie eine andere Anfrage für diesen Autor senden.

Freigeben von Lambda-Ebenen

Wenn Sie Funktionen in einer Lambda-Ebene implementiert haben, möchten Sie Ihre Ebene möglicherweise freigeben, ohne eine globale Instance davon zu hosten. Wenn Sie Ebenen auf diese Weise freigeben, können andere Benutzer eine Instance Ihrer Ebene in ihrem eigenen Konto bereitstellen. Dadurch wird verhindert, dass Clientanwendungen von einer globalen Instance Ihrer Ebene abhängig sind. Die AWS Serverless Application Repository ermöglicht es Ihnen, Lambda-Ebenen auf diese Weise einfach freizugeben.

Weitere Informationen über Lambda-Ebenen finden Sie unter [AWS Lambda Ebenen](#) im AWS Lambda Entwicklerhandbuch.

So funktioniert es

Im Folgenden finden Sie die Schritte zum Freigeben der Ebene mit dem AWS Serverless Application Repository. Dadurch kann eine Kopie Ihrer Ebene im AWS Konto.

1. Definieren einer serverlosen Anwendung mit einer AWS SAM Vorlage, die Ihren Layer als Ressource enthält - das heißt entweder eine [AWS::Serverless::LayerVersion](#) oder ein [AWS::Lambda::LayerVersion](#) Ressource.
2. Veröffentlichen Sie Ihre Anwendung im AWS Serverless Application Repository und geben Sie sie frei (öffentlich oder privat).
3. Ein Kunde stellt Ihre Anwendung bereit, die eine Kopie Ihrer Ebene in seiner eigenen erstellt AWS Konto. Der Kunde kann nun auf den Amazon-Ressourcennamen (ARN) der Ebene in seinem AWS-Konto in ihrer Client-Anwendung.

Beispiel

Im Folgenden wird ein Beispiel gezeigtAWS SAM-Vorlage für eine Anwendung, die die Lambda-Ebene enthält, die Sie freigeben möchten:

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

Wenn ein Kunde Ihre Anwendung aus dem bereitstelltAWS Serverless Application Repositorywird ein Layer in ihrerAWSKonto. Der ARN der Ebene sieht ungefähr wie folgt aus:

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

Der Kunde kann diesen ARN nun in seiner eigenen Client-Anwendung referenzieren, wie in diesem Beispiel:

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


Bereitstellen von Anwendungen

Der folgende Abschnitt hilft Ihnen dabei, mehr über die Suche und Bereitstellung von serverlosen Anwendungen zu erfahren, die im AWS Serverless Application Repository veröffentlicht wurden. Sie können nach öffentlich verfügbaren Anwendungen suchen, ohne ein AWS-Konto durch Besuch der [Publicly site](#) aus. Sie können auch in der AWS Lambda-Konsole nach Anwendungen suchen.

Einige Anwendungen verfügen über ein verifiziertes Autoren-Badge mit einem Link zum Profil des Autors. Ein Autor gilt als verifizierter Autor wenn AWS hat als vernünftiger und umsichtiger Dienstleister die vom Anfragenden bereitgestellten Informationen in gutem Glauben überprüft und bestätigt, dass die Identität des Anforderers wie behauptet ist.

Bevor Sie Anwendungen aus dem AWS Serverless Application Repository bereitstellen, informieren Sie sich in den folgenden Themen über Anwendungsbereitstellungsberechtigungen und Anwendungsfunktionen.

Themen

- [Anwendungsbereitstellungsberechtigungen](#)
- [Anwendungsfunktionen: IAM-Rollen, Ressourcenrichtlinien und verschachtelte Anwendungen](#)
- [So stellen Sie Anwendungen bereit](#)

Anwendungsbereitstellungsberechtigungen

Um eine Anwendung im AWS Serverless Application Repository bereitzustellen, müssen Sie über die Berechtigung dafür verfügen. Es gibt drei Kategorien von Anwendungen, für die Sie über Berechtigungen zum Bereitstellen verfügen:

- **Privat**— Anwendungen, die mit demselben Konto erstellt wurden und für kein anderes Konto freigegeben wurden. Sie haben die Berechtigung, Anwendungen bereitzustellen, die mit Ihrem AWS-Konto.
- **Privat freigegeben**— Anwendungen, die der Herausgeber explizit für einen bestimmten Satz freigegeben hat AWS-Konten. Sie haben die Berechtigung, Anwendungen bereitzustellen, die für Ihr freigegeben wurden AWS-Konto.
- **Öffentlich freigegeben**— Anwendungen, die der Herausgeber für alle freigegeben hat. Sie haben die Berechtigung, alle öffentlich freigegebenen Anwendungen bereitzustellen.

Sie können nur nach Anwendungen suchen, für die Sie über Berechtigungen verfügen. Dazu gehören Anwendungen, die mit Ihrem erstellt wurdenAWSKonto, privat mit IhremAWSKonto und öffentlich geteilt. Alle anderen Anwendungen werden nicht für Sie angezeigt.

Important

Anwendungen, die verschachtelte Anwendungen enthalten, erben die Freigabeeinschränkungen der verschachtelten Anwendungen. Angenommen, eine Anwendung ist öffentlich freigegeben, enthält aber eine verschachtelte Anwendung, die nur privat für denAWSKonto, das die übergeordnete Anwendung erstellt hat. In diesem Fall, wennAWSDas Konto hat keine Berechtigung, die verschachtelte Anwendung bereitzustellen, dann können Sie die übergeordnete Anwendung nicht bereitstellen. Weitere Informationen über verschachtelte Anwendungen finden Sie im Thema [Verschachtelte Anwendungen](#) im AWS Serverless Application Model-Entwicklerhandbuch.

Anwendungsfunktionen: IAM-Rollen, Ressourcenrichtlinien und verschachtelte Anwendungen

Bevor Sie eine Anwendung bereitstellen können,AWS Serverless Application Repository überprüft der die Anwendungsvorlage auf IAM-Rollen,AWS Ressourcenrichtlinien und verschachtelte Anwendungen, die laut Vorlage erstellt werden sollen. IAM-Ressourcen, wie z. B. eine IAM-Rolle mit Vollzugriff, können jede Ressource in IhremAWS -Konto ändern. Daher wird empfohlen, die mit der Anwendung verknüpften Berechtigungen zu prüfen, bevor Sie fortfahren, damit Sie nicht versehentlich Ressourcen mit eskalierten Berechtigungen erstellen. Um sicherzustellen, dass Sie dies getan haben, müssen Sie bestätigen, dass die Anwendung Funktionen enthält, ehe AWS Serverless Application Repository die Anwendung in seinem Namen bereitstellen kann.

Anwendungen können folgende vier Funktionen enthalten: `CAPABILITY_IAM`, `CAPABILITY_NAMED_IAM`, `CAPABILITY_RESOURCE_POLICY` und `CAPABILITY_AUTO_EXPAND`.

Für die folgenden Ressourcen müssen Sie`CAPABILITY_IAM` oder angeben`CAPABILITY_NAMED_IAM`: [AWS::IAM::Group](#),[AWS::IAM::InstanceProfile](#), [AWS::IAM::Policy](#), und [AWS::IAM::Role](#). Beinhaltet die Anwendung IAM-Ressourcen mit benutzerdefinierten Namen, müssen Sie `CAPABILITY_NAMED_IAM` angeben. Ein Beispiel dazu, wie Sie bestimmte Funktionen angeben, finden Sie unter [Suchen und Bestätigen von Anwendungsfunktionen \(AWS CLI\)](#).

Für die folgenden Ressourcen müssen Sie angeben `CAPABILITY_RESOURCE_POLICY`: [AWS::Lambda::LayerVersionPermission](#), [AWS::Lambda::Permission](#), [AWS::Events::EventBusPolicy](#), [AWS::IAM::Policy](#), [AWS::ApplicationAutoScaling::ScalingPolicy](#), [AWS::S3::BucketPolicy](#), [AWS::SQS::QueuePolicy](#), und [AWS::SNS::TopicPolicy](#).

Bei Anwendungen, die eine oder mehrere verschachtelte Anwendungen enthalten, müssen Sie `CAPABILITY_AUTO_EXPAND` angeben. Weitere Informationen über verschachtelte Anwendungen finden Sie im Thema [Verschachtelte Anwendungen](#) im AWS Serverless Application Model-Entwicklerhandbuch.

Suchen und Bestätigen von Anwendungsfunktionen (Konsole)

Sie finden Anwendungen, die AWS Serverless Application Repository auf der [AWS Serverless Application Repository Website](#) oder über die [Lambda-Konsole \(auf der Seite Create Function unter dem AWS Serverless Application Repository Tab\)](#) verfügbar sind.

Anwendungen, die eine Bestätigung von Funktionen für das Erstellen von benutzerdefinierten IAM-Rollen oder Ressourcenrichtlinien benötigen, werden standardmäßig nicht in den Suchergebnissen angezeigt. Zum Suchen nach Anwendungen, die diese Funktionen enthalten, müssen Sie das Kontrollkästchen `Show apps that create custom IAM roles or resource policies` (Apps zeigen, die benutzerdefinierte IAM-Rollen oder Ressourcenrichtlinien erstellen) aktivieren.

Sie können den Funktionsumfang einer Anwendung auf der Registerkarte `Permissions` (Berechtigungen) überprüfen, wenn Sie eine Anwendung auswählen. Zum Bereitstellen der Anwendung müssen Sie das Kontrollkästchen `I acknowledge this application creates custom IAM roles or resource policies` (Ich bestätige, dass diese Anwendung benutzerdefinierte IAM-Rollen und Ressourcenrichtlinien erstellt) aktivieren. Wenn Sie diese Funktionen nicht bestätigen, wird die folgende Fehlermeldung angezeigt: `Bestätigung erforderlich. Um die Bereitstellung durchzuführen, aktivieren Sie das Kontrollkästchen im Abschnitt Anwendungsparameter konfigurieren.`

Anzeigen von Anwendungsfunktionen (AWS CLI)

Um die Funktionen einer Anwendung mithilfe der AWS CLI anzuzeigen, benötigen Sie zunächst den Amazon-Ressourcennamen (ARN) der Anwendung. Sie können dann den folgenden Befehl ausführen:

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

Die Eigenschaft [requiredCapabilities](#) enthält die Liste der Anwendungsfunktionen, die Sie vor der Bereitstellung der Anwendung bestätigen müssen. Beachten Sie, dass die Anwendung keine erforderlichen Funktionen besitzt, wenn die Eigenschaft [requiredCapabilities](#) leer ist.

So stellen Sie Anwendungen bereit

In diesem Abschnitt finden Sie Verfahren zum Bereitstellen von serverlosen Anwendungen aus dem AWS Serverless Application Repository mithilfe der AWS Management Console oder der AWS CLI.

Bereitstellen einer neuen Anwendungsversion (Konsole)

In diesem Abschnitt erfahren Sie, wie Sie eine neue Anwendung vom AWS Serverless Application Repository mithilfe der AWS Management Console bereitstellen. Anweisungen zum Bereitstellen einer neuen Version einer vorhandenen Anwendung finden Sie unter [Aktualisieren von Anwendungen](#).

Suchen und Entwickeln von Anwendungen

Mithilfe des folgenden Vorgangs können Sie eine Anwendung im AWS Serverless Application Repository suchen, konfigurieren und bereitstellen.

So suchen und konfigurieren Sie eine Anwendung im AWS Serverless Application Repository

1. Öffnen Sie die [öffentliche AWS Serverless Application Repository-Startseite](#) oder öffnen Sie die [AWS Lambda-Konsole](#). Wählen Sie Create function (Funktion erstellen) aus und dann Browse serverless app repository (Serverloses App-Repository durchsuchen).
2. Suchen Sie nach einer Anwendung.

Note

Zum Anzeigen von Anwendungen, die benutzerdefinierte IAM-Rollen oder Ressourcenrichtlinien enthalten, aktivieren Sie das Kontrollkästchen Show apps that create custom IAM roles or resource policies (Apps zeigen, die benutzerdefinierte IAM-Rollen oder Ressourcenrichtlinien erstellen). Weitere Informationen zu benutzerdefinierten IAM-Rollen und Ressourcenrichtlinien finden Sie unter [Bestätigen von Anwendungsfunktionen](#).

3. Wählen Sie eine Anwendung aus, um Details wie ihre Berechtigungen, Funktionen und die Häufigkeit, mit der sie von AWS Kunden bereitgestellt wurde, einzusehen.

Die Anzahl der Bereitstellungen wird für die AWS Region angezeigt, in der Sie versuchen, die Anwendung bereitzustellen.

4. Auf der Detailseite der Anwendung können Sie die Berechtigungen der Anwendung und Anwendungsressourcen durch Ansicht der AWS SAM-Vorlage, Lizenz oder Readme-Datei anzeigen. Auf dieser Seite können Sie bei öffentlich freigegebenen Anwendungen auch nach dem Link der Source code URL (Quellcode-URL) suchen. Wenn die Anwendung verschachtelte Anwendungen enthält, können Sie auch die Details der verschachtelten Anwendungen auf dieser Seite sehen.
5. Konfigurieren Sie die Anwendung im Abschnitt Application Settings (Anwendungseinstellungen). Eine Anleitung zur Konfiguration einer bestimmten Anwendung finden Sie in der Readme-Datei der Anwendung.

Konfigurationsvoraussetzungen können unter anderem die Angabe des Namens einer Ressource sein, wenn die Anwendung Zugriff darauf erhalten soll. Eine solche Ressource kann eine Amazon DynamoDB-Tabelle, ein Amazon S3 S3-Bucket oder eine Amazon API Gateway Gateway-API sein.

6. Wählen Sie Deploy (Bereitstellen) aus. Damit gelangen Sie auf die Seite Deployment status (Bereitstellungsstatus).

Note

Wenn die Anwendung über Funktionen verfügt, die eine Bestätigung erfordern, müssen Sie das Kontrollkästchen `I acknowledge this application creates custom IAM roles or resource policies` (Ich bestätige, dass diese Anwendung benutzerdefinierte IAM-Rollen oder Ressourcenrichtlinien erstellt) aktivieren, bevor Sie die Anwendung bereitstellen. Andernfalls wird ein Fehler ausgegeben. Weitere Informationen zu benutzerdefinierten IAM-Rollen und Ressourcenrichtlinien finden Sie unter [Bestätigen von Anwendungsfunktionen](#).

7. Auf der Seite Deployment status (Bereitstellungsstatus) können Sie den Fortschritt Ihrer Bereitstellung verfolgen. Während Sie auf den Abschluss Ihrer Bereitstellung warten, können Sie nach anderen Anwendungen suchen und über die Lambda-Konsole zu dieser Seite zurückkehren.

Nachdem Ihre Anwendung erfolgreich bereitgestellt wurde, können Sie die Ressourcen überprüfen und verwalten, die mithilfe vorhandener AWS Tools erstellt wurden.

Bereitstellen einer neuen Anwendungsversion (AWS CLI)

In diesem Abschnitt erfahren Sie, wie Sie eine neue Anwendung vom AWS Serverless Application Repository mithilfe einer AWS CLI bereitstellen. Anweisungen zum Bereitstellen einer neuen Version einer vorhandenen Anwendung finden Sie unter [Aktualisieren von Anwendungen](#).

Suchen und Bestätigen von Anwendungsfunktionen (AWS CLI)

Zum Bestätigen der Anwendungsfunktionen mit der AWS CLI führen Sie die folgenden Schritte aus:

1. Überprüfen Sie die Funktionen der Anwendung. Verwenden Sie den folgenden AWS CLI Befehl, um die Funktionen einer Anwendung zu überprüfen:

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

Die Antworteigenschaft [requiredCapabilities](#) enthält die Liste der Anwendungsfunktionen, die Sie vor der Bereitstellung der Anwendung bestätigen müssen. Sie können auch die [GetApplication API](#) in den AWS SDKs verwenden, um diese Daten abzurufen.

2. Erstellen Sie das Changeset. Sie müssen die erforderlichen [Funktionen](#) bereitstellen, wenn Sie das AWS CloudFormation Changeset erstellen. Verwenden Sie z. B. den folgenden AWS CLI-Befehl, um eine Anwendung durch Bestätigung ihrer Funktionen bereitzustellen:

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

Die Änderungssatz-ID wird zurückgegeben, wenn dieser Befehl erfolgreich ausgeführt wird. Sie benötigen die Änderungssatz-ID für den nächsten Schritt. Sie können auch die [CreateCloudFormationChangeSet API](#) in den AWS SDKs verwenden, um das Changeset zu erstellen.

Der folgende AWS CLI Befehl bestätigt beispielsweise eine Anwendung, die eine [AWS::IAM::Role](#) Ressource mit einem benutzerdefinierten Namen und eine oder mehrere verschachtelte Anwendungen enthält:

```
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. Führen Sie das Changeset aus. Durch das Ausführen des Änderungssatzes wird die Bereitstellung tatsächlich ausgeführt. Geben Sie die Änderungssatz-ID an, die beim Erstellen des Änderungssatzes im vorherigen Schritt zurückgegeben wurde.

Der folgende AWS CLI-Beispielbefehl führt den Anwendungsänderungssatz aus, um die Anwendung bereitzustellen:

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

Sie können auch die [ExecuteChangeSet API](#) in den AWS SDKs verwenden, um das Changeset auszuführen.

Löschen von Anwendungs-Stacks

Um eine Anwendung zu löschen, die Sie zuvor mit AWS Serverless Application Repository bereitgestellt haben, führen Sie die gleichen Schritte wie für das Löschen eines AWS CloudFormation-Stacks aus:

- AWS Management Console: Informationen zum Löschen einer Anwendung mithilfe [von finden Sie im AWS CloudFormation Benutzerhandbuch unter Löschen eines Stacks auf der AWS CloudFormation Konsole](#). AWS Management Console
- AWS CLI: Informationen zum Löschen einer Anwendung mithilfe [von finden Sie im AWS CloudFormation Benutzerhandbuch unter Löschen eines Stacks](#). AWS CLI

Aktualisieren von Anwendungen

Nachdem Sie eine Anwendung aus dem AWS Serverless Application Repository bereitgestellt haben, möchten Sie diese möglicherweise aktualisieren. Sie möchten beispielsweise eine Anwendungseinstellung ändern oder die Anwendung auf die neueste Version aktualisieren, die veröffentlicht wurde.

In den folgenden Abschnitten wird beschrieben, wie eine neue Version einer Anwendung mithilfe der AWS Management Console oder der AWS CLI bereitgestellt wird.

Aktualisieren von Anwendungen (Konsole)

Verwenden Sie zum Aktualisieren einer zuvor bereitgestellten Anwendung dasselbe Verfahren wie beim Bereitstellen einer neuen Anwendung und geben Sie denselben Anwendungsnamen an, mit dem Sie sie ursprünglich bereitgestellt haben. Insbesondere stellt das AWS Serverless Application Repository Ihrem Anwendungsnamen `serverlessrepo-` voran. Um jedoch eine neue Version Ihrer Anwendung bereitzustellen, geben Sie den ursprünglichen Anwendungsnamen an, ohne `serverlessrepo-` voranzustellen.

Wenn Sie beispielsweise eine Anwendung mit dem Namen `MyApplication` bereitgestellt haben, wäre der Stack-Name `serverlessrepo-MyApplication`. Um diese Anwendung zu aktualisieren, müssen Sie den Namen `MyApplication` erneut angeben — geben Sie nicht den vollständigen Stacknamen von `anserverlessrepo-MyApplication`.

Für alle anderen Anwendungseinstellungen können Sie die Werte der vorherigen Bereitstellung übernehmen oder neue Werte bereitstellen.

Aktualisieren von Anwendungen (AWS CLI)

Verwenden Sie zum Aktualisieren einer zuvor bereitgestellten Anwendung dasselbe Verfahren wie beim Bereitstellen einer neuen Anwendung und geben Sie denselben `--stack-name` an, mit dem Sie sie ursprünglich bereitgestellt haben. Insbesondere stellt AWS Serverless Application Repository Ihrem Stack-Namen `serverlessrepo-` voran. Um jedoch eine neue Version Ihrer Anwendung bereitzustellen, geben Sie den ursprünglichen Stack-Namen an, ohne `serverlessrepo-` voranzustellen.

Wenn Sie beispielsweise eine Anwendung mit dem Stack-Namen `MyApplication` bereitgestellt haben, lautet der erstellte Stack-Name `serverlessrepo-MyApplication`. Um diese Anwendung zu aktualisieren, müssen Sie den Namen `MyApplication` erneut angeben — geben Sie nicht den vollständigen Stacknamen von `anserverlessrepo-MyApplication`.

Sicherheit im AWS Serverless Application Repository

Die Sicherheit in der Cloud hat AWS höchste Priorität. Als AWS-Kunde profitieren Sie von einer Rechenzentrums- und Netzwerkarchitektur, die zur Erfüllung der Anforderungen von Organisationen entwickelt wurden, für die Sicherheit eine kritische Bedeutung hat.

Sicherheit gilt zwischen AWS und Ihnen eine geteilte Verantwortung. Das [Modell der geteilten Verantwortung](#) beschreibt dies als Sicherheit der Cloud und Sicherheit in der Cloud:

- Sicherheit der Cloud – AWS ist dafür verantwortlich, die Infrastruktur zu schützen, mit der AWS-Services in der AWS-Cloud ausgeführt werden. AWS stellt Ihnen außerdem Services bereit, die Sie sicher nutzen können. Auditoren von Drittanbietern testen und überprüfen die Effektivität unserer Sicherheitsmaßnahmen im Rahmen der [AWS-Compliance-Programme](#) regelmäßig. Informationen zu den Compliance-Programmen, die für das AWS Serverless Application Repository gelten, finden Sie unter [Vom Compliance-Programm abgedeckte AWS-Services](#).
- Sicherheit in der Cloud – Ihr Verantwortungsumfang wird durch den AWS-Service bestimmt, den Sie verwenden. Sie sind auch für andere Faktoren verantwortlich, etwa für die Vertraulichkeit Ihrer Daten, für die Anforderungen Ihres Unternehmens und für die geltenden Gesetze und Vorschriften.

In dieser Dokumentation wird erläutert, wie das Modell der übergreifenden Verantwortlichkeit bei der Verwendung von AWS Serverless Application Repository zum Tragen kommt. Die folgenden Themen zeigen Ihnen, wie Sie das AWS Serverless Application Repository zur Erfüllung Ihrer Sicherheits- und Compliance-Ziele konfigurieren können. Sie erfahren außerdem, wie Sie andere AWS-Services verwenden, um Ihre AWS Serverless Application Repository-Ressourcen zu überwachen und zu schützen.

Themen

- [Datenschutz im AWS Serverless Application Repository](#)
- [Identity and Access Management für das AWS Serverless Application Repository](#)
- [Protokollierung und Überwachung im AWS Serverless Application Repository](#)
- [Compliance-Validierung für das AWS Serverless Application Repository](#)
- [Ausfallsicherheit im AWS Serverless Application Repository](#)
- [Infrastruktursicherheit im AWS Serverless Application Repository](#)

Datenschutz im AWS Serverless Application Repository

Das [Modell der geteilten Verantwortung](#) von AWS gilt für den Datenschutz in AWS Serverless Application Repository. Wie in diesem Modell beschrieben, ist AWS verantwortlich für den Schutz der globalen Infrastruktur, in der die gesamte AWS Cloud ausgeführt wird. Sie sind dafür verantwortlich, die Kontrolle über Ihre in dieser Infrastruktur gehosteten Inhalte zu behalten. Sie sind auch für die Sicherheitskonfiguration und die Verwaltungsaufgaben für die von Ihnen verwendeten AWS-Services verantwortlich. Weitere Informationen zum Datenschutz finden Sie unter [Häufig gestellte Fragen zum Datenschutz](#). Informationen zum Datenschutz in Europa finden Sie im Blog-Beitrag [AWS-Modell der geteilten Verantwortung und in der DSGVO](#) im AWS-Sicherheitsblog.

Aus Datenschutzgründen empfehlen wir, AWS-Konto-Anmeldeinformationen zu schützen und einzelne Benutzer mit AWS IAM Identity Center oder AWS Identity and Access Management (IAM) einzurichten. So erhält jeder Benutzer nur die Berechtigungen, die zum Durchführen seiner Aufgaben erforderlich sind. Außerdem empfehlen wir, die Daten mit folgenden Methoden zu schützen:

- Verwenden Sie für jedes Konto die Multi-Faktor Authentifizierung (MFA).
- Verwenden Sie SSL/TLS für die Kommunikation mit AWS-Ressourcen. Wir benötigen TLS 1.2 und empfehlen TLS 1.3.
- Richten Sie die API und die Protokollierung von Benutzeraktivitäten mit AWS CloudTrail ein.
- Verwenden Sie AWS-Verschlüsselungslösungen zusammen mit allen Standardsicherheitskontrollen in AWS-Services.
- Verwenden Sie erweiterte verwaltete Sicherheitsservices wie Amazon Macie, die dabei helfen, in Amazon S3 gespeicherte persönliche Daten zu erkennen und zu schützen.
- Wenn Sie für den Zugriff auf AWS über eine Befehlszeilenschnittstelle oder über eine API FIPS 140-2-validierte kryptografische Module benötigen, verwenden Sie einen FIPS-Endpunkt. Weitere Informationen über verfügbare FIPS-Endpunkte finden Sie unter [Federal Information Processing Standard \(FIPS\) 140-2](#).

Wir empfehlen dringend, in Freitextfeldern, z. B. im Feld Name, keine vertraulichen oder sensiblen Informationen wie die E-Mail-Adressen Ihrer Kunden einzugeben. Dies gilt auch, wenn Sie unter Verwendung der Konsole, der API, AWS CLI oder AWS SDKs mit AWS Serverless Application Repository oder anderen AWS-Services arbeiten. Alle Daten, die Sie in Tags oder Freitextfelder eingeben, die für Namen verwendet werden, können für Abrechnungs- oder Diagnoseprotokolle verwendet werden. Wenn Sie eine URL für einen externen Server bereitstellen, empfehlen wir

dringend, keine Anmeldeinformationen zur Validierung Ihrer Anforderung an den betreffenden Server in die URL einzuschließen.

Verschlüsselung während der Übertragung

AWS Serverless Application Repository-API-Endpunkte unterstützen ausschließlich sichere Verbindungen über HTTPS. Wenn Sie AWS Serverless Application Repository-Ressourcen mit der AWS Management Console, dem AWS SDK oder der AWS Serverless Application Repository-API verwalten, wird die gesamte Kommunikation mit Transport Layer Security (TLS) verschlüsselt.

Eine vollständige Liste der API-Endpunkte finden Sie unter [AWSRegionen und Endpunkte](#) in der. Allgemeine AWS-Referenz

Verschlüsselung im Ruhezustand

Das AWS Serverless Application Repository verschlüsselt Dateien, die Sie zu AWS Serverless Application Repository hochladen, einschließlich Bereitstellungspaketen und Ebenenarchiven.

Identity and Access Management für das AWS Serverless Application Repository

AWS Identity and Access Management (IAM) ist ein AWS-Service, mit dem ein Administrator den Zugriff auf AWS-Ressourcen sicher steuern kann. IAM-Administratoren steuern, wer authentifiziert (angemeldet) und autorisiert (Berechtigungen besitzt) ist, um AWS Serverless Application Repository Ressourcen zu nutzen. IAM ist ein AWS-Service, den Sie ohne zusätzliche Kosten verwenden können.

Einen Überblick über die Funktionsweise von IAM finden Sie unter [Understanding How IAM Works](#) im IAM-Benutzerhandbuch.

Themen

- [Zielgruppe](#)
- [Authentifizieren mit Identitäten](#)
- [Verwalten des Zugriffs mit Richtlinien](#)
- [Wie das mit IAMAWS Serverless Application Repository funktioniert](#)
- [Beispiele für identitätsbasierte AWS Serverless Application Repository-Richtlinien](#)

- [AWS Serverless Application Repository Beispiele für Anwendungsrichtlinien](#)
- [AWS Serverless Application Repository API-Berechtigungen: Referenz für Aktionen und Ressourcen](#)
- [Fehlerbehebung bei AWS Serverless Application Repository-Identität und -Zugriff](#)

Zielgruppe

Wie Sie AWS Identity and Access Management (IAM) verwenden, unterscheidet sich je nach Ihrer Arbeit in AWS Serverless Application Repository.

Service user (Service-Benutzer) – Wenn Sie den AWS Serverless Application Repository-Service zur Ausführung von Aufgaben verwenden, stellt Ihnen Ihr Administrator die Anmeldeinformationen und Berechtigungen bereit, die Sie benötigen. Wenn Sie für Ihre Arbeit weitere AWS Serverless Application Repository-Funktionen ausführen, benötigen Sie möglicherweise zusätzliche Berechtigungen. Wenn Sie die Funktionsweise der Zugriffskontrolle nachvollziehen, wissen Sie bereits, welche Berechtigungen Sie von Ihrem Administrator anzufordern müssen. Unter [Fehlerbehebung bei AWS Serverless Application Repository-Identität und -Zugriff](#) finden Sie nützliche Informationen für den Fall, dass Sie keinen Zugriff auf eine Funktion in AWS Serverless Application Repository haben.

Service administrator (Service-Administrator) – Wenn Sie in Ihrem Unternehmen für AWS Serverless Application Repository-Ressourcen verantwortlich sind, haben Sie wahrscheinlich vollständigen Zugriff auf AWS Serverless Application Repository. Es ist Ihre Aufgabe, zu bestimmen, auf welche AWS Serverless Application Repository-Funktionen und Ressourcen Ihre Service-Benutzer zugreifen sollen. Sie müssen dann Anträge an Ihren IAM-Administrator stellen, um die Berechtigungen Ihrer Servicenutzer zu ändern. Lesen Sie die Informationen auf dieser Seite, um die Grundkonzepte von IAM nachzuvollziehen. Weitere Informationen dazu, wie Ihr Unternehmen IAM mit AWS Serverless Application Repository verwenden kann, finden Sie unter [Wie das mit IAM AWS Serverless Application Repository funktioniert](#).

IAM-Administrator – Wenn Sie als IAM-Administrator fungieren, sollten Sie Einzelheiten dazu kennen, wie Sie Richtlinien zur Verwaltung des Zugriffs auf AWS Serverless Application Repository verfassen können. Beispiele für identitätsbasierte AWS Serverless Application Repository-Richtlinien, die Sie in IAM verwenden können, finden Sie unter [Beispiele für identitätsbasierte AWS Serverless Application Repository-Richtlinien](#).

Authentifizieren mit Identitäten

Die Authentifizierung ist die Art und Weise, wie Sie sich mit Ihren Anmeldeinformationen bei AWS anmelden. Die Authentifizierung (Anmeldung bei AWS) muss als Root-Benutzer des AWS-Kontos, als IAM-Benutzer oder durch Übernahme einer IAM-Rolle erfolgen.

Sie können sich bei AWS als Verbundidentität mit Anmeldeinformationen anmelden, die über eine Identitätsquelle bereitgestellt werden. Benutzer von AWS IAM Identity Center. (IAM Identity Center), die Single-Sign-on-Authentifizierung Ihres Unternehmens und Anmeldeinformationen für Google oder Facebook sind Beispiele für Verbundidentitäten. Wenn Sie sich als Verbundidentität anmelden, hat der Administrator vorher mithilfe von IAM-Rollen einen Identitätsverbund eingerichtet. Wenn Sie auf AWS mithilfe des Verbunds zugreifen, übernehmen Sie indirekt eine Rolle.

Je nachdem, welcher Benutzertyp Sie sind, können Sie sich bei der AWS Management Console oder beim AWS-Zugriffportal anmelden. Weitere Informationen zum Anmelden bei AWS finden Sie unter [Anmelden bei Ihrem AWS-Konto](#) im Benutzerhandbuch von AWS-Anmeldung.

Bei programmgesteuerten Zugriff auf AWS bietet AWS ein Software Development Kit (SDK) und eine Command Line Interface (CLI, Befehlszeilenschnittstelle) zum kryptographischen Signieren Ihrer Anfragen mit Ihren Anmeldeinformationen. Wenn Sie keine AWS-Tools verwenden, müssen Sie Anforderungen selbst signieren. Weitere Informationen zur Verwendung der empfohlenen Methode zum eigenen Signieren von Anforderungen finden Sie unter [Signieren von AWS-API-Anforderungen](#) im IAM-Benutzerhandbuch.

Unabhängig von der verwendeten Authentifizierungsmethode müssen Sie möglicherweise zusätzliche Sicherheitsinformationen angeben. AWS empfiehlt beispielsweise die Verwendung von Multi-Faktor Authentifizierung (MFA), um die Sicherheit Ihres Kontos zu verbessern. Weitere Informationen finden Sie unter [Multi-Faktor-Authentifizierung](#) im AWS IAM Identity Center-Benutzerhandbuch und [Verwenden der Multi-Faktor-Authentifizierung \(MFA\) in AWS](#) im IAM-Benutzerhandbuch.

AWS-Konto-Stammbenutzer

Wenn Sie ein AWS-Konto neu erstellen, beginnen Sie mit einer Anmeldeidentität, die vollständigen Zugriff auf alle AWS-Services und Ressourcen des Kontos hat. Diese Identität wird als AWS-Konto-Root-Benutzer bezeichnet. Für den Zugriff auf den Root-Benutzer müssen Sie sich mit der E-Mail-Adresse und dem Passwort anmelden, die zur Erstellung des Kontos verwendet wurden. Wir raten ausdrücklich davon ab, den Root-Benutzer für Alltagsaufgaben zu verwenden. Schützen Sie Ihre Root-Benutzer-Anmeldeinformationen und verwenden Sie diese, um die Aufgaben

auszuführen, die nur der Root-Benutzer ausführen kann. Eine vollständige Liste der Aufgaben, für die Sie sich als Root-Benutzer anmelden müssen, finden Sie unter [Aufgaben, die Root-Benutzer-Anmeldeinformationen erfordern](#) im IAM-Benutzerhandbuch.

IAM-Benutzer und -Gruppen

Ein [IAM-Benutzer](#) ist eine Identität in Ihrem AWS-Konto mit bestimmten Berechtigungen für eine einzelne Person oder eine einzelne Anwendung. Wenn möglich, empfehlen wir, temporäre Anmeldeinformationen zu verwenden, anstatt IAM-Benutzer zu erstellen, die langfristige Anmeldeinformationen wie Passwörter und Zugriffsschlüssel haben. Bei speziellen Anwendungsfällen, die langfristige Anmeldeinformationen mit IAM-Benutzern erfordern, empfehlen wir jedoch, die Zugriffsschlüssel zu rotieren. Weitere Informationen finden Sie unter [Regelmäßiges Rotieren von Zugriffsschlüsseln für Anwendungsfälle, die langfristige Anmeldeinformationen erfordern](#) im IAM-Benutzerhandbuch.

Eine [IAM-Gruppe](#) ist eine Identität, die eine Sammlung von IAM-Benutzern angibt. Sie können sich nicht als Gruppe anmelden. Mithilfe von Gruppen können Sie Berechtigungen für mehrere Benutzer gleichzeitig angeben. Gruppen vereinfachen die Verwaltung von Berechtigungen, wenn es zahlreiche Benutzer gibt. Sie könnten beispielsweise einer Gruppe mit dem Namen IAMAdmins Berechtigungen zum Verwalten von IAM-Ressourcen erteilen.

Benutzer unterscheiden sich von Rollen. Ein Benutzer ist einer einzigen Person oder Anwendung eindeutig zugeordnet. Eine Rolle kann von allen Personen angenommen werden, die sie benötigen. Benutzer besitzen dauerhafte Anmeldeinformationen. Rollen stellen temporäre Anmeldeinformationen bereit. Weitere Informationen finden Sie unter [Erstellen eines IAM-Benutzers \(anstatt einer Rolle\)](#) im IAM-Benutzerhandbuch.

IAM-Rollen

Eine [IAM-Rolle](#) ist eine Identität in Ihrem AWS-Konto mit spezifischen Berechtigungen. Sie ist einem IAM-Benutzer vergleichbar, ist aber nicht mit einer bestimmten Person verknüpft. Sie können vorübergehend eine IAM-Rolle in der AWS Management Console übernehmen, indem Sie [Rollen wechseln](#). Sie können eine Rolle annehmen, indem Sie eine AWS CLI oder AWS-API-Operation aufrufen oder eine benutzerdefinierte URL verwenden. Weitere Informationen zu Methoden für die Verwendung von Rollen finden Sie unter [Verwenden von IAM-Rollen](#) im IAM-Benutzerhandbuch.

IAM-Rollen mit temporären Anmeldeinformationen sind in folgenden Situationen hilfreich:

- Verbundbenutzerzugriff – Um einer Verbundidentität Berechtigungen zuzuweisen, erstellen Sie eine Rolle und definieren Berechtigungen für die Rolle. Wenn eine Verbundidentität

authentifiziert wird, wird die Identität der Rolle zugeordnet und erhält die von der Rolle definierten Berechtigungen. Informationen zu Rollen für den Verbund finden Sie unter [Erstellen von Rollen für externe Identitätsanbieter](#) im IAM-Benutzerhandbuch. Wenn Sie IAM Identity Center verwenden, konfigurieren Sie einen Berechtigungssatz. Wenn Sie steuern möchten, worauf Ihre Identitäten nach der Authentifizierung zugreifen können, korreliert IAM Identity Center den Berechtigungssatz mit einer Rolle in IAM. Informationen zu Berechtigungssätzen finden Sie unter [Berechtigungssätze](#) im AWS IAM Identity Center-Benutzerhandbuch.

- Temporäre IAM-Benutzerberechtigungen – Ein IAM-Benutzer oder eine -Rolle kann eine IAM-Rolle übernehmen, um vorübergehend andere Berechtigungen für eine bestimmte Aufgabe zu erhalten.
- Kontoübergreifender Zugriff – Sie können eine IAM-Rolle verwenden, um einem vertrauenswürdigen Prinzipal in einem anderen Konto den Zugriff auf Ressourcen in Ihrem Konto zu ermöglichen. Rollen stellen die primäre Möglichkeit dar, um kontoübergreifendem Zugriff zu gewähren. In einigen AWS-Services können Sie jedoch eine Richtlinie direkt an eine Ressource anfügen (anstatt eine Rolle als Proxy zu verwenden). Informationen zu den Unterschieden zwischen Rollen und ressourcenbasierten Richtlinien für den kontoübergreifenden Zugriff finden Sie unter [So unterscheiden sich IAM-Rollen von ressourcenbasierten Richtlinien](#) im IAM-Benutzerhandbuch.
- Serviceübergreifender Zugriff – Einige AWS-Services verwenden Features in anderen AWS-Services. Wenn Sie beispielsweise einen Aufruf in einem Service tätigen, führt dieser Service häufig Anwendungen in Amazon EC2 aus oder speichert Objekte in Amazon S3. Ein Service kann dies mit den Berechtigungen des aufrufenden Prinzipals mit einer Servicerolle oder mit einer serviceverknüpften Rolle tun.
 - Forward access sessions (FAS) – Wenn Sie einen IAM-Benutzer oder eine IAM-Rolle zum Ausführen von Aktionen in AWS verwenden, gelten Sie als Prinzipal. Bei einigen Services könnte es Aktionen geben, die dann eine andere Aktion in einem anderen Service initiieren. FAS verwendet die Berechtigungen des Prinzipals, der einen AWS-Service aufruft, in Kombination mit der Anforderung an den AWS-Service, Anforderungen an nachgelagerte Services zu stellen. FAS-Anforderungen werden nur dann gestellt, wenn ein Service eine Anforderung erhält, die Interaktionen mit anderen AWS-Services oder Ressourcen erfordert, um abgeschlossen werden zu können. In diesem Fall müssen Sie über Berechtigungen zum Ausführen beider Aktionen verfügen. Einzelheiten zu den Richtlinien für FAS-Anfragen finden Sie unter [Forward Access Sessions \(FAS\)](#).
- Servicerolle – Eine Servicerolle ist eine [IAM-Rolle](#), die ein Service übernimmt, um Aktionen in Ihrem Namen auszuführen. Ein IAM-Administrator kann eine Servicerolle innerhalb von IAM

erstellen, ändern und löschen. Weitere Informationen finden Sie unter [Erstellen einer Rolle zum Delegieren von Berechtigungen an einen AWS-Service](#) im IAM-Benutzerhandbuch.

- **Serviceverbundene Rolle** – Eine serviceverbundene Rolle ist ein Typ von Servicerolle, die mit einem AWS-Service verknüpft ist. Der Service kann die Rolle übernehmen, um eine Aktion in Ihrem Namen auszuführen. Serviceverknüpfte Rollen werden in Ihrem AWS-Konto angezeigt und gehören zum Service. Ein IAM-Administrator kann die Berechtigungen für serviceverbundene Rollen anzeigen, aber nicht bearbeiten.
- **Anwendungen in Amazon EC2** – Sie können eine IAM-Rolle verwenden, um temporäre Anmeldeinformationen für Anwendungen zu verwalten, die auf einer EC2-Instance ausgeführt werden und AWS CLI- oder AWS-API-Anforderungen durchführen. Das ist eher zu empfehlen, als Zugriffsschlüssel innerhalb der EC2-Instance zu speichern. Erstellen Sie ein Instance-Profil, das an die Instance angefügt ist, um eine AWS-Rolle einer EC2-Instance zuzuweisen und die Rolle für sämtliche Anwendungen der Instance bereitzustellen. Ein Instance-Profil enthält die Rolle und ermöglicht, dass Programme, die in der EC2-Instance ausgeführt werden, temporäre Anmeldeinformationen erhalten. Weitere Informationen finden Sie unter [Verwenden einer IAM-Rolle zum Erteilen von Berechtigungen für Anwendungen, die auf Amazon EC2-Instances ausgeführt werden](#) im IAM-Benutzerhandbuch.

Informationen dazu, wann Sie IAM-Rollen oder IAM-Benutzer verwenden sollten, finden Sie unter [Erstellen einer IAM-Rolle \(anstatt eines Benutzers\)](#) im IAM-Benutzerhandbuch.

Verwalten des Zugriffs mit Richtlinien

Für die Zugriffssteuerung in AWS erstellen Sie Richtlinien und weisen diese den AWS-Identitäten oder -Ressourcen zu. Eine Richtlinie ist ein Objekt in AWS, das, wenn es einer Identität oder Ressource zugeordnet wird, deren Berechtigungen definiert. AWS wertet diese Richtlinien aus, wenn ein Prinzipal (Benutzer, Root-Benutzer oder Rollensitzung) eine Anforderung stellt. Berechtigungen in den Richtlinien bestimmen, ob die Anforderung zugelassen oder abgelehnt wird. Die meisten Richtlinien werden in AWS als JSON-Dokumente gespeichert. Weitere Informationen zu Struktur und Inhalten von JSON-Richtliniendokumenten finden Sie unter [Übersicht über JSON-Richtlinien](#) im IAM-Benutzerhandbuch.

Administratoren können mithilfe von AWS-JSON-Richtlinien festlegen, wer zum Zugriff auf was berechtigt ist. Das bedeutet, welcher Prinzipal kann Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen.

Standardmäßig haben Benutzer, Gruppen und Rollen keine Berechtigungen. Ein IAM-Administrator muss IAM-Richtlinien erstellen, die Benutzern die Berechtigung erteilen, Aktionen für die Ressourcen auszuführen, die sie benötigen. Der Administrator kann dann die IAM-Richtlinien zu Rollen hinzufügen, und Benutzer können die Rollen annehmen.

IAM-Richtlinien definieren Berechtigungen für eine Aktion unabhängig von der Methode, die Sie zur Ausführung der Aktion verwenden. Angenommen, es gibt eine Richtlinie, die Berechtigungen für die `iam:GetRole` -Aktion erteilt. Ein Benutzer mit dieser Richtlinie kann Benutzerinformationen über die AWS Management Console, die AWS CLI oder die AWS -API abrufen.

Identitätsbasierte Richtlinien

Identitätsbasierte Richtlinien sind JSON-Berechtigungsrichtliniendokumente, die Sie einer Identität anfügen können, wie z. B. IAM-Benutzern, -Benutzergruppen oder -Rollen. Diese Richtlinien steuern, welche Aktionen die Benutzer und Rollen für welche Ressourcen und unter welchen Bedingungen ausführen können. Informationen zum Erstellen identitätsbasierter Richtlinien finden Sie unter [Erstellen von IAM-Richtlinien](#) im IAM-Benutzerhandbuch.

Identitätsbasierte Richtlinien können weiter als Inline-Richtlinien oder verwaltete Richtlinien kategorisiert werden. Inline-Richtlinien sind direkt in einen einzelnen Benutzer, eine einzelne Gruppe oder eine einzelne Rolle eingebettet. Verwaltete Richtlinien sind eigenständige Richtlinien, die Sie mehreren Benutzern, Gruppen und Rollen in Ihrem AWS-Konto anfügen können. Verwaltete Richtlinien umfassen von AWS verwaltete und von Kunden verwaltete Richtlinien. Informationen dazu, wie Sie zwischen einer verwalteten Richtlinie und einer eingebundenen Richtlinie wählen, finden Sie unter [Auswahl zwischen verwalteten und eingebundenen Richtlinien](#) im IAM-Benutzerhandbuch.

Ressourcenbasierte Richtlinien

Ressourcenbasierte Richtlinien sind JSON-Richtliniendokumente, die Sie an eine Ressource anfügen. Beispiele für ressourcenbasierte Richtlinien sind IAM-Rollen-Vertrauensrichtlinien und Amazon S3-Bucket-Richtlinien. In Services, die ressourcenbasierte Richtlinien unterstützen, können Service-Administratoren sie verwenden, um den Zugriff auf eine bestimmte Ressource zu steuern. Für die Ressource, an welche die Richtlinie angehängt ist, legt die Richtlinie fest, welche Aktionen ein bestimmter Prinzipal unter welchen Bedingungen für diese Ressource ausführen kann. Sie müssen in einer ressourcenbasierten Richtlinie [einen Prinzipal angeben](#). Prinzipale können Konten, Benutzer, Rollen, Verbundbenutzer oder AWS-Services umfassen.

Ressourcenbasierte Richtlinien sind Richtlinien innerhalb dieses Service. Sie können verwaltete AWS-Richtlinien von IAM nicht in einer ressourcenbasierten Richtlinie verwenden.

Zugriffskontrolllisten (ACLs)

Zugriffssteuerungslisten (ACLs) steuern, welche Prinzipale (Kontomitglieder, Benutzer oder Rollen) auf eine Ressource zugreifen können. ACLs sind ähnlich wie ressourcenbasierte Richtlinien, verwenden jedoch nicht das JSON-Richtliniendokumentformat.

Amazon S3, AWS WAF und Amazon VPC sind Beispiele für Services, die ACLs unterstützen. Weitere Informationen zu ACLs finden Sie unter [Zugriffssteuerungsliste \(ACL\) – Übersicht](#) (Access Control List) im Amazon-Simple-Storage-Service-Entwicklerhandbuch.

Weitere Richtlinientypen

AWS unterstützt zusätzliche, weniger häufig verwendete Richtlinientypen. Diese Richtlinientypen können die maximalen Berechtigungen festlegen, die Ihnen von den häufiger verwendeten Richtlinientypen erteilt werden können.

- **Berechtigungsgrenzen** – Eine Berechtigungsgrenze ist ein erweitertes Feature, mit der Sie die maximalen Berechtigungen festlegen können, die eine identitätsbasierte Richtlinie einer IAM-Entität (IAM-Benutzer oder -Rolle) erteilen kann. Sie können eine Berechtigungsgrenze für eine Entität festlegen. Die daraus resultierenden Berechtigungen sind der Schnittpunkt der identitätsbasierten Richtlinien einer Entität und ihrer Berechtigungsgrenzen. Ressourcenbasierte Richtlinien, die den Benutzer oder die Rolle im Feld `Principal` angeben, werden nicht durch Berechtigungsgrenzen eingeschränkt. Ein ausdrückliches Ablehnen in einer dieser Richtlinien setzt das Zulassen außer Kraft. Weitere Informationen über Berechtigungsgrenzen finden Sie unter [Berechtigungsgrenzen für IAM-Entitäten](#) im IAM-Benutzerhandbuch.
- **Service-Kontrollrichtlinien (SCPs)** – SCPs sind JSON-Richtlinien, die die maximalen Berechtigungen für eine Organisation oder Organisationseinheit (OE) in AWS Organizations angeben. AWS Organizations ist ein Service für die Gruppierung und zentrale Verwaltung mehrerer AWS-Konten Ihres Unternehmens. Wenn Sie innerhalb einer Organisation alle Features aktivieren, können Sie Service-Kontrollrichtlinien (SCPs) auf alle oder einzelne Ihrer Konten anwenden. SCPs schränken Berechtigungen für Entitäten in Mitgliedskonten einschließlich des jeweiligen Root-Benutzer des AWS-Kontos ein. Weitere Informationen zu Organizations und SCPs finden Sie unter [Featuresweise von SCPs](#) im AWS Organizations-Benutzerhandbuch.
- **Sitzungsrichtlinien** – Sitzungsrichtlinien sind erweiterte Richtlinien, die Sie als Parameter übergeben, wenn Sie eine temporäre Sitzung für eine Rolle oder einen verbundenen

Benutzer programmgesteuert erstellen. Die resultierenden Sitzungsberechtigungen sind eine Schnittmenge der auf der Identität des Benutzers oder der Rolle basierenden Richtlinien und der Sitzungsrichtlinien. Berechtigungen können auch aus einer ressourcenbasierten Richtlinie stammen. Eine explizite Zugriffsverweigerung in einer dieser Richtlinien setzt eine Zugriffserlaubnis außer Kraft. Weitere Informationen finden Sie unter [Sitzungsrichtlinien](#) im IAM-Benutzerhandbuch.

Mehrere Richtlinientypen

Wenn mehrere auf eine Anforderung mehrere Richtlinientypen angewendet werden können, sind die entsprechenden Berechtigungen komplizierter. Informationen dazu, wie AWS die Zulässigkeit einer Anforderung ermittelt, wenn mehrere Richtlinientypen beteiligt sind, finden Sie unter [Logik für die Richtlinienauswertung](#) im IAM-Benutzerhandbuch.

Wie das mit IAMAWS Serverless Application Repository funktioniert

Bevor Sie IAM zum Verwalten des Zugriffs auf den verwendenAWS Serverless Application Repository, sollten Sie verstehen, welche IAM-Funktionen für die Verwendung mit verfügbar sindAWS Serverless Application Repository.

Einen Überblick über die Funktionsweise von IAM finden Sie im [IAM-Benutzerhandbuch unter Understanding How IAM Works](#). Einen Überblick über dieAWS Serverless Application Repository Zusammenarbeit mit IAM finden Sie unter [AWS-Services, die mit IAM funktionieren](#) im IAM-Benutzerhandbuch.AWS

Themen

- [Identitätsbasierte AWS Serverless Application Repository-Richtlinien](#)
- [AWS Serverless Application RepositoryAnwendungsrichtlinien](#)
- [Autorisierung auf der Basis von AWS Serverless Application Repository-Tags](#)
- [AWS Serverless Application RepositoryIAM-Rollen](#)

Identitätsbasierte AWS Serverless Application Repository-Richtlinien

Mit identitätsbasierten IAM-Richtlinien können Sie erlaubte oder verweigernde Aktionen und Ressourcen sowie die Bedingungen, unter denen Aktionen erlaubt oder verweigert werden, festlegen. AWS Serverless Application Repository unterstützt bestimmte Aktionen, Ressourcen und Bedingungsschlüssel. Informationen zu sämtlichen Elementen, die Sie in einer JSON-Richtlinie verwenden, finden Sie in der [IAM-Referenz für JSON-Richtlinienelemente](#) im IAM-Benutzerhandbuch.

Hier ein Beispiel für eine Berechtigungsrichtlinie.

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

Die Richtlinie enthält zwei Anweisungen:

- Die erste Anweisung erteilt Berechtigungen für die AWS Serverless Application Repository-Aktion `serverlessrepo:CreateApplication` für alle AWS Serverless Application Repository-Ressourcen, indem das Platzhalterzeichen (*) als Resource-Wert angegeben wird.
- Die zweite Anweisung erteilt die Genehmigung für die AWS Serverless Application Repository Aktion `serverlessrepo:CreateApplicationVersion` an einer AWS Ressource, indem der Amazon Resource Name (ARN) für eine AWS Serverless Application Repository Anwendung verwendet wird. Die Anwendung wird durch den Resource-Wert angegeben.

Das Element `Principal` ist in der Richtlinie nicht angegeben, da in identitätsbasierten Richtlinien die Angabe des Prinzipals als Empfänger der Berechtigung nicht erforderlich ist. Wenn Sie einem Benutzer eine Richtlinie zuweisen, ist der Benutzer automatisch der Prinzipal. Wird die

Berechtigungsrichtlinie einer IAM-Rolle zugewiesen, erhält der in der Vertrauensrichtlinie der Rolle angegebene Prinzipal die Berechtigungen.

Eine Tabelle mit allen AWS Serverless Application Repository API-Vorgängen und den AWS Ressourcen, für die sie gelten, finden Sie unter [AWS Serverless Application Repository API-Berechtigungen: Referenz für Aktionen und Ressourcen](#).

Aktionen

Administratoren können mit AWS-JSON-Richtlinien festlegen, welche Personen zum Zugriff auf welche Ressourcen berechtigt sind. Das heißt, welcher Prinzipal kann Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen.

Das Element `Action` einer JSON-Richtlinie beschreibt die Aktionen, mit denen Sie den Zugriff in einer Richtlinie zulassen oder verweigern können. Richtlinienaktionen haben normalerweise denselben Namen wie die zugehörige AWS-API-Operation. Es gibt einige Ausnahmen, z. B. Aktionen, die nur mit Genehmigung durchgeführt werden können und für die es keinen passenden API-Vorgang gibt. Es gibt auch einige Operationen, die mehrere Aktionen in einer Richtlinie erfordern. Diese zusätzlichen Aktionen werden als abhängige Aktionen bezeichnet.

Schließen Sie Aktionen in eine Richtlinie ein, um Berechtigungen zur Durchführung der zugeordneten Operation zu erteilen.

Richtlinienaktionen im AWS Serverless Application Repository verwenden das folgende Präfix vor der Aktion: `serverlessrepo:`. Um einem Benutzer beispielsweise die Berechtigung zum Ausführen einer AWS Serverless Application Repository-Instance mit der AWS Serverless Application Repository-API-Operation `SearchApplications` zu erteilen, fügen Sie die Aktion `serverlessrepo:SearchApplications` in seine Richtlinie ein. Richtlinienanweisungen müssen entweder ein `Action`- oder ein `NotAction`-Element enthalten. AWS Serverless Application Repository definiert eine eigene Gruppe von Aktionen, die Aufgaben beschreiben, die Sie mit diesem Service durchführen können.

Um mehrere Aktionen in einer einzigen Anweisung anzugeben, trennen Sie sie wie folgt durch Kommata:

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

Sie können auch Platzhalter verwenden, um mehrere Aktionen anzugeben. Beispielsweise können Sie alle Aktionen festlegen, die mit dem Wort `List` beginnen, einschließlich der folgenden Aktion:

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

Eine Liste der AWS Serverless Application Repository-Aktionen finden Sie im IAM-Benutzerhandbuch unter [Aktionen, die von AWS Serverless Application Repository definiert werden](#).

Ressourcen

Administratoren können mit AWS-JSON-Richtlinien festlegen, welche Personen zum Zugriff auf welche Ressourcen berechtigt sind. Das bedeutet die Festlegung, welcher Prinzipal Aktionen für welche Ressourcen unter welchen Bedingungen ausführen kann.

Das JSON-Richtlinienelement `Resource` gibt die Objekte an, auf welche die Aktion angewendet wird. Anweisungen müssen entweder ein `Resource`- oder ein `NotResource`-Element enthalten. Als bewährte Methode geben Sie eine Ressource mit dem zugehörigen [Amazon-Ressourcennamen \(ARN\)](#) an. Sie können dies für Aktionen tun, die einen bestimmten Ressourcentyp unterstützen, der als Berechtigungen auf Ressourcenebene bezeichnet wird.

Verwenden Sie für Aktionen, die keine Berechtigungen auf Ressourcenebene unterstützen, z. B. Auflistungsoperationen, einen Platzhalter (`*`), um anzugeben, dass die Anweisung für alle Ressourcen gilt.

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

In der AWS Serverless Application Repository ist die primäre AWS Ressource eine AWS Serverless Application Repository Anwendung. AWS Serverless Application Repository Anwendungen sind eindeutigen Amazon-Ressourcennamen (ARNs) zugeordnet, siehe nachfolgende Tabelle.

AWS-Ressourcentyp	Format des Amazon-Ressourcennamens (ARN)
Anwendung	<code>arn:<i>partition</i> :serverlessrepo:<i>region</i>:<i>account-id</i> :applications/<i>application-name</i></code>

Weitere Informationen zum Format von ARNs finden Sie unter [Amazon-Ressourcennamen \(ARNs\) und AWS-Service-Namespaces](#).

Im Folgenden finden Sie ein Beispiel für eine Richtlinie, die Berechtigungen für `serverlessrepo:ListApplications` Aktion für alle AWS Ressourcen gewährt. In einigen der API-Aktionen unterstützt der AWS Serverless Application Repository nicht die Identifizierung bestimmter AWS Ressourcen unter Verwendung von AWS Ressourcen-ARNs (auch als Berechtigungen auf Ressourcenebene bezeichnet). In diesen Fällen müssen Sie ein Platzhalterzeichen (*) angeben.

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

Eine Tabelle mit allen AWS Serverless Application Repository API-Aktionen und den AWS Ressourcen, für die sie gelten, finden Sie unter [AWS Serverless Application Repository API-Berechtigungen: Referenz für Aktionen und Ressourcen](#).

Bedingungsschlüssel

Das AWS Serverless Application Repository stellt keine servicespezifischen Bedingungsschlüssel bereit, unterstützt jedoch die Verwendung einiger globaler Bedingungsschlüssel. Eine Liste aller globalen AWS-Bedingungsschlüssel finden Sie unter [Globale AWS-Bedingungskontextschlüssel](#) im IAM-Benutzerhandbuch.

Beispiele

Beispiele für identitätsbasierte AWS Serverless Application Repository-Richtlinien finden Sie unter [Beispiele für identitätsbasierte AWS Serverless Application Repository-Richtlinien](#).

AWS Serverless Application Repository Anwendungsrichtlinien

Anwendungsrichtlinien legen fest, welche Aktionen ein bestimmter Principal oder PrincipalOrg für eine AWS Serverless Application Repository Anwendung ausführen kann.

Sie können der Richtlinie, die einer AWS Serverless Application Repository-Anwendung zugeordnet ist, Berechtigungen hinzufügen. An AWS Serverless Application Repository Anwendungen angehängte Berechtigungsrichtlinien werden als Anwendungsrichtlinien bezeichnet. [Anwendungsrichtlinien](#) sind Erweiterungen der [ressourcenbasierten IAM-Richtlinien](#). Die primäre Ressource ist die AWS Serverless Application Repository Anwendung. Sie können AWS Serverless Application Repository-Anwendungsrichtlinien zum Verwalten von Berechtigungen für die Anwendungsbereitstellung verwenden.

AWS Serverless Application Repository-Anwendungsrichtlinien werden in erster Linie von Herausgebern verwendet, um Verbrauchern die Berechtigung zur Bereitstellung ihrer Anwendungen zu erteilen, sowie damit zusammenhängende Vorgänge wie Suchen und Anzeigen von Details dieser Anwendungen. Herausgeber können Anwendungsberechtigungen mit den folgenden drei Kategorien festlegen:

- **Privat** — Anwendungen, die mit demselben Konto erstellt und mit keinem anderen Konto geteilt wurden. Sie sind berechtigt, Anwendungen bereitzustellen, die mit Ihrem AWS Konto erstellt wurden.
- **Privat geteilt** — Anwendungen, die der Herausgeber ausdrücklich für bestimmte AWS Konten oder AWS Organizations freigegeben hat. Sie sind berechtigt, Anwendungen bereitzustellen, die mit Ihrem AWS Konto oder Ihrer AWS Organisation geteilt wurden.
- **Öffentlich geteilt** — Anwendungen, die der Herausgeber für alle freigegeben hat. Sie haben die Berechtigung, alle öffentlich freigegebenen Anwendungen bereitzustellen.

Sie können Berechtigungen gewähren, indem Sie die AWS CLI, die AWS SDKs oder die verwenden AWS Management Console.

Beispiele

Beispiele für die Verwaltung von AWS Serverless Application Repository Anwendungsrichtlinien finden Sie unter [AWS Serverless Application Repository Beispiele für Anwendungsrichtlinien](#).

Autorisierung auf der Basis von AWS Serverless Application Repository-Tags

Das AWS Serverless Application Repository unterstützt nicht das Steuern des Zugriffs auf Ressourcen oder Aktionen, die auf Tags basieren.

AWS Serverless Application Repository IAM-Rollen

Eine [IAM-Rolle](#) ist eine Entität in Ihrem AWS-Konto mit spezifischen Berechtigungen.

Verwenden temporärer Anmeldeinformationen mit AWS Serverless Application Repository

Sie können temporäre Anmeldeinformationen verwenden, um sich über einen Verbund anzumelden, eine IAM-Rolle anzunehmen oder eine kontenübergreifende Rolle anzunehmen. Temporäre Sicherheitsanmeldeinformationen erhalten Sie durch Aufrufen von AWS STS -API-Vorgängen wie [AssumeRole](#) oder [GetFederationToken](#).

Das AWS Serverless Application Repository unterstützt die Verwendung temporärer Anmeldeinformationen.

Serviceverknüpfte Rollen

Das AWS Serverless Application Repository unterstützt keine serviceverknüpften Rollen.

Servicerollen

Das AWS Serverless Application Repository unterstützt keine Service-Rollen.

Beispiele für identitätsbasierte AWS Serverless Application Repository-Richtlinien

IAM-Benutzer besitzen keine Berechtigungen zum Erstellen oder Ändern von AWS Serverless Application Repository-Ressourcen. Sie können auch keine Aufgaben ausführen, die die AWS Management Console-, AWS CLI- oder AWS-API benutzen. Ein IAM-Administrator muss IAM-Richtlinien erstellen, die Benutzern und Rollen die Berechtigung zum Ausführen bestimmter API-Operationen für die angegebenen Ressourcen gewähren, die diese benötigen. Der Administrator muss diese Richtlinien anschließend den IAM-Benutzern oder -Gruppen anfügen, die diese Berechtigungen benötigen.

Informationen dazu, wie Sie unter Verwendung dieser beispielhaften JSON-Richtliniendokumente eine identitätsbasierte IAM-Richtlinie erstellen, finden Sie unter [Erstellen von Richtlinien auf der JSON-Registerkarte](#) im IAM-Benutzerhandbuch.

Themen

- [Bewährte Methoden für Richtlinien](#)
- [Verwenden der AWS Serverless Application Repository-Konsole](#)
- [Gewähren der Berechtigung zur Anzeige der eigenen Berechtigungen für Benutzer](#)
- [Beispiele für vom Kunden verwaltete Richtlinien](#)

Bewährte Methoden für Richtlinien

Identitätsbasierte Richtlinien sind sehr leistungsfähig. Sie können festlegen, ob jemand AWS Serverless Application Repository-Ressourcen in Ihrem Konto erstellen oder löschen oder auf sie zugreifen kann. Dies kann zusätzliche Kosten für Ihr AWS-Konto verursachen. Befolgen Sie beim Erstellen oder Bearbeiten identitätsbasierter Richtlinien die folgenden Anleitungen und Empfehlungen:

- **Gewähren von geringsten Rechten** – Gewähren Sie beim Erstellen benutzerdefinierter Richtlinien nur die Berechtigungen, die zum Ausführen einer Aufgabe erforderlich sind. Beginnen Sie mit einem Mindestsatz von Berechtigungen und gewähren Sie zusätzliche Berechtigungen wie erforderlich. Dies ist sicherer, als mit Berechtigungen zu beginnen, die zu weit gefasst sind, und dann später zu versuchen, sie zu begrenzen. Weitere Informationen finden Sie unter [Gewähren von geringsten Rechten](#) im IAM-Benutzerhandbuch.
- **Aktivieren von MFA für sensible Vorgänge** – Sie sollten die Verwendung der Multi-Faktor-Authentifizierung (MFA) von IAM-Benutzern fordern, um beim Zugriff auf sensible Ressourcen oder API-Operationen zusätzliche Sicherheit zu erhalten. Weitere Informationen finden Sie unter [Verwenden der Multi-Faktor-Authentifizierung \(MFA\) in AWS](#) im IAM-Benutzerhandbuch.
- **Verwenden von Richtlinienbedingungen für zusätzliche Sicherheit** – Definieren Sie die Bedingungen, unter denen Ihre identitätsbasierten Richtlinien den Zugriff auf eine Ressource zulassen, soweit praktikabel. Beispielsweise können Sie Bedingungen schreiben, die eine Reihe von zulässigen IP-Adressen festlegen, von denen eine Anforderung stammen muss. Sie können auch Bedingungen schreiben, die Anforderungen nur innerhalb eines bestimmten Datums- oder Zeitbereichs zulassen oder die Verwendung von SSL oder MFA fordern. Weitere Informationen finden Sie unter [IAM-JSON-Richtlinienelemente: Bedingung](#) im IAM-Benutzerhandbuch.

Verwenden der AWS Serverless Application Repository-Konsole

Die AWS Serverless Application Repository-Konsole bietet eine integrierte Umgebung zum Erkennen und Verwalten von AWS Serverless Application Repository-Anwendungen. Die Konsole bietet viele Funktionen und Workflows, die häufig Berechtigungen zum Verwalten einer AWS Serverless Application Repository-Anwendung erfordern, sowie API-spezifische Berechtigungen, die unter [AWS Serverless Application Repository API-Berechtigungen: Referenz für Aktionen und Ressourcen](#) dokumentiert sind.

Weitere Informationen zu den erforderlichen Berechtigungen für die Verwendung der AWS Serverless Application Repository-Konsole finden Sie unter [Beispiele für vom Kunden verwaltete Richtlinien](#).

Gewähren der Berechtigung zur Anzeige der eigenen Berechtigungen für Benutzer

In diesem Beispiel wird gezeigt, wie Sie eine Richtlinie erstellen, die IAM-Benutzern die Berechtigung zum Anzeigen der eingebundenen Richtlinien und verwalteten Richtlinien gewährt, die ihrer Benutzeridentität angefügt sind. Diese Richtlinie enthält Berechtigungen für die Ausführung dieser Aktion auf der Konsole oder für die programmgesteuerte Ausführung über die AWS CLI oder die AWS-API.

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

Beispiele für vom Kunden verwaltete Richtlinien

In den Beispielen in diesem Abschnitt finden Sie eine Gruppe von Beispielrichtlinien, die Sie Benutzern zuweisen können. Wenn Sie mit dem Erstellen von Richtlinien noch nicht vertraut sind, sollten Sie zunächst einen IAM-Benutzer in Ihrem Konto erstellen und ihm dann die Richtlinien der Reihe nach zuweisen. Sie können diese Beispiele auch verwenden, um eine einzelne benutzerdefinierte Richtlinie zu erstellen, die Berechtigungen zum Ausführen mehrerer Aktionen enthält, und sie dann an den Benutzer anfügen.

Weitere Informationen zum Anhängen von Richtlinien an Benutzer finden Sie im [IAM-Benutzerhandbuch unter Adding Permissions to a User Guide](#).

Beispiele

- [Publisher-Beispiel 1: Berechtigen eines Publishers zum Auflisten von Anwendungen](#)
- [Publisher-Beispiel 2: Berechtigen eines Publishers zum Anzeigen von Details zu einer Anwendung oder der Anwendungsversion](#)
- [Publisher-Beispiel 3: Berechtigen eines Publishers zum Erstellen einer Anwendung oder Anwendungsversion](#)
- [Publisher-Beispiel 4: Berechtigen eines Publishers zum Erstellen einer Anwendungsrichtlinie zur gemeinsamen Nutzung von Anwendungen mit anderen](#)
- [Konsumenten-Beispiel 1: Berechtigen eines Konsumenten zum Suchen nach Anwendungen](#)
- [Konsumenten-Beispiel 2: Berechtigen eines Konsumenten zum Anzeigen von Details zu einer Anwendung](#)
- [Konsumenten-Beispiel 3: Berechtigen eines Konsumenten zum Bereitstellen einer Anwendung](#)
- [Beispiel 4: Verweigern von Zugriff auf Bereitstellungsressourcen](#)
- [Beispiel 5: Verhindern, dass Konsumenten öffentliche Anwendungen suchen und bereitstellen](#)

Publisher-Beispiel 1: Berechtigen eines Publishers zum Auflisten von Anwendungen

Ein IAM-Benutzer in Ihrem Konto muss über Berechtigungen für die `serverlessrepo:ListApplications`-Operation verfügen, bevor der Benutzer etwas in der Konsole sehen kann. Wenn Sie diese Berechtigungen gewähren, kann die Konsole die Liste der AWS Serverless Application Repository Anwendungen in dem AWS Konto anzeigen, das in der spezifischen AWS Region erstellt wurde, zu der der Benutzer gehört.

```
{
```

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

Publisher-Beispiel 2: Berechtigen eines Publishers zum Anzeigen von Details zu einer Anwendung oder der Anwendungsversion

Ein Benutzer kann eine AWS Serverless Application Repository-Anwendung auswählen und Details der Anwendung anzeigen. Diese Details umfassen Autor, Beschreibung, Versionen und andere Konfigurationsinformationen. Zu diesem Zweck benötigt der Benutzer Berechtigungen für die API-Operationen `serverlessrepo:GetApplication` und `serverlessrepo:ListApplicationVersions` für das AWS Serverless Application Repository.

Im folgenden Beispiel werden diese Berechtigungen für die betreffende Anwendung gewährt, deren Amazon-Ressourcenname (ARN) als Resource-Wert angegeben wird.

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

Publisher-Beispiel 3: Berechtigen eines Publishers zum Erstellen einer Anwendung oder Anwendungsversion

Wenn Sie einem Benutzer Berechtigungen zum Erstellen von AWS Serverless Application Repository-Anwendungen gewähren möchten, müssen Sie wie in der folgenden Richtlinie veranschaulicht Berechtigungen für die Operationen `serverlessrepo:CreateApplication` und `serverlessrepo:CreateApplicationVersions` gewähren.

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

Publisher-Beispiel 4: Berechtigen eines Publishers zum Erstellen einer Anwendungsrichtlinie zur gemeinsamen Nutzung von Anwendungen mit anderen

Damit Benutzer Anwendungen gemeinsam mit anderen nutzen können, müssen Sie ihnen, wie in der folgenden Richtlinie gezeigt, Berechtigungen zum Erstellen von Anwendungsrichtlinien gewähren.

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

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

Konsumenten-Beispiel 1: Berechtigen eines Konsumenten zum Suchen nach Anwendungen

Damit Konsumenten nach Anwendungen suchen können, müssen Sie ihnen die folgenden Berechtigungen gewähren.

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

Konsumenten-Beispiel 2: Berechtigen eines Konsumenten zum Anzeigen von Details zu einer Anwendung

Ein Benutzer kann eine AWS Serverless Application Repository-Anwendung auswählen und Details der Anwendung, z. B. Autor, Beschreibung, Versionen und andere Konfigurationsinformationen, anzeigen. Dazu benötigt der Benutzer Berechtigungen für die folgenden AWS Serverless Application Repository-Operationen.

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

```

        "Sid": "ViewApplication",
        "Effect": "Allow",
        "Action": [
            "serverlessrepo:GetApplication",
            "serverlessrepo:ListApplicationVersions"
        ],
        "Resource": "*"
    }
]
}

```

Konsumenten-Beispiel 3: Berechtigen eines Konsumenten zum Bereitstellen einer Anwendung

Damit Kunden Anwendungen bereitstellen können, müssen Sie ihnen Berechtigungen für das Durchführen einer Reihe von Operationen gewähren. Die folgende Richtlinie erteilt den Kunden die erforderlichen Berechtigungen.

```

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

```

Note

Für die Bereitstellung einer Anwendung sind möglicherweise Berechtigungen zur Nutzung zusätzlicher AWS Ressourcen erforderlich. Weil der denselben zugrunde liegenden Bereitstellungsmechanismus AWS Serverless Application Repository verwendet wie AWS

CloudFormation weitere Informationen unter [Steuern des Zugriffs mit AWS Identity and Access Management](#). Hilfe zu Bereitstellungsproblemen im Zusammenhang mit Berechtigungen finden Sie unter [Fehlerbehebung: Unzureichende IAM-Berechtigungen](#).

Beispiel 4: Verweigern von Zugriff auf Bereitstellungsressourcen

Wenn eine Anwendung privat mit einem AWS Konto geteilt wird, können standardmäßig alle Benutzer dieses Kontos auf die Deployment-Assets aller anderen Benutzer desselben Kontos zugreifen. Die folgende Richtlinie verhindert, dass Benutzer in einem Konto auf Deployment-Assets zugreifen, die im Amazon S3 S3-Bucket für die gespeichert sind AWS Serverless Application Repository.

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

Beispiel 5: Verhindern, dass Konsumenten öffentliche Anwendungen suchen und bereitstellen

Sie können verhindern, dass Benutzer bestimmte Aktionen für Anwendungen ausführen.

Die folgende Richtlinie gilt für öffentliche Anwendungen, indem für `serverlessrepo:applicationType public` angegeben wird. Sie verhindert, dass Benutzer eine Reihe von Aktionen ausführen, indem für Effect Deny angegeben wird. Weitere Informationen zu Bedingungsschlüsseln, die für AWS Serverless Application Repository verfügbar sind, finden Sie unter [Aktionen, Ressourcen und Bedingungsschlüssel für 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

Diese Richtlinienerklärung kann auch als Service Control Policy verwendet und auf eine AWS Organisation angewendet werden. Weitere Informationen zu Service Control-Richtlinien finden Sie unter [Service Control-Richtlinien](#) im AWS Organizations Benutzerhandbuch.

AWS Serverless Application Repository Beispiele für Anwendungsrichtlinien

An AWS Serverless Application Repository Anwendungen angehängte Berechtigungsrichtlinien werden als Anwendungsrichtlinien bezeichnet. Anwendungsrichtlinien legen fest, welche Aktionen ein bestimmter Principal oder PrincipalOrg für eine AWS Serverless Application Repository Anwendung ausführen kann.

Eine AWS Serverless Application Repository Anwendung ist die primäre AWS Ressource in der AWS Serverless Application Repository. AWS Serverless Application Repository Anwendungsrichtlinien werden in erster Linie von Verlagen verwendet, um Verbrauchern die Genehmigung zur Bereitstellung ihrer Anwendungen und damit verbundenen Vorgängen zu erteilen, z. B. zum Suchen und Anzeigen von Details zu diesen Anwendungen.

Herausgeber können Anwendungsberechtigungen mit den folgenden drei Kategorien festlegen:

- **Privat** — Anwendungen, die mit demselben Konto erstellt und mit keinem anderen Konto geteilt wurden. Nur Verbraucher, die Ihr AWS Konto teilen, haben die Erlaubnis, private Anwendungen bereitzustellen.
- **Privat geteilt** — Anwendungen, die der Herausgeber ausdrücklich für eine bestimmte Gruppe von AWS Konten oder für AWS Konten in einer AWS Organisation freigegeben hat. Verbraucher haben die Erlaubnis, Anwendungen bereitzustellen, die mit ihrem AWS Konto oder ihrer AWS Organisation geteilt wurden. Weitere Informationen zu AWS Organisationen finden Sie im [AWS Organizations Benutzerhandbuch](#).
- **Öffentlich geteilt** — Anwendungen, die der Herausgeber für alle freigegeben hat. Alle Verbraucher haben die Berechtigung, alle öffentlich freigegebenen Anwendungen bereitzustellen.

Note

Für privat gemeinsam genutzte Anwendungen werden AWS Serverless Application Repository nur AWS Konten als Prinzipale unterstützt. Herausgeber können allen Benutzern innerhalb eines AWS Kontos als einzelne Gruppe eine AWS Serverless Application Repository Anwendung zuweisen oder verweigern. Herausgeber können einzelnen Benutzern innerhalb eines AWS Kontos keine AWS Serverless Application Repository Anwendung gewähren oder verweigern.

Anweisungen zum Festlegen von Anwendungsberechtigungen mithilfe der AWS Management Console finden Sie unter [Freigeben einer Anwendung](#).

Anweisungen zum Festlegen von Anwendungsberechtigungen mithilfe der AWS CLI sowie entsprechende Beispiele finden Sie in den folgenden Abschnitten.

Anwendungsberechtigungen (AWS CLI und AWS SDKs)

Wenn Sie die SDKs AWS CLI oder die AWS SDKs verwenden, um Berechtigungen für eine AWS Serverless Application Repository Anwendung festzulegen, können Sie die folgenden Aktionen angeben:

Action	Beschreibung
GetApplication	Erteilt die Berechtigung zum Anzeigen von Informationen zur Anwendung.

Action	Beschreibung
CreateCloudFormationChangeSet	Erteilt die Berechtigung zum Bereitstellen der Anwendung. Hinweis: Diese Aktion gewährt keine anderen Berechtigungen außer der Bereitstellung.
CreateCloudFormationTemplate	Erteilt die Berechtigung zum Erstellen einer AWS CloudFormation-Vorlage für die Anwendung.
ListApplicationVersions	Erteilt die Berechtigung zum Auflisten der Versionen der Anwendung.
ListApplicationDependencies	Erteilt die Berechtigung zum Auflisten der Listenanwendungen, die in der enthaltenen Anwendung verschachtelt sind.
SearchApplications	Erteilt die Berechtigung zum Suchen nach der Anwendung.
Bereitstellen	Mit dieser Aktion werden alle zuvor in der Tabelle aufgeführten Aktionen aktiviert. Das heißt, sie erteilt die Berechtigung für die Anzeige der Anwendung, für die Bereitstellung, für die Liste der Versionen und für die Suche.

Beispiele für Benutzerrichtlinien

In den folgenden Beispielen wird gezeigt, wie Berechtigungen mithilfe der AWS CLI erteilt werden. Weitere Informationen zum Erteilen von Berechtigungen unter Verwendung der AWS Management Console finden Sie unter [Freigeben einer Anwendung](#).

In allen Beispielen in diesem Abschnitt werden diese AWS CLI-Befehle zum Verwalten von Berechtigungsrichtlinien verwendet, die mit AWS Serverless Application Repository-Anwendungen verknüpft sind:

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

Themen

- [Beispiel 1: Freigeben einer Anwendung für ein anderes Konto](#)
- [Beispiel 2: Freigeben einer Anwendung zur öffentlichen gemeinsamen Nutzung](#)

- [Beispiel 3: Einschränken einer Anwendung auf private Nutzung](#)
- [Beispiel 4: Angeben mehrerer Konten und Berechtigungen](#)
- [Beispiel 5: Teilen Sie eine Anwendung mit allen Konten in einerAWS Organisation](#)
- [Beispiel 6: Gemeinsame Nutzung einer Anwendung mit einigen Konten in einerAWS Organisation](#)
- [Beispiel 7: Abrufen einer Anwendungsrichtlinie](#)
- [Beispiel 8: Zulassen, dass die Anwendung nach bestimmten Konten verschachtelt wird](#)

Beispiel 1: Freigeben einer Anwendung für ein anderes Konto

Um eine Anwendung mit einem anderen bestimmten Konto zu teilen, aber zu verhindern, dass sie mit anderen geteilt wird, geben Sie dieAWS Konto-ID, mit der Sie sie teilen möchten, als Principal an. Dies wird auch als Festlegen der Anwendung auf privat freigegeben bezeichnet. Verwenden Sie dazu den folgenden AWS CLI-Befehl.

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

Note

Privat gemeinsam genutzte Anwendungen können nur in derselbenAWS Region verwendet werden, in der die Anwendung erstellt wurde.

Beispiel 2: Freigeben einer Anwendung zur öffentlichen gemeinsamen Nutzung

Um eine Anwendung öffentlich gemeinsam nutzbar zu machen, müssen Sie sie wie im folgenden Beispiel durch Angabe von "*" als Prinzipal für jeden freigeben. Anwendungen, die öffentlich freigegeben werden, sind in allen Regionen verfügbar.

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

Note

Um eine Anwendung öffentlich freigeben zu können, müssen die Eigenschaften `SemanticVersion` und `LicenseUrl` festgelegt sein.

Beispiel 3: Einschränken einer Anwendung auf private Nutzung

Sie können eine Anwendung privat machen, sodass sie mit niemandem geteilt wird und nur von dem AWS Konto bereitgestellt werden kann, dem sie gehört. Dazu löschen Sie die Prinzipien und Aktionen aus der Richtlinie, wodurch auch die Berechtigungen anderer Konten innerhalb Ihrer AWS Organisation für die Bereitstellung Ihrer Anwendung entzogen werden.

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

Note

Private Anwendungen können nur in derselben AWS Region verwendet werden, in der die Anwendung erstellt wurde.

Beispiel 4: Angeben mehrerer Konten und Berechtigungen

Sie können mehrere Berechtigungen erteilen, und Sie können sie mehreren AWS Konten gleichzeitig gewähren. Dazu geben Sie Listen als Prinzipal und Aktionen an, wie im folgenden Beispiel gezeigt.

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

Beispiel 5: Teilen Sie eine Anwendung mit allen Konten in einer AWS Organisation

Berechtigungen können allen Benutzern innerhalb einer AWS Organisation erteilt werden. Dazu geben Sie Ihre Organisations-ID an, wie im folgenden Beispiel.

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

Weitere Informationen zu AWS Organisationen finden Sie im [AWS Organizations Benutzerhandbuch](#).

Note

Sie können nur die AWS Organisation angeben, der Ihr AWS Konto angehört. Wenn Sie versuchen, eine AWS Organisation anzugeben, der Sie nicht angehören, wird ein Fehler angezeigt.

Um Ihre Bewerbung mit Ihrer AWS Organisation zu teilen, müssen Sie die Genehmigung für die `UnshareApplication` Aktion beifügen, falls die Weitergabe in future widerrufen werden muss.

Beispiel 6: Gemeinsame Nutzung einer Anwendung mit einigen Konten in einer AWS Organisation

Berechtigungen können bestimmten Konten innerhalb einer AWS Organisation erteilt werden. Dazu geben Sie eine Liste von AWS Konten als Principal und Ihre Organisations-ID an, wie im folgenden Beispiel.

```
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

Sie können nur die AWS Organisation angeben, der Ihr AWS Konto angehört. Wenn Sie versuchen, eine AWS Organisation anzugeben, der Sie nicht angehören, wird ein Fehler angezeigt.

Um Ihre Bewerbung mit Ihrer AWS Organisation zu teilen, müssen Sie die Genehmigung für die `UnshareApplication` Aktion beifügen, falls die Weitergabe in future widerrufen werden muss.

Beispiel 7: Abrufen einer Anwendungsrichtlinie

Um die aktuelle Richtlinie einer Anwendung anzuzeigen, etwa um nachzusehen, ob sie derzeit freigegeben ist, verwenden Sie den Befehl `get-application-policy`, wie im folgenden Beispiel gezeigt.

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

Beispiel 8: Zulassen, dass die Anwendung nach bestimmten Konten verschachtelt wird

Öffentliche Anwendungen dürfen von jedem Benutzer verschachtelt werden. Wenn Sie möchten, dass Ihre Anwendung nur von bestimmten Konten verschachtelt werden darf, müssen Sie die folgenden minimalen Berechtigungen festlegen, wie im folgenden Beispiel gezeigt.

```
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-API-Berechtigungen: Referenz für Aktionen und Ressourcen

Wenn Sie die [Zugriffskontrolle](#) einrichten und Berechtigungsrichtlinien für eine IAM-Identität (identitätsbasierte Richtlinie) verfassen, können Sie die folgende Tabelle als Referenz verwenden. Die jeder AWS Serverless Application Repository-API-Operationen, die zugehörigen Aktionen, für die Sie Berechtigungen erteilen können, AWS Die Ressource, für die Sie die Berechtigungen erteilen können. Die Aktionen geben Sie im Feld `Action` und den Wert für die Ressource im Feld `Resource` der Richtlinie an.

Um eine Aktion anzugeben, verwenden Sie das Präfix `serverlessrepo:` gefolgt vom Namen der API-Operation (z. B. `serverlessrepo:ListApplications`).

Operation	URI	Method	AWSRessourcen (ARNs)
verwenden: ListApplications Erforderliche Berechtigungen: serverlessrepo:ListApplications	/applications	GET	*
verwenden: CreateApplication Erforderliche Berechtigungen: serverlessrepo:CreateApplication	/applications	POST	*
verwenden: GetApplication Erforderliche Berechtigungen: serverlessrepo:GetApplication	/applications/ <i>Anwendungs-ID</i>	GET	arn:aws:serverlessrepo: <i>Region</i> : <i>Konto-ID</i> :applications/ <i>Anwendung sname</i>
verwenden: DeleteApplication Erforderliche Berechtigungen: serverlessrepo>DeleteApplication	/applications/ <i>Anwendungs-ID</i>	DELETE	arn:aws:serverlessrepo: <i>Region</i> : <i>Konto-ID</i> :applications/ <i>Anwendung sname</i>

Operation	URI	Method	AWSRessourcen (ARNs)
verwenden: UpdateApplication Erforderliche Berechtigungen: serverlessrepo:UpdateApplication	/applications/ <i>Anwendungs-ID</i>	PATCH	arn:aws:serverlessrepo: <i>Region</i> : <i>Konto-ID</i> :application/ <i>Anwendung</i> <i>sname</i>
verwenden: CreateCloudFormationChangeLegen Sie fest. Erforderliche Berechtigungen: serverlessrepo:CreateCloudFormationChangeLegen Sie fest.	/applications/ <i>Anwendungs-ID</i> /changesets	POST	arn:aws:serverlessrepo: <i>Region</i> : <i>Konto-ID</i> :application/ <i>Anwendung</i> <i>sname</i>
verwenden: GetApplication--Richtlinie Erforderliche Berechtigungen: serverlessrepo:GetApplication--Richtlinie	/applications/ <i>Anwendungs-ID</i> /policy	GET	arn:aws:serverlessrepo: <i>Region</i> : <i>Konto-ID</i> :application/ <i>Anwendung</i> <i>sname</i>

Operation	URI	Method	AWSRessourcen (ARNs)
verwenden: PutApplication--Richtlinie Erforderliche Berechtigungen: serverlessrepo:PutApplication--Richtlinie	/applications/ <i>Anwendungs-ID</i> /policy	PUT	arn:aws:serverlessrepo: <i>Region</i> : <i>Konto-ID</i> :application/ <i>Anwendungsname</i>
verwenden: ListApplication-Versionen Erforderliche Berechtigungen: serverlessrepo:ListApplication-Versionen	/applications/ <i>Anwendungs-ID</i> /versions	GET	arn:aws:serverlessrepo: <i>Region</i> : <i>Konto-ID</i> :application/ <i>Anwendungsname</i>
verwenden: CreateApplicationVersion Erforderliche Berechtigungen: serverlessrepo:CreateApplicationVersion	/applications/ <i>Anwendungs-ID</i> /versions/ <i>semantische Version</i>	PUT	arn:aws:serverlessrepo: <i>Region</i> : <i>Konto-ID</i> :application/ <i>Anwendungsname</i>

Operation	URI	Method	AWSRessourcen (ARNs)
verwenden: ListApplicationAbhängigkeiten Erforderliche Berechtigungen: serverlessrepo:ListApplicationAbhängigkeiten	/applications/ <i>Anwendungs-ID</i> /dependencies	GET	arn:aws:serverlessrepo: <i>Region</i> : <i>Konto-ID</i> :application/ <i>Anwendungssname</i>
verwenden: SearchApplications Erforderliche Berechtigungen: serverlessrepo:SearchApplications	–	–	*

Fehlerbehebung bei AWS Serverless Application Repository-Identität und -Zugriff

Mit den folgenden Informationen können, diagnostizieren und beheben, die beim Arbeiten mit dem AWS Serverless Application Repository und IAM auftreten könnten.

Themen

- [Ich bin nicht autorisiert, eine Aktion in AWS Serverless Application Repository auszuführen](#)
- [Ich bin nicht berechtigt, iam auszuführen:PassRole](#)
- [Ich bin Administrator und möchte anderen den Zugriff auf das AWS Serverless Application Repository ermöglichen](#)
- [Ich möchte Personen außerhalb meines AWS-Kontos Zugriff auf meine AWS Serverless Application Repository-Ressourcen gewähren](#)

Ich bin nicht autorisiert, eine Aktion in AWS Serverless Application Repository auszuführen

Wenn die AWS Management Console Ihnen mitteilt, dass Sie nicht zur Ausführung einer Aktion autorisiert sind, müssen Sie sich an Ihren Administrator wenden, um Unterstützung zu erhalten. Ihr Administrator ist die Person, die Ihnen Ihren Benutzernamen und Ihr Passwort bereitgestellt hat.

Der folgende Beispielfehler tritt auf, wenn der `mateojackson` -IAM-Benutzer versucht, die Konsole zum Anzeigen von Details zu einer Anwendung zu verwenden, jedoch nicht über `serverlessrepo:GetApplication` -Berechtigungen verfügt.

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

In diesem Fall bittet Mateo seinen Administrator, seine Richtlinien zu aktualisieren, um ihm den Zugriff auf die `my-example-application`-Ressource mit der `serverlessrepo:GetApplication`-Operation zu ermöglichen.

Ich bin nicht berechtigt, iam auszuführen:PassRole

Wenn Sie die Fehlermeldung erhalten, dass Sie nicht zum Durchführen der `iam:PassRole`-Aktion autorisiert sind, müssen Ihre Richtlinien aktualisiert werden, um eine Rolle an AWS Serverless Application Repository übergeben zu können.

Einige AWS-Services erlauben die Übergabe einer vorhandenen Rolle an diesen Service, sodass keine neue Servicerolle oder serviceverknüpfte Rolle erstellt werden muss. Hierzu benötigen Sie Berechtigungen für die Übergabe der Rolle an den Service.

Der folgende Beispielfehler tritt auf, wenn ein IAM-Benutzer mit dem Namen `marymajor` versucht, die Konsole zu verwenden, um eine Aktion in AWS Serverless Application Repository auszuführen. Die Aktion erfordert jedoch, dass der Service über Berechtigungen verfügt, die durch eine Servicerolle gewährt werden. Mary besitzt keine Berechtigungen für die Übergabe der Rolle an den Service.

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

In diesem Fall müssen die Richtlinien von Mary aktualisiert werden, um die Aktion `iam:PassRole` ausführen zu können.

Wenden Sie sich an Ihren AWS-Administrator, falls Sie weitere Unterstützung benötigen. Ihr Administrator hat Ihnen Ihre Anmeldeinformationen odzur Verfügung gestellt.

Ich bin Administrator und möchte anderen den Zugriff auf das AWS Serverless Application Repository ermöglichen

Um anderen Personen oder einer Anwendung Zugriff auf AWS Serverless Application Repository zu gewähren, müssen Sie eine IAM-Entität (Benutzer oder Rolle) für die Person oder Anwendung erstellen, die Zugriff benötigt. Sie werden die Anmeldeinformationen für diese Einrichtung verwenden, um auf AWS zuzugreifen. Anschließend müssen Sie der Entität eine Richtlinie anfügen, die dieser die korrekten Berechtigungen in AWS Serverless Application Repository gewährt.

Informationen zum Einstieg finden Sie unter [Erstellen Ihrer ersten delegierten IAM-Benutzer und -Gruppen](#) im IAM-Benutzerhandbuch.

Ich möchte Personen außerhalb meines AWS-Kontos Zugriff auf meine AWS Serverless Application Repository-Ressourcen gewähren

Sie können eine Rolle erstellen, die Benutzer in anderen Konten oder Personen außerhalb Ihrer Organisation für den Zugriff auf Ihre Ressourcen verwenden können. Sie können festlegen, wem die Übernahme der Rolle anvertraut wird. Im Fall von Services, die ressourcenbasierte Richtlinien oder Zugriffskontrolllisten (Access Control Lists, ACLs) verwenden, können Sie diese Richtlinien verwenden, um Personen Zugriff auf Ihre Ressourcen zu gewähren.

Weitere Informationen dazu finden Sie hier:

- Informationen dazu, ob AWS Serverless Application Repository diese Funktionen unterstützt, finden Sie unter [Wie das mit IAMAWS Serverless Application Repository funktioniert](#).
- Informationen zum Gewähren des Zugriffs auf Ihre Ressourcen für alle Ihre AWS-Konten finden Sie unter [Gewähren des Zugriffs für einen IAM-Benutzer in einem anderen Ihrer AWS-Konto](#) im IAM-Benutzerhandbuch.
- Informationen dazu, wie Sie AWS-Konten-Drittanbieter Zugriff auf Ihre Ressourcen bereitstellen, finden Sie unter [Gewähren des Zugriffs auf AWS-Konten von externen Benutzern](#) im IAM-Benutzerhandbuch.
- Informationen dazu, wie Sie über einen Identitätsverbund Zugriff gewähren, finden Sie unter [Gewähren von Zugriff für extern authentifizierte Benutzer \(Identitätsverbund\)](#) im IAM-Benutzerhandbuch.

- Informationen zum Unterschied zwischen der Verwendung von Rollen und ressourcenbasierten Richtlinien für den kontenübergreifenden Zugriff finden Sie unter [So unterscheiden sich IAM-Rollen von ressourcenbasierten Richtlinien](#) im IAM-Benutzerhandbuch.

Protokollierung und Überwachung im AWS Serverless Application Repository

Die Überwachung ist ein wichtiger Teil der Aufrechterhaltung von Zuverlässigkeit, Verfügbarkeit und Performance Ihrer AWS-Lösungen. Sammeln Sie Überwachungsdaten aller Bestandteile Ihrer AWS-Lösung, damit Sie Ausfälle an mehreren Punkten leichter debuggen können. AWS bietet mehrere Tools für die Überwachung Ihrer AWS Serverless Application Repository-Ressourcen und die Reaktion auf mögliche Vorfälle, wie folgende:

AWS CloudTrail-Protokolle

Die AWS Serverless Application Repository ist in integriert AWS CloudTrail, ein -Service, der die Aktionen eines Benutzers, einer Rolle oder eines AWS Service im AWS Serverless Application Repository aus. CloudTrail erfasst alle API-Aufrufe für AWS Serverless Application Repository als Ereignisse.

Themen

- [Protokollieren von AWS Serverless Application Repository-API-Aufrufen mit AWS CloudTrail](#)

Protokollieren von AWS Serverless Application Repository-API-Aufrufen mit AWS CloudTrail

AWS Serverless Application Repository ist in integriert AWS CloudTrail, ein -Service, der eine Aufzeichnung der von einem Benutzer, einer Rolle oder einem durchgeführten Aktionen bietet AWS Service im AWS Serverless Application Repository aus. CloudTrail erfasst alle API-Aufrufe für den AWS Serverless Application Repository als Ereignisse. Zu den erfassten Aufrufen gehören Aufrufe von der AWS Serverless Application Repository-Konsole und Code-Aufrufe der AWS Serverless Application Repository-API-Operationen.

Wenn Sie einen Trail erstellen, aktivieren Sie die Continuous Delivery von CloudTrail Ereignisse in einem Amazon-S3-Bucket, einschließlich Ereignisse für die AWS Serverless Application

Repositoryaus. Auch wenn Sie keinen Trail konfigurieren, können Sie die neuesten Ereignisse in der CloudTrail-Konsole in Event history (Ereignisverlauf) anzeigen.

Mit den von CloudTrail erfassten Informationen können Sie ermitteln, welche Anforderung an AWS Serverless Application Repository gerichtet wurde. Sie können auch die IP-Adresse, von der die Anforderung ausging, den Ersteller und den Erstellungszeitpunkt sowie weitere Details bestimmen.

Weitere Informationen zu CloudTrail finden Sie im [AWS CloudTrail-Benutzerhandbuch](#).

AWS Serverless Application Repository-Informationen in CloudTrail

CloudTrail wird beim Erstellen Ihres AWS-Kontos für Sie aktiviert. Die in AWS Serverless Application Repository auftretenden Aktivitäten werden als CloudTrail-Ereignis zusammen mit anderen AWS-Serviceereignissen in Event history (Ereignisverlauf) aufgezeichnet. Sie können die neusten Ereignisse in Ihr AWS-Konto downloaden und dort suchen und anzeigen. Weitere Informationen finden Sie unter [Anzeigen von Ereignissen mit dem CloudTrail -API-Ereignisverlauf](#).

Um die Ereignisse in Ihrem AWS Konto einschließlich Ereignissen für AWS Serverless Application Repository kontinuierlich aufzuzeichnen, erstellen Sie einen Trail.

EINWanderwegaktiviertCloudTrailum Protokolldateien an einen Amazon-S3-Bucket bereitzustellen.

Wenn Sie einen Trail in der Konsole anlegen, gilt dieser für alle AWS-Regionen. Der Trail protokolliert Ereignisse aus allen AWS-Regionen in der AWS-Partition und stellt die Protokolldateien in dem Amazon-S3-Bucket bereit, den Sie angeben. Darüber hinaus können Sie andere AWS-Services konfigurieren, um die in den CloudTrail-Protokollen erfassten Ereignisdaten weiter zu analysieren und entsprechend zu agieren. Weitere Informationen finden Sie unter:

- [Übersicht zum Erstellen eines Trails](#)
- [Von CloudTrail unterstützte Services und Integrationen](#)
- [Konfigurieren von Amazon SNS-Benachrichtigungen für CloudTrail](#)
- [Empfangen von CloudTrail-Protokolldateien aus mehreren Regionen](#) und [Empfangen von CloudTrail-Protokolldateien aus mehreren Konten](#)

Alle AWS Serverless Application Repository-Aktionen werden von CloudTrail protokolliert und auf der Seite [AWS Serverless Application Repository-Ressourcen](#) dokumentiert. Zum Beispiel werden durch Aufrufe der Operationen `CreateApplication`, `UpdateApplications` und `ListApplications` Einträge in den CloudTrail-Protokolldateien generiert.

Jeder Ereignis- oder Protokolleintrag enthält Informationen zu dem Benutzer, der die Anforderung generiert hat. Anhand der Identitätsinformationen zur Benutzeridentität können Sie Folgendes bestimmen:

- Ob die Anfrage mit Stammbenutzer- oder AWS Identity and Access Management (IAM)-Anmeldeinformationen ausgeführt wurde.
- Ob die Anforderung mit temporären Sicherheitsanmeldeinformationen für eine Rolle oder einen Verbundbenutzer ausgeführt wurde.
- Ob die Anforderung aus einem anderen AWS-Service gesendet wurde

Weitere Informationen finden Sie unter [CloudTrail userIdentity-Element](#).

Grundlagen von AWS Serverless Application Repository-Protokolldateieinträgen

Ein Trail ist eine Konfiguration, durch die Ereignisse als Protokolldateien an den von Ihnen angegebenen Amazon-S3-Bucket übermittelt werden. CloudTrail-Protokolldateien können einen oder mehrere Einträge enthalten. Ein Ereignis stellt eine einzelne Anfrage aus einer beliebigen Quelle dar und enthält unter anderem Informationen über die angeforderte Aktion, das Datum und die Uhrzeit der Aktion sowie über die Anfrageparameter. CloudTrail-Protokolleinträge sind kein geordnetes Stacktrace der öffentlichen API-Aufrufe und erscheinen daher nicht in einer bestimmten Reihenfolge.

Das folgende Beispiel zeigt einen CloudTrail-Protokolleintrag, der die Aktion `CreateApplication` demonstriert.

```
{
  "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"
}
```

Compliance-Validierung für das AWS Serverless Application Repository

Externe Auditoren bewerten im Rahmen verschiedener AWS-Compliance-Programme die Sicherheit und Compliance von AWS Serverless Application Repository. Hierzu zählen unter anderem SOC, PCI, FedRAMP und andere.

Für eine Liste AWS-Services, die in bestimmten Compliance-Programmen enthalten sind, finden Sie unter [AWS-Dienstleistungen in Scope nach Compliance-Programmen](#) aus. Allgemeine Informationen finden Sie unter [AWS-Compliance-Programme](#).

Die Auditberichte von Drittanbietern lassen sich mit AWS Artifact herunterladen. Weitere Informationen finden Sie unter [Herunterladen von Berichten in AWS-Artifact](#).

Ihre Verantwortung in Bezug auf die Compliance bei der Verwendung von AWS Serverless Application Repository ergibt sich aus der Vertraulichkeit der Daten, den Compliance-Zielen des Unternehmens sowie den einschlägigen Gesetzen und Vorschriften. AWS stellt die folgenden Ressourcen zur Sicherstellung der Compliance bereit:

- [Schnellstartanleitungen für Sicherheit und Compliance](#)— In diesen Bereitstellungsleitfäden finden Sie grundlegende Überlegungen zur Architektur sowie die einzelnen Schritte zur Bereitstellung von Sicherheits- und Compliance-orientierten Basisumgebungen in AWS aus.
- [AWS-Compliance-Ressourcen](#) – Diese Arbeitsbücher und Leitfäden könnten für Ihre Branche und Ihren Standort interessant sein.

- [AWS Config](#) – Dieser AWS-Service bewertet, zu welchem Grad die Konfiguration Ihrer Ressourcen den internen Vorgehensweisen, Branchenrichtlinien und Vorschriften entspricht.
- [AWS Security Hub](#) – Dieser AWS-Service liefert einen umfassenden Überblick über den Sicherheitsstatus in AWS. So können Sie die Compliance mit den Sicherheitsstandards in der Branche und den bewährten Methoden abgleichen.

Ausfallsicherheit im AWS Serverless Application Repository

Im Zentrum der globalen AWS-Infrastruktur stehen die AWS-Regionen und Availability Zones. AWS Regionen stellen mehrere physisch getrennte und isolierte Availability Zones bereit, die mit Netzwerken mit geringer Latenz, hohem Durchsatz und hochredundanten Vernetzungen verbunden sind. Mithilfe von Availability Zones können Sie Anwendungen und Datenbanken erstellen und ausführen, die automatisch Failover zwischen Availability Zones ausführen, ohne dass es zu Unterbrechungen kommt. Availability Zones sind besser hoch verfügbar, fehlertoleranter und skalierbarer als herkömmliche Infrastrukturen mit einem oder mehreren Rechenzentren.

Weitere Informationen über AWS Regionen und Availability Zones finden Sie unter [AWS Globale Infrastruktur](#).

Infrastruktursicherheit im AWS Serverless Application Repository

Als verwalteter Service AWS Serverless Application Repository ist geschützt durch AWS globale Netzwerksicherheit. Informationen zu AWS-Sicherheitsdiensten und wie AWS die Infrastruktur schützt, finden Sie unter [AWS Cloud-Sicherheit](#). Informationen zum Entwerfen Ihrer AWS-Umgebung anhand der bewährten Methoden für die Infrastruktursicherheit finden Sie unter [Infrastrukturschutz](#) im Security Pillar AWS Well-Architected Framework.

Sie verwenden durch AWS veröffentlichte API-Aufrufe, um über das Netzwerk auf AWS Serverless Application Repository zuzugreifen. Kunden müssen Folgendes unterstützen:

- Transport Layer Security (TLS). Wir benötigen TLS 1.2 und empfehlen TLS 1.3.
- Verschlüsselungs-Suiten mit Perfect Forward Secrecy (PFS) wie DHE (Ephemeral Diffie-Hellman) oder ECDHE (Elliptic Curve Ephemeral Diffie-Hellman). Die meisten modernen Systemen wie Java 7 und höher unterstützen diese Modi.

Außerdem müssen Anforderungen mit einer Zugriffsschlüssel-ID und einem geheimen Zugriffsschlüssel signiert sein, der einem IAM-Prinzipal zugeordnet ist. Alternativ können Sie mit [AWS](#)

[Security Token Service](#) (AWS STS) temporäre Sicherheitsanmeldeinformationen erstellen, um die Anforderungen zu signieren.

AWS Serverless Application Repository-Kontingente

Die AWS Serverless Application Repository verfügt über ein Kontingent für die Anzahl der öffentlichen Anwendungen, die ein AWS-Konto in jedem AWS-Region haben kann. Dieses Kontingent gilt pro Region und kann erhöht werden. Um eine Erhöhung anzufordern, verwenden Sie die [Support-Center-Konsole](#).

Ressource	Standardkontingent
Öffentliche Anwendungen (Pro AWS-Konto pro Region)	100

Die folgenden Kontingente gelten für Speicher, der für Codepakete und Anwendungsrichtlinien verfügbar ist. Sie können diese Kontingente nicht ändern.

Ressource	Quota
Kostenloser Amazon-S3-Speicher für Code-Pakete (Pro AWS-Konto pro Region)	5 GB
Länge der Anwendungsrichtlinie	6.144 Zeichen

AWS Serverless Application Repository-Fehlerbehebung

Wenn Sie das AWS Serverless Application Repository verwenden, treten möglicherweise Probleme beim Erstellen, Aktualisieren und Löschen Ihrer Anwendungen auf. In diesem Abschnitt finden Sie Hilfe zur Behebung typischer Probleme. Sie können auch in den [AWS Serverless Application Repository-Foren](#) nach Antworten suchen und Fragen posten.

Note

Anwendungen im AWS Serverless Application Repository werden mit AWS CloudFormation bereitgestellt. Weitere Informationen zur Fehlerbehebung von Problemen mit AWS CloudFormation finden Sie unter [AWS CloudFormation-Fehlerbehebung](#).

Themen

- [Sie können eine Anwendung nicht veröffentlichen](#)
- [Ein Kontingent wurde überschritten](#)
- [Eine aktualisierte Readme-Datei erscheint nicht sofort](#)
- [Sie können eine Anwendung infolge von unzureichenden IAM-Berechtigungen nicht bereitstellen](#)
- [Sie können eine Anwendung nicht zweimal bereitstellen](#)
- [Warum ist meine Anwendung nicht öffentlich verfügbar](#)
- [Den -Support kontaktieren](#)

Sie können eine Anwendung nicht veröffentlichen

Wenn Sie Ihre Anwendung nicht veröffentlichen können, fehlt möglicherweise eine von der Open Source Initiative (OSI) genehmigte Lizenzdatei für Ihre Anwendung.

Zur Veröffentlichung Ihrer Anwendung benötigen Sie eine von OSI genehmigte Lizenzdatei und eine erfolgreich veröffentlichte Version der Anwendung mit einer Quellcode-URL für die Version. Sie können die Lizenz einer Anwendung nach der Erstellung der Anwendung nicht mehr aktualisieren.

Wenn Sie Ihre Anwendung wegen einer fehlenden Lizenzdatei nicht veröffentlichen können, löschen Sie die Anwendung und erstellen Sie eine neue mit dem gleichen Namen. Stellen Sie sicher, dass Sie

mindestens eine Open-Source-Lizenz dafür bereitstellen, die von der Open Source Initiative (OSI)-Organisation genehmigt wurde.

Ein Kontingent wurde überschritten

Wenn eine Fehlermeldung angezeigt wird, die angibt, dass ein Kontingent überschritten wurde, überprüfen Sie, ob Sie ein Ressourcenkontingent erreicht haben. AWS Serverless Application Repository-Kontingente finden Sie unter [AWS Serverless Application Repository-Kontingente](#).

Eine aktualisierte Readme-Datei erscheint nicht sofort

Bei Veröffentlichung Ihrer Anwendung kann die Aktualisierung der Inhalte Ihrer Anwendung bis zu 24 Stunden dauern. Bei Verzögerungen über 24 Stunden wenden Sie sich bitte [AWS Support](#) für Hilfe. Details dazu finden im folgenden Abschnitt.

Sie können eine Anwendung infolge von unzureichenden IAM-Berechtigungen nicht bereitstellen

Um eine AWS Serverless Application Repository-Anwendung bereitstellen zu können, benötigen Sie Berechtigungen für AWS Serverless Application Repository-Ressourcen und AWS CloudFormation-Stacks. Möglicherweise benötigen Sie auch eine Berechtigung für die Verwendung der zugrunde liegenden Services in der Anwendung. Wenn Sie beispielsweise einen Amazon S3-Bucket oder eine Amazon DynamoDB DynamoDB-Tabelle erstellen, brauchen Sie Berechtigungen für Amazon S3 oder DynamoDB.

Wenn sich Ihnen diese Art von Problem stellt, überprüfen Sie Ihre AWS Identity and Access Management (IAM)-Richtlinie und stellen Sie sicher, dass Sie über die erforderlichen Berechtigungen verfügen. Weitere Informationen finden Sie unter [Zugriffskontrolle mit AWS Identity and Access Management](#) aus.

Sie können eine Anwendung nicht zweimal bereitstellen

Der von Ihnen angegebene Anwendungsname wird als Name des AWS CloudFormation-Stacks verwendet. Wenn Sie bei der Bereitstellung einer Anwendung ein Problem haben, vergewissern Sie sich, dass kein AWS CloudFormation-Stack mit dem gleichen Namen vorhanden ist. Falls doch, geben Sie einen anderen Anwendungsnamen an oder löschen Sie den vorhandenen Stack, um die Anwendung erneut mit dem gleichen Namen bereitzustellen.

Warum ist meine Anwendung nicht öffentlich verfügbar

Standardmäßig werden Anwendungen als privat eingestuft. Führen Sie [diese](#) Schritte aus, um Ihre Anwendung öffentlich zu machen.

Den -Support kontaktieren

In einigen Fällen können Sie möglicherweise in diesem Abschnitt oder über die [AWS Serverless Application Repository-Foren](#) keine Lösungen zur Fehlerbehebung finden. Wenn Sie [AWS](#) [Bei Premium-Support](#) können Sie unter einen technischen Supportfall erstellen [AWS-Support](#) aus.

Bevor Sie Kontakt aufnehmen [AWS](#) [Bei Support](#) erhalten Sie zuvor den Amazon-Ressourcennamen (ARN) der Anwendung, zu der Sie Fragen haben. Sie finden den Anwendungs-ARN in der [AWS Serverless Application Repository-Konsole](#).

Operationen

Die REST-API des AWS Serverless Application Repository umfasst die folgenden Vorgänge.

- [CreateApplication](#)

Erstellt eine Anwendung, die optional eine AWSSAM-Datei, um die erste Anwendungsversion im selben Aufruf zu erstellen

- [CreateApplicationVersion](#)

Erstellt eine Anwendungsversion

- [CreateCloudFormationChangeSet](#)

Erzeugt ein AWS CloudFormation Änderungssatz für die gegebene Anwendung

- [CreateCloudFormationTemplate](#)

Erzeugt ein AWS CloudFormation-Vorlage

- [DeleteApplication](#)

Löscht die angegebene Anwendung

- [GetApplication](#)

Ruft die angegebene Anwendung ab

- [GetApplicationPolicy](#)

Ruft die Richtlinie für die Anwendung ab.

- [GetCloudFormationTemplate](#)

Ruft das angegebene ab AWS CloudFormation-Vorlage

- [ListApplicationDependencies](#)

Ruft die Liste der Anwendungen ab, die in der enthaltenen Anwendung verschachtelt sind

- [ListApplications](#)

Listet Anwendungen im Besitz des Auftraggebers auf

- [ListApplicationVersions](#)

Listet die Versionen für die angegebene Anwendung auf

- [PutApplicationPolicy](#)

Legt die Berechtigungsrichtlinie für eine Anwendung fest. Eine Liste der für diesen Vorgang unterstützten Aktionen finden Sie unter [Anwendungsberechtigungen](#) aus.

- [UnshareApplication](#)

Heben die Freigabe einer Bewerbung von einem AWS Organisation

Dieser Vorgang kann nur über das Verwaltungskonto der Organisation aufgerufen werden.

- [UpdateApplication](#)

Aktualisiert die angegebene Anwendung

Ressourcen

Die AWS Serverless Application Repository REST-API umfasst die folgenden Ressourcen.

Themen

- [Applications](#)
- [applicationId](#)
- [Application-ID-Changesets für Anwendungen](#)
- [Applications applicationId Dependencies](#)
- [ApplicationID-Richtlinie für Anwendungen](#)
- [Applications applicationId Templates](#)
- [Applications applicationId Templates templateId](#)
- [Applications applicationId Unshare](#)
- [ApplicationID-Versionen von Anwendungen](#)
- [Anwendungen ApplicationID-Versionen SemanticVersion](#)

Applications

URI

/applications

HTTP-Methoden

GET

Vorgangs-ID: ListApplications

Listet Anwendungen im Besitz des Auftraggebers auf

Abfrageparameter

Name	Typ	Erforderlich	Beschreibung
MaxItems	String	False	Die Gesamtzahl der zurückzusendenden Artikel.
nextToken	String	False	Ein Token, um anzugeben, wo mit dem Paginieren begonnen werden soll.

Antworten

Statuscode	Antwortmodell	Beschreibung
200	ApplicationPage	Herzlichen Glückwunsch
400	BadRequestException	Einer der Parameter der Anfrage ist ungültig.
403	ForbiddenException	Der Client ist nicht authentifiziert.
404	NotFoundException	Die in der Anfrage angegebene Ressource (z. B. eine Anweisung zur Zugriffsrichtlinie) ist nicht vorhanden.
500	InternalServerErrorException	Beim Service AWS Serverless Application Repository trat ein interner Fehler auf.

POST

Vorgangs-ID: CreateApplication

Erstellt eine Anwendung, optional einschließlich einer AWS SAM-Datei, um die erste Anwendungsversion im selben Aufruf zu erstellen.

Antworten

Statuscode	Antwortmodell	Beschreibung
201	<u>Application</u>	Herzlichen Glückwunsch
400	<u>BadRequestException</u>	Einer der Parameter der Anfrage ist ungültig.
403	<u>ForbiddenException</u>	Der Client ist nicht authentifiziert.
409	<u>ConflictException</u>	Die Ressource ist bereits vorhanden.
429	<u>TooManyRequestsException</u>	Der Client sendet mehr als die zulässige Anzahl von Anfragen pro Zeiteinheit.
500	<u>InternalServerErrorException</u>	Beim Service AWS Serverless Application Repository trat ein interner Fehler auf.

OPTIONS

Antworten

Statuscode	Antwortmodell	Beschreibung
200	None	200 Antworten

Schemata

Anfragetexte

POST-Schema

```
{
  "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"
}
```

Antworttexte

ApplicationPage-Schema

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

Application-Schema

```
{
  "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-Schema

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

ForbiddenException-Schema

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

NotFoundException-Schema

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

ConflictException-Schema

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

TooManyRequestsException-Schema

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

InternalServerErrorException-Schema

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

Eigenschaften

Application

Einzelheiten zur Bewerbung.

applicationId

Der Amazon-Ressourcenname (ARN) der -Anwendung

Typ: Zeichenfolge

Erforderlich: True

name

Der Name der Anwendung.

Minimale Länge = 1. Maximale Länge = 140

Pattern: "[a-zA-Z0-9\-\-]+";

Typ: Zeichenfolge

Erforderlich: True

description

Die Beschreibung der Anwendung.

Minimale Länge = 1. Maximale Länge = 256

Typ: Zeichenfolge

Erforderlich: True

author

Der Name des Autors, der die App veröffentlicht.

Minimale Länge = 1. Maximale Länge = 127.

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

Typ: Zeichenfolge

Erforderlich: True

isVerifiedAuthor

Gibt an, ob der Autor dieser Anwendung verifiziert wurde. Das bedeutet, AWS dass das Unternehmen als vernünftiger und umsichtiger Dienstleister die vom Antragsteller bereitgestellten Informationen nach bestem Wissen und Gewissen geprüft und bestätigt hat, dass die Identität des Antragstellers den Angaben entspricht.

Typ: Boolesch

Erforderlich: Falsch

verifiedAuthorUrl

Die URL zum öffentlichen Profil eines verifizierten Autors. Diese URL wurde vom Autor eingereicht.

Typ: Zeichenfolge

Erforderlich: Falsch

spdxLicenseId

Eine gültige Kennung von <https://spdx.org/licenses/>.

Typ: Zeichenfolge

Erforderlich: Falsch

licenseUrl

Ein Link zu einer Lizenzdatei der App, die dem sdxLicenseID-Wert Ihrer Anwendung entspricht.

Maximale Größe 5 MB

Typ: Zeichenfolge

Erforderlich: Falsch

readmeUrl

Ein Link zur Readme-Datei in Markdown-Sprache, die eine detailliertere Beschreibung der Anwendung und ihrer Funktionsweise enthält.

Maximale Größe 5 MB

Typ: Zeichenfolge

Erforderlich: Falsch

labels

Labels zur besseren Auffindbarkeit von Apps in Suchergebnissen.

Minimale Länge = 1. Maximale Länge = 127. Maximale Anzahl von Labels: 10

Pattern: "`^[a-zA-Z0-9+\\-_:\\@]+`";

Typ: Array des Typs string

Erforderlich: Falsch

creationTime

Datum und Uhrzeit der Erstellung dieser Ressource.

Typ: Zeichenfolge

Erforderlich: Falsch

homePageUrl

Eine URL mit weiteren Informationen über die Anwendung, z. B. den Speicherort Ihres GitHub Repositorys für die Anwendung.

Typ: Zeichenfolge

Erforderlich: Falsch

version

Versionsinformationen über die Anwendung.

Typ: [Version](#)

Erforderlich: Falsch

ApplicationPage

Eine Liste mit Anwendungsdetails.

applications

Eine Reihe von Anwendungszusammenfassungen.

Typ: Array des Typs [ApplicationSummary](#)

Erforderlich: True

nextToken

Das Token zum Anfordern der nächsten Ergebnisseite.

Typ: Zeichenfolge

Erforderlich: Falsch

ApplicationSummary

Zusammenfassung der Details zur Anwendung.

applicationId

Der Amazon-Ressourcenname (ARN) der -Anwendung

Typ: Zeichenfolge

Erforderlich: True

name

Der Name der Anwendung.

Minimale Länge = 1. Maximale Länge = 140

Pattern: "[a-zA-Z0-9\\-]+";

Typ: Zeichenfolge

Erforderlich: True

description

Die Beschreibung der Anwendung.

Minimale Länge = 1. Maximale Länge = 256

Typ: Zeichenfolge

Erforderlich: True

author

Der Name des Autors, der die App veröffentlicht.

Minimale Länge = 1. Maximale Länge = 127.

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

Typ: Zeichenfolge

Erforderlich: True

spdxLicenseId

Eine gültige Kennung von <https://spdx.org/licenses/>.

Typ: Zeichenfolge

Erforderlich: Falsch

labels

Labels zur besseren Auffindbarkeit von Apps in Suchergebnissen.

Minimale Länge = 1. Maximale Länge = 127. Maximale Anzahl von Labels: 10

Pattern: "`^[a-zA-Z0-9+\\-_:\\@]+$`";

Typ: Array des Typs string

Erforderlich: Falsch

creationTime

Datum und Uhrzeit der Erstellung dieser Ressource.

Typ: Zeichenfolge

Erforderlich: Falsch

homePageUrl

Eine URL mit weiteren Informationen über die Anwendung, z. B. den Speicherort Ihres GitHub Repositorys für die Anwendung.

Typ: Zeichenfolge

Erforderlich: Falsch

BadRequestException

Einer der Parameter der Anfrage ist ungültig.

message

Einer der Parameter der Anfrage ist ungültig.

Typ: Zeichenfolge

Erforderlich: Falsch

errorCode

400

Typ: Zeichenfolge

Erforderlich: Falsch

Capability

Werte, die angegeben werden müssen, um einige Anwendungen bereitzustellen.

CAPABILITY_IAM
CAPABILITY_NAMED_IAM
CAPABILITY_AUTO_EXPAND
CAPABILITY_RESOURCE_POLICY

ConflictException

Die Ressource ist bereits vorhanden.

message

Die Ressource ist bereits vorhanden.

Typ: Zeichenfolge
Erforderlich: Falsch

errorCode

409

Typ: Zeichenfolge
Erforderlich: Falsch

CreateApplicationInput

Erstellen Sie eine Anwendungsanfrage.

name

Der Name der Anwendung, die Sie veröffentlichen möchten.

Minimale Länge = 1. Maximale Länge = 140

Pattern: "[a-zA-Z0-9\-\-]+";

Typ: Zeichenfolge

Erforderlich: True

description

Die Beschreibung der Anwendung.

Minimale Länge = 1. Maximale Länge = 256

Typ: Zeichenfolge

Erforderlich: True

author

Der Name des Autors, der die App veröffentlicht.

Minimale Länge = 1. Maximale Länge = 127.

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

Typ: Zeichenfolge

Erforderlich: True

spdxLicenseId

Eine gültige Kennung von <https://spdx.org/licenses/>.

Typ: Zeichenfolge

Erforderlich: Falsch

licenseBody

Eine lokale Textdatei, die die Lizenz der App enthält, die dem `spdxLicenseId`-Wert Ihrer Anwendung entspricht. Die Datei hat das Format. `file://<path>/<filename>`

Maximale Größe 5 MB

Sie können nur einen Wert von `licenseBody` und angeben. `licenseUrl` Andernfalls tritt ein Fehler auf.

Typ: Zeichenfolge

Erforderlich: Falsch

licenseUrl

Ein Link zum S3-Objekt, das die Lizenz der App enthält, die dem `spdxLicenseId`-Wert Ihrer Anwendung entspricht.

Maximale Größe 5 MB

Sie können nur einen Wert von `licenseBody` und angeben. `licenseUrl` Andernfalls tritt ein Fehler auf.

Typ: Zeichenfolge

Erforderlich: Falsch

readmeBody

Eine lokale Text-Readme-Datei in Markdown-Sprache, die eine detailliertere Beschreibung der Anwendung und ihrer Funktionsweise enthält. Die Datei hat das Format. `file:///<path>/<filename>`

Maximale Größe 5 MB

Sie können nur einen Wert von `readmeBody` und angeben. `readmeUrl` Andernfalls tritt ein Fehler auf.

Typ: Zeichenfolge

Erforderlich: Falsch

readmeUrl

Ein Link zum S3-Objekt in Markdown-Sprache, der eine detailliertere Beschreibung der Anwendung und ihrer Funktionsweise enthält.

Maximale Größe 5 MB

Sie können nur einen Wert von `readmeBody` und angeben. `readmeUrl` Andernfalls tritt ein Fehler auf.

Typ: Zeichenfolge

Erforderlich: Falsch

labels

Labels zur besseren Auffindbarkeit von Apps in Suchergebnissen.

Minimale Länge = 1. Maximale Länge = 127. Maximale Anzahl von Labels: 10

Pattern: `"^[a-zA-Z0-9+\\-_:\\@]+$"`;

Typ: Array des Typs string

Erforderlich: Falsch

homePageUrl

Eine URL mit weiteren Informationen über die Anwendung, z. B. den Speicherort Ihres GitHub Repositorys für die Anwendung.

Typ: Zeichenfolge

Erforderlich: Falsch

semanticVersion

Die semantische Version der Anwendung:

<https://semver.org/>

Typ: Zeichenfolge

Erforderlich: Falsch

templateBody

Die lokale rohe, verpackte AWS SAM Vorlagendatei Ihrer Anwendung. Die Datei hat das Format `file://<path>/<filename>`.

Sie können nur eines von `templateBody` und `templateUrl` angeben, `templateUrl` andernfalls wird ein Fehler angezeigt.

Typ: Zeichenfolge

Erforderlich: Falsch

templateUrl

Ein Link zu dem S3-Objekt, das die verpackte AWS SAM Vorlage Ihrer Anwendung enthält.

Sie können nur eines von `templateBody` und `templateUrl` angeben. Andernfalls wird ein Fehler angezeigt.

Typ: Zeichenfolge

Erforderlich: Falsch

sourceCodeUrl

Ein Link zu einem öffentlichen Repository für den Quellcode Ihrer Anwendung, z. B. die URL eines bestimmten GitHub Commits.

Typ: Zeichenfolge

Erforderlich: Falsch

sourceCodeArchiveUrl

Ein Link zu dem S3-Objekt, das das ZIP-Archiv des Quellcodes für diese Version Ihrer Anwendung enthält.

Maximale Größe 50 MB

Typ: Zeichenfolge

Erforderlich: Falsch

ForbiddenException

Der Client ist nicht authentifiziert.

message

Der Client ist nicht authentifiziert.

Typ: Zeichenfolge

Erforderlich: Falsch

errorCode

403

Typ: Zeichenfolge

Erforderlich: Falsch

InternalServerErrorException

Beim Service AWS Serverless Application Repository trat ein interner Fehler auf.

message

Beim Service AWS Serverless Application Repository trat ein interner Fehler auf.

Typ: Zeichenfolge

Erforderlich: Falsch

errorCode

500

Typ: Zeichenfolge

Erforderlich: Falsch

NotFoundException

Die in der Anfrage angegebene Ressource (z. B. eine Anweisung zur Zugriffsrichtlinie) ist nicht vorhanden.

message

Die in der Anfrage angegebene Ressource (z. B. eine Erklärung zur Zugriffsrichtlinie) ist nicht vorhanden.

Typ: Zeichenfolge

Erforderlich: Falsch

errorCode

404

Typ: Zeichenfolge

Erforderlich: Falsch

ParameterDefinition

Von der Anwendung unterstützte Parameter.

name

Der Name des Parameters.

Typ: Zeichenfolge

Erforderlich: True

defaultValue

Gibt einen Wert des entsprechenden Typs an, den die Vorlage nutzen soll, wenn zum Zeitpunkt der Stack-Erstellung kein Wert angegeben wird. Wenn Sie Einschränkungen für den Parameter definieren, müssen Sie einen Wert angeben, der diese Einschränkungen einhält.

Typ: Zeichenfolge

Erforderlich: Falsch

description

Eine Zeichenfolge mit bis zu 4.000 Zeichen, die den Parameter beschreibt.

Typ: Zeichenfolge

Erforderlich: Falsch

type

Der Typ des Parameters.

Zulässige Werte: `String` | `Number` | `List<Number>` | `CommaDelimitedList`

`String`: Eine wörtliche Zeichenfolge.

Benutzer können beispielsweise angeben `"MyUserName"`.

`Number`: Eine Ganzzahl oder eine Fließkommazahl. AWS CloudFormation validiert den Parameterwert als Zahl. Wenn Sie den Parameter jedoch an einer anderen Stelle in Ihrer Vorlage verwenden (z. B. mithilfe der `Ref` systeminternen Funktion), wird der Parameterwert zu einer Zeichenfolge.

Benutzer könnten beispielsweise angeben `"8888"`

`List<Number>`: Eine Reihe von Ganzzahlen oder Gleitkommazahlen, die durch Kommas getrennt sind. AWS CloudFormation validiert den Parameterwert als Zahlen. Wenn Sie den Parameter jedoch an einer anderen Stelle in Ihrer Vorlage verwenden (z. B. mithilfe der `Ref` systeminternen Funktion), wird der Parameterwert zu einer Liste von Zeichenfolgen.

Benutzer könnten beispielsweise `„80,20“` angeben und dann das Ergebnis anzeigen. `Ref`
`["80", "20"]`

`CommaDelimitedList`: Ein Array von Literalzeichenfolgen, die durch Kommas getrennt sind. Die Gesamtanzahl von Zeichenfolgen sollte um 1 höher sein als die Gesamtanzahl von Kommas. Außerdem wird jede Elementzeichenfolge durch Leerzeichen gekürzt.

Benutzer könnten beispielsweise `„test, dev, prod“` angeben und dann die Ergebnisse angeben. `Ref`
`["test", "dev", "prod"]`

Typ: Zeichenfolge

Erforderlich: Falsch

`noEcho`

Ob der Parameterwert maskiert werden soll, wenn jemand einen Aufruf tätigt, der den Stack beschreibt. Wenn Sie den Wert auf `true` setzen, wird der Parameterwert mit Sternchen (`*****`) maskiert.

Typ: Boolesch

Erforderlich: Falsch

allowedPattern

Gibt einen regulären Ausdruck an, der die zulässigen Muster für `String`-Typen beschreibt.

Typ: Zeichenfolge

Erforderlich: Falsch

constraintDescription

Gibt eine Zeichenfolge als Beschreibung für eine Einschränkung an. Diese Beschreibung wird bei Verstößen gegen die Einschränkung angezeigt. Für einen Parameter mit einem zulässigen Muster `[A-Za-z0-9]+` beispielsweise wird ohne Einschränkungsbeschreibung die folgende Fehlermeldung angezeigt, wenn der Benutzer einen ungültigen Wert angibt:

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

Durch Hinzufügen einer Beschränkungsbeschreibung, z. B. „darf nur Groß- und Kleinbuchstaben und Zahlen enthalten“, können Sie die folgende benutzerdefinierte Fehlermeldung anzeigen:

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

Typ: Zeichenfolge

Erforderlich: Falsch

minValue

Ein numerischer Wert, der den kleinsten numerischen Wert bestimmt, den Sie für `Number` Typen zulassen möchten.

Typ: Ganzzahl

Erforderlich: Falsch

maxValue

Ein numerischer Wert, der den größten numerischen Wert bestimmt, den Sie für `Number` Typen zulassen möchten.

Typ: Ganzzahl

Erforderlich: Falsch

minLength

Ein ganzzahliger Wert, der die kleinste Anzahl von Zeichen bestimmt, die Sie für `String` Typen zulassen möchten.

Typ: Ganzzahl

Erforderlich: Falsch

maxLength

Ein ganzzahliger Wert, der die größte Anzahl von Zeichen bestimmt, die Sie für `String` Typen zulassen möchten.

Typ: Ganzzahl

Erforderlich: Falsch

allowedValues

Gibt ein Array mit einer Liste der für den Parameter zulässigen Werte an.

Typ: Array des Typs `string`

Erforderlich: Falsch

referencedByResources

Eine Liste von AWS SAM Ressourcen, die diesen Parameter verwenden.

Typ: Array des Typs `string`

Erforderlich: `True`

TooManyRequestsException

Der Client sendet mehr als die zulässige Anzahl von Anfragen pro Zeiteinheit.

message

Der Client sendet mehr als die zulässige Anzahl von Anfragen pro Zeiteinheit.

Typ: Zeichenfolge

Erforderlich: Falsch

errorCode

429

Typ: Zeichenfolge

Erforderlich: Falsch

Version

Einzelheiten zur Anwendungsversion.

applicationId

Der Amazon-Ressourcenname (ARN) der -Anwendung

Typ: Zeichenfolge

Erforderlich: True

semanticVersion

Die semantische Version der Anwendung:

<https://semver.org/>

Typ: Zeichenfolge

Erforderlich: True

sourceCodeUrl

Ein Link zu einem öffentlichen Repository für den Quellcode Ihrer Anwendung, z. B. die URL eines bestimmten GitHub Commits.

Typ: Zeichenfolge

Erforderlich: Falsch

sourceCodeArchiveUrl

Ein Link zu dem S3-Objekt, das das ZIP-Archiv des Quellcodes für diese Version Ihrer Anwendung enthält.

Maximale Größe 50 MB

Typ: Zeichenfolge

Erforderlich: Falsch

templateUrl

Ein Link zur verpackten AWS SAM Vorlage Ihrer Anwendung.

Typ: Zeichenfolge

Erforderlich: True

creationTime

Datum und Uhrzeit der Erstellung dieser Ressource.

Typ: Zeichenfolge

Erforderlich: True

parameterDefinitions

Eine Reihe von Parametertypen, die von der Anwendung unterstützt werden.

Typ: Array des Typs [ParameterDefinition](#)

Erforderlich: True

requiredCapabilities

Eine Liste von Werten, die Sie angeben müssen, bevor Sie bestimmte Anwendungen bereitstellen können. Einige Anwendungen können Ressourcen enthalten, die sich auf die Berechtigungen in Ihrem AWS Konto auswirken können, z. B. indem neue Benutzer AWS Identity and Access Management (IAM) erstellt werden. Für diese Anwendungen müssen Sie ihre Funktionen ausdrücklich bestätigen, indem Sie diesen Parameter angeben.

Die einzig gültigen Werte sind

CAPABILITY_IAM, CAPABILITY_NAMED_IAM, CAPABILITY_RESOURCE_POLICY,
und CAPABILITY_AUTO_EXPAND.

Für die folgenden Ressourcen müssen Sie CAPABILITY_IAM oder

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

und [AWS::IAM::Role](#). Wenn die Anwendung IAM-Ressourcen enthält, können Sie entweder `CAPABILITY_IAM` oder `CAPABILITY_NAMED_IAM` angeben. Beinhaltet die Anwendung IAM-Ressourcen mit benutzerdefinierten Namen, müssen Sie `CAPABILITY_NAMED_IAM` angeben.

Für die folgenden Ressourcen müssen Sie Folgendes angeben `CAPABILITY_RESOURCE_POLICY`: [AWS::Lambda::Permission](#), [AWS::IAM::Policy](#), [AWS::ApplicationAutoScaling::ScalingPolicy](#), [AWS::S3::BucketPolicy](#) und [AWS::SQS::QueuePolicy](#) [AWS::SNS::TopicPolicy](#)

Bei Anwendungen, die eine oder mehrere verschachtelte Anwendungen enthalten, müssen Sie `CAPABILITY_AUTO_EXPAND` angeben.

Wenn Ihre Anwendungsvorlage eine der oben genannten Ressourcen enthält, empfehlen wir Ihnen, vor der Bereitstellung alle mit der Anwendung verbundenen Berechtigungen zu überprüfen. Wenn Sie diesen Parameter nicht für eine Anwendung angeben, die Funktionen benötigt, schlägt der Aufruf fehl.

Typ: Array des Typs [Capability](#)

Erforderlich: True

`resourcesSupported`

Ob alle in dieser Anwendung enthaltenen AWS Ressourcen in der Region unterstützt werden, in der sie abgerufen wird.

Typ: Boolesch

Erforderlich: True

Weitere Informationen finden Sie auch unter

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs sowie Referenzen finden Sie unter:

ListApplications

- [AWS-Befehlszeilenschnittstelle](#)
- [AWS SDK für .NET](#)
- [AWS SDK für C++](#)
- [AWS SDK for Go](#)

- [AWS SDK für Java V2](#)
- [AWS-SDK für JavaScript Version 3](#)
- [AWS SDK für PHP V3](#)
- [AWS SDK für Python](#)
- [AWS SDK für Ruby V3](#)

CreateApplication

- [AWS-Befehlszeilenschnittstelle](#)
- [AWS SDK für .NET](#)
- [AWS SDK für C++](#)
- [AWS SDK for Go](#)
- [AWS SDK für Java V2](#)
- [AWS-SDK für JavaScript Version 3](#)
- [AWS SDK für PHP V3](#)
- [AWS SDK für Python](#)
- [AWS SDK für Ruby V3](#)

applicationId

URI

/applications/*applicationId*

HTTP-Methoden

GET

Vorgangs-ID: GetApplication

Ruft die angegebene Anwendung ab

Pfadparameter

Name	Typ	Erforderlich	Beschreibung
<i>applicationId</i>	String	True	Der Amazon-Ressourcenname (ARN) der Anwendung.

Abfrageparameter

Name	Typ	Erforderlich	Beschreibung
Semantische Version	String	False	Die semantische Version der Anwendung, die abgerufen werden soll.

Antworten

Statuscode	Antwortmodell	Beschreibung
200	Application	Herzlichen Glückwunsch
400	BadRequestException	Einer der Parameter der Anfrage ist ungültig.
403	ForbiddenException	Der Client ist nicht authentifiziert.
404	NotFoundException	Die in der Anfrage angegebene Ressource (z. B. eine Anweisung zur Zugriffsrichtlinie) ist nicht vorhanden.
429	TooManyRequestsException	Der Client sendet mehr als die zulässige Anzahl von Anfragen pro Zeiteinheit.

Statuscode	Antwortmodell	Beschreibung
500	<u>InternalServerErrorException</u>	Beim Service AWS Serverless Application Repository trat ein interner Fehler auf.

DELETE

Vorgangs-ID: DeleteApplication

Löscht die angegebene Anwendung.

Pfadparameter

Name	Typ	Erforderlich	Beschreibung
<i>applicationId</i>	String	True	Der Amazon-Ressourcenname (ARN) der Anwendung.

Antworten

Statuscode	Antwortmodell	Beschreibung
204	None	Herzlichen Glückwunsch
400	<u>BadRequestException</u>	Einer der Parameter der Anfrage ist ungültig.
403	<u>ForbiddenException</u>	Der Client ist nicht authentifiziert.
404	<u>NotFoundException</u>	Die in der Anfrage angegebene Ressource (z. B. eine Anweisung zur Zugriffsrichtlinie) ist nicht vorhanden.
409	<u>ConflictException</u>	Die Ressource ist bereits vorhanden.

Statuscode	Antwortmodell	Beschreibung
429	<u>TooManyRequestsException</u>	Der Client sendet mehr als die zulässige Anzahl von Anfragen pro Zeiteinheit.
500	<u>InternalServerErrorException</u>	Beim Service AWS Serverless Application Repository trat ein interner Fehler auf.

OPTIONS

Pfadparameter

Name	Typ	Erforderlich	Beschreibung
<i>applicationId</i>	String	True	Der Amazon-Ressourcenname (ARN) der Anwendung.

Antworten

Statuscode	Antwortmodell	Beschreibung
200	None	Antwort von 200

PATCH

Vorgangs-ID: UpdateApplication

Aktualisiert die angegebene Anwendung.

Pfadparameter

Name	Typ	Erforderlich	Beschreibung
<i>applicationId</i>	String	True	Der Amazon-Ressourcenname (ARN) der Anwendung.

Antworten

Statuscode	Antwortmodell	Beschreibung
200	Application	Herzlichen Glückwunsch
400	BadRequestException	Einer der Parameter der Anfrage ist ungültig.
403	ForbiddenException	Der Client ist nicht authentifiziert.
404	NotFoundException	Die in der Anfrage angegebene Ressource (z. B. eine Anweisung zur Zugriffsrichtlinie) ist nicht vorhanden.
409	ConflictException	Die Ressource ist bereits vorhanden.
429	TooManyRequestsException	Der Client sendet mehr als die zulässige Anzahl von Anfragen pro Zeiteinheit.
500	InternalServerErrorException	Beim Service AWS Serverless Application Repository trat ein interner Fehler auf.

Schemata

Anfragetexte

PATCH-Schema

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

Antworttexte

Application-Schema

```
{
  "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-Schema

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

ForbiddenException-Schema

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

NotFoundException-Schema

```
{
```

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

ConflictException-Schema

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

TooManyRequestsException-Schema

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

InternalServerErrorException-Schema

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

Eigenschaften

Application

Einzelheiten zur Anwendung.

applicationId

Der Amazon-Ressourcenname (ARN) der -Anwendung

Typ: Zeichenfolge

Erforderlich: True

name

Der Name der Anwendung.

Minimale Länge = 1. Maximale Länge = 140

Pattern: "[a-zA-Z0-9\-\-]+";

Typ: Zeichenfolge

Erforderlich: True

description

Die Beschreibung der Anwendung.

Minimale Länge = 1. Maximale Länge = 256

Typ: Zeichenfolge

Erforderlich: True

author

Der Name des Autors, der die App veröffentlicht.

Minimale Länge = 1. Maximale Länge = 127.

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

Typ: Zeichenfolge

Erforderlich: True

isVerifiedAuthor

Gibt an, ob der Autor dieser Anwendung verifiziert wurde. Das bedeutet, AWS dass das Unternehmen als vernünftiger und umsichtiger Dienstleister die vom Antragsteller bereitgestellten Informationen nach bestem Wissen und Gewissen geprüft und bestätigt hat, dass die Identität des Antragstellers den Angaben entspricht.

Typ: Boolesch

Erforderlich: Falsch

verifiedAuthorUrl

Die URL zum öffentlichen Profil eines verifizierten Autors. Diese URL wurde vom Autor eingereicht.

Typ: Zeichenfolge

Erforderlich: Falsch

spdxLicenseId

Eine gültige Kennung von <https://spdx.org/licenses/>.

Typ: Zeichenfolge

Erforderlich: Falsch

licenseUrl

Ein Link zu einer Lizenzdatei der App, die dem spdxLicenseID-Wert Ihrer Anwendung entspricht.

Maximale Größe 5 MB

Typ: Zeichenfolge

Erforderlich: Falsch

readmeUrl

Ein Link zur Readme-Datei in Markdown-Sprache, die eine detailliertere Beschreibung der Anwendung und ihrer Funktionsweise enthält.

Maximale Größe 5 MB

Typ: Zeichenfolge

Erforderlich: Falsch

labels

Labels zur besseren Auffindbarkeit von Apps in Suchergebnissen.

Minimale Länge = 1. Maximale Länge = 127. Maximale Anzahl von Labels: 10

Pattern: `"^[a-zA-Z0-9+\\-_:\\V@]+$"`;

Typ: Array des Typs string

Erforderlich: Falsch

creationTime

Datum und Uhrzeit der Erstellung dieser Ressource.

Typ: Zeichenfolge

Erforderlich: Falsch

homePageUrl

Eine URL mit weiteren Informationen über die Anwendung, z. B. den Speicherort Ihres GitHub Repositorys für die Anwendung.

Typ: Zeichenfolge

Erforderlich: Falsch

version

Versionsinformationen über die Anwendung.

Typ: [Version](#)

Erforderlich: Falsch

BadRequestException

Einer der Parameter der Anfrage ist ungültig.

message

Einer der Parameter der Anfrage ist ungültig.

Typ: Zeichenfolge

Erforderlich: Falsch

errorCode

400

Typ: Zeichenfolge
Erforderlich: Falsch

Capability

Werte, die angegeben werden müssen, um einige Anwendungen bereitzustellen.

CAPABILITY_IAM
CAPABILITY_NAMED_IAM
CAPABILITY_AUTO_EXPAND
CAPABILITY_RESOURCE_POLICY

ConflictException

Die Ressource ist bereits vorhanden.

message

Die Ressource ist bereits vorhanden.

Typ: Zeichenfolge
Erforderlich: Falsch

errorCode

409

Typ: Zeichenfolge
Erforderlich: Falsch

ForbiddenException

Der Client ist nicht authentifiziert.

message

Der Client ist nicht authentifiziert.

Typ: Zeichenfolge

Erforderlich: Falsch

errorCode

403

Typ: Zeichenfolge

Erforderlich: Falsch

InternalServerErrorException

Beim Service AWS Serverless Application Repository trat ein interner Fehler auf.

message

Beim Service AWS Serverless Application Repository trat ein interner Fehler auf.

Typ: Zeichenfolge

Erforderlich: Falsch

errorCode

500

Typ: Zeichenfolge

Erforderlich: Falsch

NotFoundException

Die in der Anfrage angegebene Ressource (z. B. eine Anweisung zur Zugriffsrichtlinie) ist nicht vorhanden.

message

Die in der Anfrage angegebene Ressource (z. B. eine Erklärung zur Zugriffsrichtlinie) ist nicht vorhanden.

Typ: Zeichenfolge

Erforderlich: Falsch

errorCode

404

Typ: Zeichenfolge

Erforderlich: Falsch

ParameterDefinition

Von der Anwendung unterstützte Parameter.

name

Der Name des Parameters.

Typ: Zeichenfolge

Erforderlich: True

defaultValue

Gibt einen Wert des entsprechenden Typs an, den die Vorlage nutzen soll, wenn zum Zeitpunkt der Stack-Erstellung kein Wert angegeben wird. Wenn Sie Einschränkungen für den Parameter definieren, müssen Sie einen Wert angeben, der diese Einschränkungen einhält.

Typ: Zeichenfolge

Erforderlich: Falsch

description

Eine Zeichenfolge mit bis zu 4.000 Zeichen, die den Parameter beschreibt.

Typ: Zeichenfolge

Erforderlich: Falsch

type

Der Typ des Parameters.

Zulässige Werte: `String` | `Number` | `List<Number>` | `CommaDelimitedList`

`String`: Eine wörtliche Zeichenfolge.

Benutzer können beispielsweise angeben `"MyUserName"`.

`Number`: Eine Ganzzahl oder eine Fließkommazahl. AWS CloudFormation validiert den Parameterwert als Zahl. Wenn Sie den Parameter jedoch an einer anderen Stelle in Ihrer Vorlage verwenden (z. B. mithilfe der `Ref` systeminternen Funktion), wird der Parameterwert zu einer Zeichenfolge.

Benutzer könnten beispielsweise angeben `"8888"`

`List<Number>`: Eine Reihe von Ganzzahlen oder Gleitkommazahlen, die durch Kommas getrennt sind. AWS CloudFormation validiert den Parameterwert als Zahlen. Wenn Sie den Parameter jedoch an einer anderen Stelle in Ihrer Vorlage verwenden (z. B. mithilfe der `Ref` systeminternen Funktion), wird der Parameterwert zu einer Liste von Zeichenfolgen.

Benutzer könnten beispielsweise `„80,20“` angeben und dann das Ergebnis anzeigen. `Ref` `["80", "20"]`

`CommaDelimitedList`: Ein Array von Literalzeichenfolgen, die durch Kommas getrennt sind. Die Gesamtanzahl von Zeichenfolgen sollte um 1 höher sein als die Gesamtanzahl von Kommas. Außerdem wird jede Elementzeichenfolge durch Leerzeichen gekürzt.

Benutzer könnten beispielsweise `„test, dev, prod“` angeben und dann die Ergebnisse angeben. `Ref` `["test", "dev", "prod"]`

Typ: Zeichenfolge

Erforderlich: Falsch

`noEcho`

Ob der Parameterwert maskiert werden soll, wenn jemand einen Aufruf tätigt, der den Stack beschreibt. Wenn Sie den Wert auf `true` setzen, wird der Parameterwert mit Sternchen (`*****`) maskiert.

Typ: Boolesch

Erforderlich: Falsch

allowedPattern

Gibt einen regulären Ausdruck an, der die zulässigen Muster für `String`-Typen beschreibt.

Typ: Zeichenfolge

Erforderlich: Falsch

constraintDescription

Gibt eine Zeichenfolge als Beschreibung für eine Einschränkung an. Diese Beschreibung wird bei Verstößen gegen die Einschränkung angezeigt. Für einen Parameter mit einem zulässigen Muster `[A-Za-z0-9]+` beispielsweise wird ohne Einschränkungsbeschreibung die folgende Fehlermeldung angezeigt, wenn der Benutzer einen ungültigen Wert angibt:

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

Durch Hinzufügen einer Beschränkungsbeschreibung, z. B. „darf nur Groß- und Kleinbuchstaben und Zahlen enthalten“, können Sie die folgende benutzerdefinierte Fehlermeldung anzeigen:

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

Typ: Zeichenfolge

Erforderlich: Falsch

minValue

Ein numerischer Wert, der den kleinsten numerischen Wert bestimmt, den Sie für `Number` Typen zulassen möchten.

Typ: Ganzzahl

Erforderlich: Falsch

maxValue

Ein numerischer Wert, der den größten numerischen Wert bestimmt, den Sie für `Number` Typen zulassen möchten.

Typ: Ganzzahl

Erforderlich: Falsch

minLength

Ein ganzzahliger Wert, der die kleinste Anzahl von Zeichen bestimmt, die Sie für `String` Typen zulassen möchten.

Typ: Ganzzahl

Erforderlich: Falsch

maxLength

Ein ganzzahliger Wert, der die größte Anzahl von Zeichen bestimmt, die Sie für `String` Typen zulassen möchten.

Typ: Ganzzahl

Erforderlich: Falsch

allowedValues

Gibt ein Array mit einer Liste der für den Parameter zulässigen Werte an.

Typ: Array des Typs `string`

Erforderlich: Falsch

referencedByResources

Eine Liste von AWS SAM Ressourcen, die diesen Parameter verwenden.

Typ: Array des Typs `string`

Erforderlich: `True`

TooManyRequestsException

Der Client sendet mehr als die zulässige Anzahl von Anfragen pro Zeiteinheit.

message

Der Client sendet mehr als die zulässige Anzahl von Anfragen pro Zeiteinheit.

Typ: Zeichenfolge

Erforderlich: Falsch

errorCode

429

Typ: Zeichenfolge

Erforderlich: Falsch

UpdateApplicationInput

Aktualisieren Sie die Anwendungsanfrage.

description

Die Beschreibung der Anwendung.

Minimale Länge = 1. Maximale Länge = 256

Typ: Zeichenfolge

Erforderlich: Falsch

author

Der Name des Autors, der die App veröffentlicht.

Minimale Länge = 1. Maximale Länge = 127.

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

Typ: Zeichenfolge

Erforderlich: Falsch

readmeBody

Eine Text-Readme-Datei in Markdown-Sprache, die eine detailliertere Beschreibung der Anwendung und ihrer Funktionsweise enthält.

Maximale Größe 5 MB

Typ: Zeichenfolge

Erforderlich: Falsch

readmeUrl

Ein Link zur Readme-Datei in Markdown-Sprache, die eine detailliertere Beschreibung der Anwendung und ihrer Funktionsweise enthält.

Maximale Größe 5 MB

Typ: Zeichenfolge

Erforderlich: Falsch

labels

Labels zur besseren Auffindbarkeit von Apps in Suchergebnissen.

Minimale Länge = 1. Maximale Länge = 127. Maximale Anzahl von Labels: 10

Pattern: `^[a-zA-Z0-9+\\-_:\\@]+$`;

Typ: Array des Typs string

Erforderlich: Falsch

homePageUrl

Eine URL mit weiteren Informationen über die Anwendung, z. B. den Speicherort Ihres GitHub Repositorys für die Anwendung.

Typ: Zeichenfolge

Erforderlich: Falsch

Version

Einzelheiten zur Anwendungsversion.

applicationId

Der Amazon-Ressourcenname (ARN) der -Anwendung

Typ: Zeichenfolge

Erforderlich: True

semanticVersion

Die semantische Version der Anwendung:

<https://semver.org/>

Typ: Zeichenfolge

Erforderlich: True

sourceCodeUrl

Ein Link zu einem öffentlichen Repository für den Quellcode Ihrer Anwendung, z. B. die URL eines bestimmten GitHub Commits.

Typ: Zeichenfolge

Erforderlich: Falsch

sourceCodeArchiveUrl

Ein Link zu dem S3-Objekt, das das ZIP-Archiv des Quellcodes für diese Version Ihrer Anwendung enthält.

Maximale Größe 50 MB

Typ: Zeichenfolge

Erforderlich: Falsch

templateUrl

Ein Link zur verpackten AWS SAM Vorlage Ihrer Anwendung.

Typ: Zeichenfolge

Erforderlich: True

creationTime

Datum und Uhrzeit der Erstellung dieser Ressource.

Typ: Zeichenfolge

Erforderlich: True

parameterDefinitions

Eine Reihe von Parametertypen, die von der Anwendung unterstützt werden.

Typ: Array des Typs [ParameterDefinition](#)

Erforderlich: True

requiredCapabilities

Eine Liste von Werten, die Sie angeben müssen, bevor Sie bestimmte Anwendungen bereitstellen können. Einige Anwendungen können Ressourcen enthalten, die sich auf die Berechtigungen in Ihrem AWS Konto auswirken können, z. B. indem neue Benutzer AWS Identity and Access Management (IAM) erstellt werden. Für diese Anwendungen müssen Sie ihre Funktionen ausdrücklich bestätigen, indem Sie diesen Parameter angeben.

Die einzig gültigen Werte sind

CAPABILITY_IAM, CAPABILITY_NAMED_IAM, CAPABILITY_RESOURCE_POLICY,
und CAPABILITY_AUTO_EXPAND.

Für die folgenden Ressourcen müssen Sie CAPABILITY_IAM oder angeben CAPABILITY_NAMED_IAM: [AWS::IAM::Group](#), [AWS::IAM::InstanceProfile](#), [AWS::IAM::Policy](#), und [AWS::IAM::Role](#). Wenn die Anwendung IAM-Ressourcen enthält, können Sie entweder CAPABILITY_IAM oder CAPABILITY_NAMED_IAM angeben. Beinhaltet die Anwendung IAM-Ressourcen mit benutzerdefinierten Namen, müssen Sie CAPABILITY_NAMED_IAM angeben.

Für die folgenden Ressourcen müssen Sie Folgendes angeben CAPABILITY_RESOURCE_POLICY: [AWS::Lambda::Permission](#), [AWS::IAM::Policy](#), [AWS::ApplicationAutoScaling::ScalingPolicy](#), [AWS::S3::BucketPolicy](#) und [AWS::SQS::QueuePolicy](#), [AWS::SNS::TopicPolicy](#)

Bei Anwendungen, die eine oder mehrere verschachtelte Anwendungen enthalten, müssen Sie CAPABILITY_AUTO_EXPAND angeben.

Wenn Ihre Anwendungsvorlage eine der oben genannten Ressourcen enthält, empfehlen wir Ihnen, vor der Bereitstellung alle mit der Anwendung verbundenen Berechtigungen zu überprüfen. Wenn Sie diesen Parameter nicht für eine Anwendung angeben, die Funktionen benötigt, schlägt der Aufruf fehl.

Typ: Array des Typs [Capability](#)

Erforderlich: True

resourcesSupported

Ob alle in dieser Anwendung enthaltenen AWS Ressourcen in der Region unterstützt werden, in der sie abgerufen wird.

Typ: Boolesch

Erforderlich: True

Weitere Informationen finden Sie auch unter

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs sowie Referenzen finden Sie unter:

GetApplication

- [AWS-Befehlszeilenschnittstelle](#)
- [AWS SDK für .NET](#)
- [AWS SDK für C++](#)
- [AWS SDK for Go](#)
- [AWS SDK für Java V2](#)
- [AWS-SDK für JavaScript Version 3](#)
- [AWS SDK für PHP V3](#)
- [AWS SDK für Python](#)
- [AWS SDK für Ruby V3](#)

DeleteApplication

- [AWS-Befehlszeilenschnittstelle](#)
- [AWS SDK für .NET](#)
- [AWS SDK für C++](#)
- [AWS SDK for Go](#)
- [AWS SDK für Java V2](#)
- [AWS-SDK für JavaScript Version 3](#)
- [AWS SDK für PHP V3](#)

- [AWS SDK für Python](#)
- [AWS SDK für Ruby V3](#)

UpdateApplication

- [AWS-Befehlszeilenschnittstelle](#)
- [AWS SDK für .NET](#)
- [AWS SDK für C++](#)
- [AWS SDK for Go](#)
- [AWS SDK für Java V2](#)
- [AWS-SDK für JavaScript Version 3](#)
- [AWS SDK für PHP V3](#)
- [AWS SDK für Python](#)
- [AWS SDK für Ruby V3](#)

Application-ID-Changesets für Anwendungen

URI

/applications/*applicationId*/changesets

HTTP-Methoden

POST

Vorgangs-ID: CreateCloudFormationChangeSet

Erstellt einen AWS CloudFormation Änderungssatz für die angegebene Anwendung.

Pfadparameter

Name	Typ	Erforderlich	Beschreibung
<i>applicationId</i>	String	True	Der Amazon-Ressourcenname (ARN) der Anwendung.

Antworten

Statuscode	Antwortmodell	Beschreibung
201	<u>ChangeSetDetails</u>	Herzlichen Glückwunsch
400	<u>BadRequestException</u>	Einer der Parameter der Anfrage ist ungültig.
403	<u>ForbiddenException</u>	Der Client ist nicht authentifiziert.
429	<u>TooManyRequestsException</u>	Der Client sendet mehr als die zulässige Anzahl von Anfragen pro Zeiteinheit.
500	<u>InternalServerErrorException</u>	Beim Service AWS Serverless Application Repository trat ein interner Fehler auf.

OPTIONS

Pfadparameter

Name	Typ	Erforderlich	Beschreibung
<i>applicationId</i>	String	True	Der Amazon-Ressourcenname (ARN) der Anwendung.

Antworten

Statuscode	Antwortmodell	Beschreibung
200	None	Antwort von 200

Schemata

Anfragetexte

POST-Schema

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

```
}
```

Antworttexte

ChangeSetDetails-Schema

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

BadRequestException-Schema

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

ForbiddenException-Schema

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

TooManyRequestsException-Schema

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

InternalServerErrorException-Schema

```
{
```

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

Eigenschaften

BadRequestException

Einer der Parameter der Anfrage ist ungültig.

message

Einer der Parameter der Anfrage ist ungültig.

Typ: Zeichenfolge

Erforderlich: Falsch

errorCode

400

Typ: Zeichenfolge

Erforderlich: Falsch

ChangeSetDetails

Einzelheiten des Änderungssatzes.

applicationId

Der Amazon-Ressourcenname (ARN) der -Anwendung

Typ: Zeichenfolge

Erforderlich: True

semanticVersion

Die semantische Version der Anwendung:

<https://semver.org/>

Typ: Zeichenfolge

Erforderlich: True

changeSetId

Der Amazon-Ressourcenname (ARN) des Änderungssatzes.

Längenbeschränkungen: Minimale Länge von 1.

Muster: ARN: [-a-Za-Z0-9:/] *

Typ: Zeichenfolge

Erforderlich: True

stackId

Die eindeutige ID des Stacks.

Typ: Zeichenfolge

Erforderlich: True

CreateCloudFormationChangeSetInput

Erstellen Sie eine Anfrage für ein Anwendungs-Änderungssatz.

stackName

Diese Eigenschaft entspricht dem AWS CloudFormation [CreateChangeSet](#) gleichnamigen Parameter für die API.

Typ: Zeichenfolge

Erforderlich: True

semanticVersion

Die semantische Version der Anwendung:

<https://semver.org/>

Typ: Zeichenfolge

Erforderlich: Falsch

templateId

Die von zurückgegebene UUID. CreateCloudFormationTemplate

Muster: [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}

Typ: Zeichenfolge

Erforderlich: Falsch

parameterOverrides

Eine Liste von Parameterwerten für die Parameter der Anwendung.

Typ: Array des Typs [ParameterValue](#)

Erforderlich: Falsch

capabilities

Eine Liste von Werten, die Sie angeben müssen, bevor Sie bestimmte Anwendungen bereitstellen können. Einige Anwendungen können Ressourcen enthalten, die sich auf die Berechtigungen in Ihrem AWS Konto auswirken können, z. B. indem neue Benutzer AWS Identity and Access Management (IAM) erstellt werden. Für diese Anwendungen müssen Sie ihre Funktionen ausdrücklich bestätigen, indem Sie diesen Parameter angeben.

Die einzig gültigen Werte sind

CAPABILITY_IAM, CAPABILITY_NAMED_IAM, CAPABILITY_RESOURCE_POLICY,
und CAPABILITY_AUTO_EXPAND.

Für die folgenden Ressourcen müssen Sie CAPABILITY_IAM oder angeben CAPABILITY_NAMED_IAM: [AWS::IAM::Group](#), [AWS::IAM::InstanceProfile](#), [AWS::IAM::Policy](#), und [AWS::IAM::Role](#). Wenn die Anwendung IAM-Ressourcen enthält, können Sie entweder CAPABILITY_IAM oder CAPABILITY_NAMED_IAM angeben. Beinhaltet die Anwendung IAM-Ressourcen mit benutzerdefinierten Namen, müssen Sie CAPABILITY_NAMED_IAM angeben.

Für die folgenden Ressourcen müssen Sie `CAPABILITY_RESOURCE_POLICY`:

[AWS::Lambda::Permission](#), [AWS::IAM::Policy](#),
[AWS::ApplicationAutoScaling::ScalingPolicy](#), [AWS::S3::BucketPolicy](#), [AWS::SQS::QueuePolicy](#), und
[AWS::SNS::TopicPolicy](#) angeben.

Bei Anwendungen, die eine oder mehrere verschachtelte Anwendungen enthalten, müssen Sie `CAPABILITY_AUTO_EXPAND` angeben.

Wenn Ihre Anwendungsvorlage eine der oben genannten Ressourcen enthält, empfehlen wir Ihnen, vor der Bereitstellung alle mit der Anwendung verbundenen Berechtigungen zu überprüfen. Wenn Sie diesen Parameter nicht für eine Anwendung angeben, die Funktionen benötigt, schlägt der Aufruf fehl.

Typ: Array des Typs string

Erforderlich: Falsch

changeSetName

Diese Eigenschaft entspricht dem gleichnamigen Parameter für die AWS CloudFormation [CreateChangeSet](#) API.

Typ: Zeichenfolge

Erforderlich: Falsch

clientToken

Diese Eigenschaft entspricht dem gleichnamigen Parameter für die AWS CloudFormation [CreateChangeSet](#) API.

Typ: Zeichenfolge

Erforderlich: Falsch

description

Diese Eigenschaft entspricht dem gleichnamigen Parameter für die AWS CloudFormation [CreateChangeSet](#) API.

Typ: Zeichenfolge

Erforderlich: Falsch

notificationArns

Diese Eigenschaft entspricht dem gleichnamigen Parameter für die AWS CloudFormation [CreateChangeSet](#)API.

Typ: Array des Typs string

Erforderlich: Falsch

resourceTypes

Diese Eigenschaft entspricht dem gleichnamigen Parameter für die AWS CloudFormation [CreateChangeSet](#)API.

Typ: Array des Typs string

Erforderlich: Falsch

rollbackConfiguration

Diese Eigenschaft entspricht dem gleichnamigen Parameter für die AWS CloudFormation [CreateChangeSet](#)API.

Typ: [RollbackConfiguration](#)

Erforderlich: Falsch

tags

Diese Eigenschaft entspricht dem gleichnamigen Parameter für die AWS CloudFormation [CreateChangeSet](#)API.

Typ: Array des Typs [Tag](#)

Erforderlich: Falsch

ForbiddenException

Der Client ist nicht authentifiziert.

message

Der Client ist nicht authentifiziert.

Typ: Zeichenfolge

Erforderlich: Falsch

errorCode

403

Typ: Zeichenfolge

Erforderlich: Falsch

InternalServerErrorException

Beim Service AWS Serverless Application Repository trat ein interner Fehler auf.

message

Beim Service AWS Serverless Application Repository trat ein interner Fehler auf.

Typ: Zeichenfolge

Erforderlich: Falsch

errorCode

500

Typ: Zeichenfolge

Erforderlich: Falsch

ParameterValue

Parameterwert der Anwendung.

name

Der Schlüssel, der dem Parameter zugeordnet ist. Wenn Sie keinen Schlüssel und Wert für einen bestimmten Parameter angeben, AWS CloudFormation verwendet es den Standardwert, der in Ihrer Vorlage angegeben ist.

Typ: Zeichenfolge

Erforderlich: True

value

Der dem Parameter zugeordnete Eingabewert.

Typ: Zeichenfolge

Erforderlich: True

RollbackConfiguration

Diese Eigenschaft entspricht dem AWS CloudFormation [RollbackConfiguration](#) Datentyp.

rollbackTriggers

Diese Eigenschaft entspricht dem gleichnamigen Inhalt für den AWS CloudFormation [RollbackConfiguration](#) Datentyp.

Typ: Array des Typs [RollbackTrigger](#)

Erforderlich: Falsch

monitoringTimeInMinutes

Diese Eigenschaft entspricht dem gleichnamigen Inhalt für den AWS CloudFormation [RollbackConfiguration](#) Datentyp.

Typ: Ganzzahl

Erforderlich: Falsch

RollbackTrigger

Diese Eigenschaft entspricht dem AWS CloudFormation [RollbackTrigger](#) Datentyp.

arn

Diese Eigenschaft entspricht dem gleichnamigen Inhalt für den AWS CloudFormation [RollbackTrigger](#) Datentyp.

Typ: Zeichenfolge

Erforderlich: True

type

Diese Eigenschaft entspricht dem gleichnamigen Inhalt für den AWS CloudFormation [RollbackTrigger](#) Datentyp.

Typ: Zeichenfolge

Erforderlich: True

Tag

Diese Eigenschaft entspricht dem AWS CloudFormation [Tag](#)-Datentyp.

key

Diese Eigenschaft entspricht dem gleichnamigen Inhalt für den AWS CloudFormation [Tag](#)-Datentyp.

Typ: Zeichenfolge

Erforderlich: True

value

Diese Eigenschaft entspricht dem gleichnamigen Inhalt für den AWS CloudFormation [Tag](#)-Datentyp.

Typ: Zeichenfolge

Erforderlich: True

TooManyRequestsException

Der Client sendet mehr als die zulässige Anzahl von Anfragen pro Zeiteinheit.

message

Der Client sendet mehr als die zulässige Anzahl von Anfragen pro Zeiteinheit.

Typ: Zeichenfolge

Erforderlich: Falsch

errorCode

429

Typ: Zeichenfolge

Erforderlich: Falsch

Weitere Informationen finden Sie auch unter

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs sowie Referenzen finden Sie unter:

CreateCloudFormationChangeSet

- [AWS-Befehlszeilenschnittstelle](#)
- [AWS SDK für .NET](#)
- [AWS SDK für C++](#)
- [AWS SDK for Go](#)
- [AWS SDK für Java V2](#)
- [AWS-SDK für JavaScript Version 3](#)
- [AWS SDK für PHP V3](#)
- [AWS SDK für Python](#)
- [AWS SDK für Ruby V3](#)

Applications applicationId Dependencies

URI

/applications/*applicationId*/dependencies

HTTP-Methoden

GET

Vorgangs-ID: ListApplicationDependencies

Ruft die Liste der Anwendungen ab, die in der enthaltenen Anwendung verschachtelt sind.

Pfadparameter

Name	Typ	Erforderlich	Beschreibung
<i>applicationId</i>	String	True	Der Amazon-Ressourcenname (ARN) der Anwendung.

Abfrageparameter

Name	Typ	Erforderlich	Beschreibung
nextToken	String	False	Ein Token, um anzugeben, wo mit der Paginierung begonnen werden soll.
MaxItems	String	False	Die Gesamtzahl der zurückzusendenden Artikel.
Semantische Version	String	False	Die semantische Version der Anwendung, die abgerufen werden soll.

Antworten

Statuscode	Antwortmodell	Beschreibung
200	ApplicationDependencyPage	Herzlichen Glückwunsch
400	BadRequestException	Einer der Parameter der Anfrage ist ungültig.

Statuscode	Antwortmodell	Beschreibung
403	ForbiddenException	Der Client ist nicht authentifiziert.
404	NotFoundException	Die in der Anfrage angegebene Ressource (z. B. eine Anweisung zur Zugriffsrichtlinie) ist nicht vorhanden.
429	TooManyRequestsException	Der Client sendet mehr als die zulässige Anzahl von Anfragen pro Zeiteinheit.
500	InternalServerErrorException	Beim Service AWS Serverless Application Repository trat ein interner Fehler auf.

OPTIONS

Pfadparameter

Name	Typ	Erforderlich	Beschreibung
<i>applicationId</i>	String	True	Der Amazon-Ressourcenname (ARN) der Anwendung.

Antworten

Statuscode	Antwortmodell	Beschreibung
200	None	Antwort von 200

Schemata

Antworttexte

ApplicationDependencyPage-Schema

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

BadRequestException-Schema

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

ForbiddenException-Schema

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

NotFoundException-Schema

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

TooManyRequestsException-Schema

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

```
}
```

InternalServerErrorException-Schema

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

Eigenschaften

ApplicationDependencyPage

Eine Liste von Anwendungszusammenfassungen, die in der Anwendung verschachtelt sind.

dependencies

Eine Reihe von Anwendungszusammenfassungen, die in der Anwendung verschachtelt sind.

Typ: Array des Typs [ApplicationDependencySummary](#)

Erforderlich: True

nextToken

Das Token zum Anfordern der nächsten Ergebnisseite.

Typ: Zeichenfolge

Erforderlich: Falsch

ApplicationDependencySummary

Eine Zusammenfassung der verschachtelten Anwendung.

applicationId

Der Amazon-Ressourcenname (ARN) der verschachtelten Anwendung.

Typ: Zeichenfolge

Erforderlich: True

semanticVersion

Die semantische Version der verschachtelten Anwendung.

Typ: Zeichenfolge

Erforderlich: True

BadRequestException

Einer der Parameter der Anfrage ist ungültig.

message

Einer der Parameter der Anfrage ist ungültig.

Typ: Zeichenfolge

Erforderlich: Falsch

errorCode

400

Typ: Zeichenfolge

Erforderlich: Falsch

ForbiddenException

Der Client ist nicht authentifiziert.

message

Der Client ist nicht authentifiziert.

Typ: Zeichenfolge

Erforderlich: Falsch

errorCode

403

Typ: Zeichenfolge

Erforderlich: Falsch

InternalServerErrorException

Beim Service AWS Serverless Application Repository trat ein interner Fehler auf.

message

Beim Service AWS Serverless Application Repository trat ein interner Fehler auf.

Typ: Zeichenfolge

Erforderlich: Falsch

errorCode

500

Typ: Zeichenfolge

Erforderlich: Falsch

NotFoundException

Die in der Anfrage angegebene Ressource (z. B. eine Anweisung zur Zugriffsrichtlinie) ist nicht vorhanden.

message

Die in der Anfrage angegebene Ressource (z. B. eine Erklärung zur Zugriffsrichtlinie) ist nicht vorhanden.

Typ: Zeichenfolge

Erforderlich: Falsch

errorCode

404

Typ: Zeichenfolge

Erforderlich: Falsch

TooManyRequestsException

Der Client sendet mehr als die zulässige Anzahl von Anfragen pro Zeiteinheit.

message

Der Client sendet mehr als die zulässige Anzahl von Anfragen pro Zeiteinheit.

Typ: Zeichenfolge

Erforderlich: Falsch

errorCode

429

Typ: Zeichenfolge

Erforderlich: Falsch

Weitere Informationen finden Sie auch unter

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs sowie Referenzen finden Sie unter:

ListApplicationDependencies

- [AWS-Befehlszeilenschnittstelle](#)
- [AWS SDK für .NET](#)
- [AWS SDK für C++](#)
- [AWS SDK for Go](#)
- [AWS SDK für Java V2](#)
- [AWS-SDK für JavaScript Version 3](#)
- [AWS SDK für PHP V3](#)
- [AWS SDK für Python](#)
- [AWS SDK für Ruby V3](#)

ApplicationID-Richtlinie für Anwendungen

URI

/applications/*applicationId*/policy

HTTP-Methoden

GET

Vorgangs-ID: GetApplicationPolicy

Ruft die Richtlinie für die Anwendung ab.

Pfadparameter

Name	Typ	Erforderlich	Beschreibung
<i>applicationId</i>	String	True	Der Amazon-Ressourcenname (ARN) der Anwendung.

Antworten

Statuscode	Antwortmodell	Beschreibung
200	ApplicationPolicy	Herzlichen Glückwunsch
400	BadRequestException	Einer der Parameter der Anfrage ist ungültig.
403	ForbiddenException	Der Client ist nicht authentifiziert.
404	NotFoundException	Die in der Anfrage angegebene Ressource (z. B. eine Anweisung zur Zugriffsrichtlinie) ist nicht vorhanden.

Statuscode	Antwortmodell	Beschreibung
429	TooManyRequestsException	Der Client sendet mehr als die zulässige Anzahl von Anfragen pro Zeiteinheit.
500	InternalServerErrorException	Beim Service AWS Serverless Application Repository trat ein interner Fehler auf.

PUT

Vorgangs-ID: PutApplicationPolicy

Legt die Berechtigungsrichtlinie für eine Anwendung fest. Eine Liste der für diesen Vorgang unterstützten Aktionen finden Sie unter [Anwendungsberechtigungen](#).

Pfadparameter

Name	Typ	Erforderlich	Beschreibung
<i>applicationId</i>	String	True	Der Amazon-Ressourcenname (ARN) der Anwendung.

Antworten

Statuscode	Antwortmodell	Beschreibung
200	ApplicationPolicy	Herzlichen Glückwunsch
400	BadRequestException	Einer der Parameter der Anfrage ist ungültig.
403	ForbiddenException	Der Client ist nicht authentifiziert.
404	NotFoundException	Die in der Anfrage angegebene Ressource (z. B. eine

Statuscode	Antwortmodell	Beschreibung
429	TooManyRequestsException	Anweisung zur Zugriffsrichtlinie) ist nicht vorhanden. Der Client sendet mehr als die zulässige Anzahl von Anfragen pro Zeiteinheit.
500	InternalServerErrorException	Beim Service AWS Serverless Application Repository trat ein interner Fehler auf.

OPTIONS

Pfadparameter

Name	Typ	Erforderlich	Beschreibung
<i>applicationId</i>	String	True	Der Amazon-Ressourcenname (ARN) der Anwendung.

Antworten

Statuscode	Antwortmodell	Beschreibung
200	None	Antwort von 200

Schemata

Anfragetexte

PUT-Schema

```
{
  "statements": [
    {
```

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

Antworttexte

ApplicationPolicy-Schema

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

BadRequestException-Schema

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

ForbiddenException-Schema

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

NotFoundException-Schema

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

TooManyRequestsException-Schema

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

InternalServerErrorException-Schema

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

Eigenschaften

ApplicationPolicy

Grundsatzserklärungen, die auf die Anwendung angewendet wurden.

statements

Eine Reihe von Grundsatzserklärungen, die auf die Anwendung angewendet wurden.

Typ: Array des Typs [ApplicationPolicyStatement](#)

Erforderlich: True

ApplicationPolicyStatement

Für die Anwendung geltende Grundsatzerklärung.

statementId

Eine eindeutige ID für den Kontoauszug.

Typ: Zeichenfolge

Erforderlich: Falsch

principals

Eine Reihe von AWS Konto-IDs, mit denen die Anwendung gemeinsam genutzt werden soll, oder *, um die Anwendung öffentlich zu machen.

Typ: Array des Typs string

Erforderlich: True

actions

Eine Liste der Aktionen, die für diesen Vorgang unterstützt werden, finden Sie unter [Anwendungsberechtigungen](#).

Typ: Array des Typs string

Erforderlich: True

principalOrgIDs

Die AWS Organizations ID, mit der die Anwendung gemeinsam genutzt werden soll.

Typ: Array des Typs string

Erforderlich: Falsch

BadRequestException

Einer der Parameter der Anfrage ist ungültig.

message

Einer der Parameter der Anfrage ist ungültig.

Typ: Zeichenfolge

Erforderlich: Falsch

errorCode

400

Typ: Zeichenfolge

Erforderlich: Falsch

ForbiddenException

Der Client ist nicht authentifiziert.

message

Der Client ist nicht authentifiziert.

Typ: Zeichenfolge

Erforderlich: Falsch

errorCode

403

Typ: Zeichenfolge

Erforderlich: Falsch

InternalServerErrorException

Beim Service AWS Serverless Application Repository trat ein interner Fehler auf.

message

Beim Service AWS Serverless Application Repository trat ein interner Fehler auf.

Typ: Zeichenfolge

Erforderlich: Falsch

errorCode

500

Typ: Zeichenfolge

Erforderlich: Falsch

NotFoundException

Die in der Anfrage angegebene Ressource (z. B. eine Anweisung zur Zugriffsrichtlinie) ist nicht vorhanden.

message

Die in der Anfrage angegebene Ressource (z. B. eine Erklärung zur Zugriffsrichtlinie) ist nicht vorhanden.

Typ: Zeichenfolge

Erforderlich: Falsch

errorCode

404

Typ: Zeichenfolge

Erforderlich: Falsch

TooManyRequestsException

Der Client sendet mehr als die zulässige Anzahl von Anfragen pro Zeiteinheit.

message

Der Client sendet mehr als die zulässige Anzahl von Anfragen pro Zeiteinheit.

Typ: Zeichenfolge

Erforderlich: Falsch

errorCode

429

Typ: Zeichenfolge

Erforderlich: Falsch

Weitere Informationen finden Sie auch unter

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs sowie Referenzen finden Sie unter:

GetApplicationPolicy

- [AWS-Befehlszeilenschnittstelle](#)
- [AWS SDK für .NET](#)
- [AWS SDK für C++](#)
- [AWS SDK for Go](#)
- [AWS SDK für Java V2](#)
- [AWS-SDK für JavaScript Version 3](#)
- [AWS SDK für PHP V3](#)
- [AWS SDK für Python](#)
- [AWS SDK für Ruby V3](#)

PutApplicationPolicy

- [AWS-Befehlszeilenschnittstelle](#)
- [AWS SDK für .NET](#)
- [AWS SDK für C++](#)
- [AWS SDK for Go](#)
- [AWS SDK für Java V2](#)
- [AWS-SDK für JavaScript Version 3](#)

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

Applications applicationId Templates

URI

/applications/*applicationId*/templates

HTTP-Methoden

POST

Vorgangs-ID: CreateCloudFormationTemplate

Erstellt eine AWS CloudFormation Vorlage.

Pfadparameter

Name	Typ	Erforderlich	Beschreibung
<i>applicationId</i>	String	True	Der Amazon-Resourcenname (ARN) der Anwendung.

Antworten

Statuscode	Antwortmodell	Beschreibung
201	TemplateDetails	Herzlichen Glückwunsch
400	BadRequestException	Einer der Parameter der Anfrage ist ungültig.
403	ForbiddenException	Der Client ist nicht authentifiziert.

Statuscode	Antwortmodell	Beschreibung
404	NotFoundException	Die in der Anfrage angegebene Ressource (z. B. eine Anweisung zur Zugriffsrichtlinie) ist nicht vorhanden.
429	TooManyRequestsException	Der Client sendet mehr als die zulässige Anzahl von Anfragen pro Zeiteinheit.
500	InternalServerErrorException	Beim Service AWS Serverless Application Repository trat ein interner Fehler auf.

OPTIONS

Pfadparameter

Name	Typ	Erforderlich	Beschreibung
<i>applicationId</i>	String	True	Der Amazon-Ressourcenname (ARN) der Anwendung.

Antworten

Statuscode	Antwortmodell	Beschreibung
200	None	Antwort von 200

Schemata

Anfragetexte

POST-Schema

```
{
```

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

Antworttexte

TemplateDetails-Schema

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

BadRequestException-Schema

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

ForbiddenException-Schema

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

NotFoundException-Schema

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

TooManyRequestsException-Schema

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

InternalServerErrorException-Schema

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

Eigenschaften

BadRequestException

Einer der Parameter der Anfrage ist ungültig.

message

Einer der Parameter der Anfrage ist ungültig.

Typ: Zeichenfolge

Erforderlich: Falsch

errorCode

400

Typ: Zeichenfolge

Erforderlich: Falsch

CreateCloudFormationTemplateInput

Erstellen Sie eine Vorlagenanfrage.

semanticVersion

Die semantische Version der Anwendung:

<https://semver.org/>

Typ: Zeichenfolge

Erforderlich: Falsch

ForbiddenException

Der Client ist nicht authentifiziert.

message

Der Client ist nicht authentifiziert.

Typ: Zeichenfolge

Erforderlich: Falsch

errorCode

403

Typ: Zeichenfolge

Erforderlich: Falsch

InternalServerErrorException

Beim Service AWS Serverless Application Repository trat ein interner Fehler auf.

message

Beim Service AWS Serverless Application Repository trat ein interner Fehler auf.

Typ: Zeichenfolge

Erforderlich: Falsch

errorCode

500

Typ: Zeichenfolge

Erforderlich: Falsch

NotFoundException

Die in der Anfrage angegebene Ressource (z. B. eine Anweisung zur Zugriffsrichtlinie) ist nicht vorhanden.

message

Die in der Anfrage angegebene Ressource (z. B. eine Erklärung zur Zugriffsrichtlinie) ist nicht vorhanden.

Typ: Zeichenfolge

Erforderlich: Falsch

errorCode

404

Typ: Zeichenfolge

Erforderlich: Falsch

TemplateDetails

Details der Vorlage.

templateId

Die von zurückgegebene UUID. CreateCloudFormationTemplate

Muster: [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}

Typ: Zeichenfolge

Erforderlich: True

templateUrl

Ein Link zur Vorlage, mit AWS CloudFormation der die Anwendung bereitgestellt werden kann.

Typ: Zeichenfolge

Erforderlich: True

applicationId

Der Amazon-Ressourcenname (ARN) der -Anwendung

Typ: Zeichenfolge

Erforderlich: True

semanticVersion

Die semantische Version der Anwendung:

<https://semver.org/>

Typ: Zeichenfolge

Erforderlich: True

status

Status des Workflows zur Vorlagenerstellung.

Mögliche Werte: PREPARING | ACTIVE | EXPIRED

Typ: Zeichenfolge

Erforderlich: True

Werte: PREPARING | ACTIVE | EXPIRED

creationTime

Datum und Uhrzeit der Erstellung dieser Ressource.

Typ: Zeichenfolge

Erforderlich: True

expirationTime

Datum und Uhrzeit des Ablaufs dieser Vorlage. Vorlagen laufen 1 Stunde nach ihrer Erstellung ab.

Typ: Zeichenfolge

Erforderlich: True

TooManyRequestsException

Der Client sendet mehr als die zulässige Anzahl von Anfragen pro Zeiteinheit.

message

Der Client sendet mehr als die zulässige Anzahl von Anfragen pro Zeiteinheit.

Typ: Zeichenfolge

Erforderlich: Falsch

errorCode

429

Typ: Zeichenfolge

Erforderlich: Falsch

Weitere Informationen finden Sie auch unter

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs sowie Referenzen finden Sie unter:

CreateCloudFormationTemplate

- [AWS-Befehlszeilenschnittstelle](#)
- [AWS SDK für .NET](#)
- [AWS SDK für C++](#)
- [AWS SDK for Go](#)
- [AWS SDK für Java V2](#)
- [AWS-SDK für JavaScript Version 3](#)
- [AWS SDK für PHP V3](#)
- [AWS SDK für Python](#)
- [AWS SDK für Ruby V3](#)

Applications applicationId Templates templateId

URI

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

HTTP-Methoden

GET

Vorgangs-ID: GetCloudFormationTemplate

Ruft die angegebene AWS CloudFormation Vorlage ab.

Pfadparameter

Name	Typ	Erforderlich	Beschreibung
<i>applicationId</i>	String	True	Der Amazon-Ressourcenname (ARN) der Anwendung.
<i>Vorlagen-ID</i>	String	True	Die von zurückgegebene UUID. CreateCloudFormationTemplate Muster: [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}

Antworten

Statuscode	Antwortmodell	Beschreibung
200	TemplateDetails	Herzlichen Glückwunsch

Statuscode	Antwortmodell	Beschreibung
400	BadRequestException	Einer der Parameter der Anfrage ist ungültig.
403	ForbiddenException	Der Client ist nicht authentifiziert.
404	NotFoundException	Die in der Anfrage angegebene Ressource (z. B. eine Anweisung zur Zugriffsrichtlinie) ist nicht vorhanden.
429	TooManyRequestsException	Der Client sendet mehr als die zulässige Anzahl von Anfragen pro Zeiteinheit.
500	InternalServerErrorException	Beim Service AWS Serverless Application Repository trat ein interner Fehler auf.

OPTIONS

Pfadparameter

Name	Typ	Erforderlich	Beschreibung
<i>applicationId</i>	String	True	Der Amazon-Ressourcenname (ARN) der Anwendung.
<i>Vorlagen-ID</i>	String	True	Die von zurückgegebene UUID. CreateCloudFormationTemplate Muster: [0-9a-fa-F]{8}\ - [0-9a-fa-F]{4}\ - [0-9a-fa-F]{4}\ - [0-9a-

Name	Typ	Erforderlich	Beschreibung
			fa-F] {4}\ - [0-9a-fa-F]{12}
Antworten			
Statuscode		Antwortmodell	Beschreibung
200		None	200 Antworten

Schemata

Antworttexte

TemplateDetails-Schema

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

BadRequestException-Schema

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

ForbiddenException-Schema

```
{
```

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

NotFoundException-Schema

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

TooManyRequestsException-Schema

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

InternalServerErrorException-Schema

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

Eigenschaften

BadRequestException

Einer der Parameter der Anfrage ist ungültig.

message

Einer der Parameter der Anfrage ist ungültig.

Typ: Zeichenfolge

Erforderlich: Falsch

errorCode

400

Typ: Zeichenfolge

Erforderlich: Falsch

ForbiddenException

Der Client ist nicht authentifiziert.

message

Der Client ist nicht authentifiziert.

Typ: Zeichenfolge

Erforderlich: Falsch

errorCode

403

Typ: Zeichenfolge

Erforderlich: Falsch

InternalServerErrorException

Beim Service AWS Serverless Application Repository trat ein interner Fehler auf.

message

Beim Service AWS Serverless Application Repository trat ein interner Fehler auf.

Typ: Zeichenfolge

Erforderlich: Falsch

errorCode

500

Typ: Zeichenfolge
Erforderlich: Falsch

NotFoundException

Die in der Anfrage angegebene Ressource (z. B. eine Anweisung zur Zugriffsrichtlinie) ist nicht vorhanden.

message

Die in der Anfrage angegebene Ressource (z. B. eine Erklärung zur Zugriffsrichtlinie) ist nicht vorhanden.

Typ: Zeichenfolge
Erforderlich: Falsch

errorCode

404

Typ: Zeichenfolge
Erforderlich: Falsch

TemplateDetails

Details der Vorlage.

templateId

Die von zurückgegebene UUID. CreateCloudFormationTemplate

Muster: [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}

Typ: Zeichenfolge
Erforderlich: True

templateUrl

Ein Link zur Vorlage, mit AWS CloudFormation der die Anwendung bereitgestellt werden kann.

Typ: Zeichenfolge

Erforderlich: True

applicationId

Der Amazon-Ressourcenname (ARN) der -Anwendung

Typ: Zeichenfolge

Erforderlich: True

semanticVersion

Die semantische Version der Anwendung:

<https://semver.org/>

Typ: Zeichenfolge

Erforderlich: True

status

Status des Workflows zur Vorlagenerstellung.

Mögliche Werte: PREPARING | ACTIVE | EXPIRED

Typ: Zeichenfolge

Erforderlich: True

Werte: PREPARING | ACTIVE | EXPIRED

creationTime

Datum und Uhrzeit der Erstellung dieser Ressource.

Typ: Zeichenfolge

Erforderlich: True

expirationTime

Datum und Uhrzeit des Ablaufs dieser Vorlage. Vorlagen laufen 1 Stunde nach ihrer Erstellung ab.

Typ: Zeichenfolge

Erforderlich: True

TooManyRequestsException

Der Client sendet mehr als die zulässige Anzahl von Anfragen pro Zeiteinheit.

message

Der Client sendet mehr als die zulässige Anzahl von Anfragen pro Zeiteinheit.

Typ: Zeichenfolge

Erforderlich: Falsch

errorCode

429

Typ: Zeichenfolge

Erforderlich: Falsch

Weitere Informationen finden Sie auch unter

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs sowie Referenzen finden Sie unter:

GetCloudFormationTemplate

- [AWS-Befehlszeilenschnittstelle](#)
- [AWS SDK für .NET](#)
- [AWS SDK für C++](#)
- [AWS SDK for Go](#)
- [AWS SDK für Java V2](#)
- [AWS-SDK für JavaScript Version 3](#)
- [AWS SDK für PHP V3](#)
- [AWS SDK für Python](#)

- [AWS SDK für Ruby V3](#)

Applications applicationId Unshare

URI

/applications/*applicationId*/unshare

HTTP-Methoden

POST

Vorgangs-ID: UnshareApplication

Macht die Freigabe einer Anwendung aus einer AWS Organisation rückgängig.

Dieser Vorgang kann nur über das Verwaltungskonto der Organisation aufgerufen werden.

Pfadparameter

Name	Typ	Erforderlich	Beschreibung
<i>applicationId</i>	String	True	Der Amazon-Ressourcenname (ARN) der Anwendung.

Antworten

Statuscode	Antwortmodell	Beschreibung
204	None	Herzlichen Glückwunsch
400	BadRequestException	Einer der Parameter der Anfrage ist ungültig.
403	ForbiddenException	Der Client ist nicht authentifiziert.
404	NotFoundException	Die in der Anfrage angegebene Ressource (z. B. eine

Statuscode	Antwortmodell	Beschreibung
429	TooManyRequestsException	Anweisung zur Zugriffsr ichtlinie) ist nicht vorhanden. Der Client sendet mehr als die zulässige Anzahl von Anfragen pro Zeiteinheit.
500	InternalServerErrorException	Beim Service AWS Serverless Application Repository trat ein interner Fehler auf.

OPTIONS

Pfadparameter

Name	Typ	Erforderlich	Beschreibung
<i>applicationId</i>	String	True	Der Amazon-Re sourcename (ARN) der Anwendung.

Antworten

Statuscode	Antwortmodell	Beschreibung
200	None	Antwort von 200

Schemata

Anfragetexte

POST-Schema

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

Antworttexte

BadRequestException-Schema

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

ForbiddenException-Schema

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

NotFoundException-Schema

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

TooManyRequestsException-Schema

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

InternalServerErrorException-Schema

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

Eigenschaften

BadRequestException

Einer der Parameter der Anfrage ist ungültig.

message

Einer der Parameter der Anfrage ist ungültig.

Typ: Zeichenfolge

Erforderlich: Falsch

errorCode

400

Typ: Zeichenfolge

Erforderlich: Falsch

ForbiddenException

Der Client ist nicht authentifiziert.

message

Der Client ist nicht authentifiziert.

Typ: Zeichenfolge

Erforderlich: Falsch

errorCode

403

Typ: Zeichenfolge

Erforderlich: Falsch

InternalServerErrorException

Beim Service AWS Serverless Application Repository trat ein interner Fehler auf.

message

Beim Service AWS Serverless Application Repository trat ein interner Fehler auf.

Typ: Zeichenfolge

Erforderlich: Falsch

errorCode

500

Typ: Zeichenfolge

Erforderlich: Falsch

NotFoundException

Die in der Anfrage angegebene Ressource (z. B. eine Anweisung zur Zugriffsrichtlinie) ist nicht vorhanden.

message

Die in der Anfrage angegebene Ressource (z. B. eine Erklärung zur Zugriffsrichtlinie) ist nicht vorhanden.

Typ: Zeichenfolge

Erforderlich: Falsch

errorCode

404

Typ: Zeichenfolge

Erforderlich: Falsch

TooManyRequestsException

Der Client sendet mehr als die zulässige Anzahl von Anfragen pro Zeiteinheit.

message

Der Client sendet mehr als die zulässige Anzahl von Anfragen pro Zeiteinheit.

Typ: Zeichenfolge

Erforderlich: Falsch

errorCode

429

Typ: Zeichenfolge

Erforderlich: Falsch

UnshareApplicationInput

Teilen Sie die Anwendungsanfrage nicht weiter.

organizationId

Die AWS Organizations ID, für die die Freigabe der Anwendung aufgehoben werden soll.

Typ: Zeichenfolge

Erforderlich: True

Weitere Informationen finden Sie auch unter

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs sowie Referenzen finden Sie unter:

UnshareApplication

- [AWS-Befehlszeilenschnittstelle](#)
- [AWS SDK für .NET](#)
- [AWS SDK für C++](#)

- [AWS SDK for Go](#)
- [AWS SDK für Java V2](#)
- [AWS-SDK für JavaScript Version 3](#)
- [AWS SDK für PHP V3](#)
- [AWS SDK für Python](#)
- [AWS SDK für Ruby V3](#)

ApplicationID-Versionen von Anwendungen

URI

/applications/*applicationId*/versions

HTTP-Methoden

GET

Vorgangs-ID: ListApplicationVersions

Listet die Versionen für die angegebene Anwendung auf.

Pfadparameter

Name	Typ	Erforderlich	Beschreibung
<i>applicationId</i>	String	True	Der Amazon-Ressourcenname (ARN) der Anwendung.

Abfrageparameter

Name	Typ	Erforderlich	Beschreibung
Max. Anzahl Artikel	String	False	Die Gesamtzahl der zurückzusendenden Artikel.

Name	Typ	Erforderlich	Beschreibung
nextToken	String	False	Ein Token, um anzugeben, wo mit dem Paginieren begonnen werden soll.

Antworten

Statuscode	Antwortmodell	Beschreibung
200	ApplicationVersionPage	Herzlichen Glückwunsch
400	BadRequestException	Einer der Parameter der Anfrage ist ungültig.
403	ForbiddenException	Der Client ist nicht authentifiziert.
404	NotFoundException	Die in der Anfrage angegebene Ressource (z. B. eine Anweisung zur Zugriffsrichtlinie) ist nicht vorhanden.
429	TooManyRequestsException	Der Client sendet mehr als die zulässige Anzahl von Anfragen pro Zeiteinheit.
500	InternalServerErrorException	Beim Service AWS Serverless Application Repository trat ein interner Fehler auf.

OPTIONS

Pfadparameter

Name	Typ	Erforderlich	Beschreibung
<i>applicationId</i>	String	True	Der Amazon-Ressourcenname (ARN) der Anwendung.

Antworten

Statuscode	Antwortmodell	Beschreibung
200	None	Antwort von 200

Schemata

Antworttexte

ApplicationVersionPage-Schema

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

BadRequestException-Schema

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



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

ForbiddenException-Schema

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

NotFoundException-Schema

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

TooManyRequestsException-Schema

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

InternalServerErrorException-Schema

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

Eigenschaften

ApplicationVersionPage

Eine Liste der Versionszusammenfassungen für die Anwendung.

versions

Eine Reihe von Versionszusammenfassungen für die Anwendung.

Typ: Array des Typs [VersionSummary](#)

Erforderlich: True

nextToken

Das Token zum Anfordern der nächsten Ergebnisseite.

Typ: Zeichenfolge

Erforderlich: Falsch

BadRequestException

Einer der Parameter der Anfrage ist ungültig.

message

Einer der Parameter der Anfrage ist ungültig.

Typ: Zeichenfolge

Erforderlich: Falsch

errorCode

400

Typ: Zeichenfolge

Erforderlich: Falsch

ForbiddenException

Der Client ist nicht authentifiziert.

message

Der Client ist nicht authentifiziert.

Typ: Zeichenfolge

Erforderlich: Falsch

errorCode

403

Typ: Zeichenfolge

Erforderlich: Falsch

InternalServerErrorException

Beim Service AWS Serverless Application Repository trat ein interner Fehler auf.

message

Beim Service AWS Serverless Application Repository trat ein interner Fehler auf.

Typ: Zeichenfolge

Erforderlich: Falsch

errorCode

500

Typ: Zeichenfolge

Erforderlich: Falsch

NotFoundException

Die in der Anfrage angegebene Ressource (z. B. eine Anweisung zur Zugriffsrichtlinie) ist nicht vorhanden.

message

Die in der Anfrage angegebene Ressource (z. B. eine Erklärung zur Zugriffsrichtlinie) ist nicht vorhanden.

Typ: Zeichenfolge

Erforderlich: Falsch

errorCode

404

Typ: Zeichenfolge

Erforderlich: Falsch

TooManyRequestsException

Der Client sendet mehr als die zulässige Anzahl von Anfragen pro Zeiteinheit.

message

Der Client sendet mehr als die zulässige Anzahl von Anfragen pro Zeiteinheit.

Typ: Zeichenfolge

Erforderlich: Falsch

errorCode

429

Typ: Zeichenfolge

Erforderlich: Falsch

VersionSummary

Eine Zusammenfassung der Anwendungsversion.

applicationId

Der Amazon-Ressourcenname (ARN) der -Anwendung

Typ: Zeichenfolge

Erforderlich: True

semanticVersion

Die semantische Version der Anwendung:

<https://semver.org/>

Typ: Zeichenfolge

Erforderlich: True

sourceCodeUrl

Ein Link zu einem öffentlichen Repository für den Quellcode Ihrer Anwendung, z. B. die URL eines bestimmten GitHub Commits.

Typ: Zeichenfolge

Erforderlich: Falsch

creationTime

Datum und Uhrzeit der Erstellung dieser Ressource.

Typ: Zeichenfolge

Erforderlich: True

Weitere Informationen finden Sie auch unter

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs sowie Referenzen finden Sie unter:

ListApplicationVersions

- [AWS-Befehlszeilenschnittstelle](#)
- [AWS SDK für .NET](#)
- [AWS SDK für C++](#)
- [AWS SDK for Go](#)
- [AWS SDK für Java V2](#)
- [AWS-SDK für JavaScript Version 3](#)
- [AWS SDK für PHP V3](#)
- [AWS SDK für Python](#)
- [AWS SDK für Ruby V3](#)

Anwendungen ApplicationID-Versionen SemanticVersion

URI

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

HTTP-Methoden

PUT

Vorgangs-ID: CreateApplicationVersion

Erstellt eine Anwendungsversion

Pfadparameter

Name	Typ	Erforderlich	Beschreibung
<i>applicationId</i>	String	True	Der Amazon-Ressourcenname (ARN) der Anwendung.
<i>Semantische Version</i>	String	True	Die semantische Version der neuen Version.

Antworten

Statuscode	Antwortmodell	Beschreibung
201	Version	Herzlichen Glückwunsch
400	BadRequestException	Einer der Parameter der Anfrage ist ungültig.
403	ForbiddenException	Der Client ist nicht authentifiziert.
409	ConflictException	Die Ressource ist bereits vorhanden.

Statuscode	Antwortmodell	Beschreibung
429	TooManyRequestsException	Der Client sendet mehr als die zulässige Anzahl von Anfragen pro Zeiteinheit.
500	InternalServerErrorException	Beim Service AWS Serverless Application Repository trat ein interner Fehler auf.

OPTIONS

Pfadparameter

Name	Typ	Erforderlich	Beschreibung
<i>applicationId</i>	String	True	Der Amazon-Ressourcenname (ARN) der Anwendung.
<i>Semantische Version</i>	String	True	Die semantische Version der neuen Version.

Antworten

Statuscode	Antwortmodell	Beschreibung
200	None	Antwort von 200

Schemata

Anfragetexte

PUT-Schema

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

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

Antworttexte

Version-Schema

```
{
  "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-Schema

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

ForbiddenException-Schema

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

ConflictException-Schema

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

TooManyRequestsException-Schema

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

InternalServerErrorException-Schema

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

Eigenschaften

BadRequestException

Einer der Parameter der Anfrage ist ungültig.

message

Einer der Parameter der Anfrage ist ungültig.

Typ: Zeichenfolge

Erforderlich: Falsch

errorCode

400

Typ: Zeichenfolge

Erforderlich: Falsch

Capability

Werte, die angegeben werden müssen, um einige Anwendungen bereitzustellen.

CAPABILITY_IAM

CAPABILITY_NAMED_IAM

CAPABILITY_AUTO_EXPAND

CAPABILITY_RESOURCE_POLICY

ConflictException

Die Ressource ist bereits vorhanden.

message

Die Ressource ist bereits vorhanden.

Typ: Zeichenfolge

Erforderlich: Falsch

errorCode

409

Typ: Zeichenfolge

Erforderlich: Falsch

CreateApplicationVersionInput

Erstellen Sie eine Versionsanfrage.

templateBody

Die AWS SAM unformatierte Paketvorlage Ihrer Anwendung.

Typ: Zeichenfolge

Erforderlich: Falsch

templateUrl

Ein Link zur verpackten AWS SAM Vorlage Ihrer Anwendung.

Typ: Zeichenfolge

Erforderlich: Falsch

sourceCodeUrl

Ein Link zu einem öffentlichen Repository für den Quellcode Ihrer Anwendung, z. B. die URL eines bestimmten GitHub Commits.

Typ: Zeichenfolge

Erforderlich: Falsch

sourceCodeArchiveUrl

Ein Link zu dem S3-Objekt, das das ZIP-Archiv des Quellcodes für diese Version Ihrer Anwendung enthält.

Maximale Größe 50 MB

Typ: Zeichenfolge

Erforderlich: Falsch

ForbiddenException

Der Client ist nicht authentifiziert.

message

Der Client ist nicht authentifiziert.

Typ: Zeichenfolge

Erforderlich: Falsch

errorCode

403

Typ: Zeichenfolge

Erforderlich: Falsch

InternalServerErrorException

Beim Service AWS Serverless Application Repository trat ein interner Fehler auf.

message

Beim Service AWS Serverless Application Repository trat ein interner Fehler auf.

Typ: Zeichenfolge

Erforderlich: Falsch

errorCode

500

Typ: Zeichenfolge

Erforderlich: Falsch

ParameterDefinition

Von der Anwendung unterstützte Parameter.

name

Der Name des Parameters.

Typ: Zeichenfolge

Erforderlich: True

defaultValue

Gibt einen Wert des entsprechenden Typs an, den die Vorlage nutzen soll, wenn zum Zeitpunkt der Stack-Erstellung kein Wert angegeben wird. Wenn Sie Einschränkungen für den Parameter definieren, müssen Sie einen Wert angeben, der diese Einschränkungen einhält.

Typ: Zeichenfolge

Erforderlich: Falsch

description

Eine Zeichenfolge mit bis zu 4.000 Zeichen, die den Parameter beschreibt.

Typ: Zeichenfolge

Erforderlich: Falsch

type

Der Typ des Parameters.

Zulässige Werte: `String` | `Number` | `List<Number>` | `CommaDelimitedList`

`String`: Eine wörtliche Zeichenfolge.

Benutzer können beispielsweise angeben `"MyUserName"`.

Number: Eine Ganzzahl oder eine Fließkommazahl. AWS CloudFormationvalidiert den Parameterwert als Zahl. Wenn Sie den Parameter jedoch an einer anderen Stelle in Ihrer Vorlage verwenden (z. B. mithilfe der Ref systeminternen Funktion), wird der Parameterwert zu einer Zeichenfolge.

Benutzer könnten beispielsweise angeben. "8888"

List<Number>: Eine Reihe von Ganzzahlen oder Gleitkommazahlen, die durch Kommas getrennt sind. AWS CloudFormationvalidiert den Parameterwert als Zahlen. Wenn Sie den Parameter jedoch an einer anderen Stelle in Ihrer Vorlage verwenden (z. B. mithilfe der Ref systeminternen Funktion), wird der Parameterwert zu einer Liste von Zeichenfolgen.

Benutzer könnten beispielsweise „80,20“ angeben und dann das Ergebnis anzeigen. Ref ["80", "20"]

CommaDelimitedList: Ein Array von Literalzeichenfolgen, die durch Kommas getrennt sind. Die Gesamtanzahl von Zeichenfolgen sollte um 1 höher sein als die Gesamtanzahl von Kommas. Außerdem wird jede Elementzeichenfolge durch Leerzeichen gekürzt.

Benutzer könnten beispielsweise „test, dev, prod“ angeben und dann die Ergebnisse angeben. Ref ["test", "dev", "prod"]

Typ: Zeichenfolge
Erforderlich: Falsch

noEcho

Ob der Parameterwert maskiert werden soll, wenn jemand einen Aufruf tätigt, der den Stack beschreibt. Wenn Sie den Wert auf true setzen, wird der Parameterwert mit Sternchen (*****) maskiert.

Typ: Boolesch
Erforderlich: Falsch

allowedPattern

Gibt einen regulären Ausdruck an, der die zulässigen Muster für String-Typen beschreibt.

Typ: Zeichenfolge
Erforderlich: Falsch

constraintDescription

Gibt eine Zeichenfolge als Beschreibung für eine Einschränkung an. Diese Beschreibung wird bei Verstößen gegen die Einschränkung angezeigt. Für einen Parameter mit einem zulässigen Muster `[A-Za-z0-9]+` beispielsweise wird ohne Einschränkungsbeschreibung die folgende Fehlermeldung angezeigt, wenn der Benutzer einen ungültigen Wert angibt:

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

Durch Hinzufügen einer Beschränkungsbeschreibung, z. B. „darf nur Groß- und Kleinbuchstaben und Zahlen enthalten“, können Sie die folgende benutzerdefinierte Fehlermeldung anzeigen:

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

Typ: Zeichenfolge

Erforderlich: Falsch

minValue

Ein numerischer Wert, der den kleinsten numerischen Wert bestimmt, den Sie für `Number` Typen zulassen möchten.

Typ: Ganzzahl

Erforderlich: Falsch

maxValue

Ein numerischer Wert, der den größten numerischen Wert bestimmt, den Sie für `Number` Typen zulassen möchten.

Typ: Ganzzahl

Erforderlich: Falsch

minLength

Ein ganzzahliger Wert, der die kleinste Anzahl von Zeichen bestimmt, die Sie für `String` Typen zulassen möchten.

Typ: Ganzzahl

Erforderlich: Falsch

maxLength

Ein ganzzahliger Wert, der die größte Anzahl von Zeichen bestimmt, die Sie für `String` Typen zulassen möchten.

Typ: Ganzzahl

Erforderlich: Falsch

allowedValues

Gibt ein Array mit einer Liste der für den Parameter zulässigen Werte an.

Typ: Array des Typs `string`

Erforderlich: Falsch

referencedByResources

Eine Liste von AWS SAM Ressourcen, die diesen Parameter verwenden.

Typ: Array des Typs `string`

Erforderlich: `True`

TooManyRequestsException

Der Client sendet mehr als die zulässige Anzahl von Anfragen pro Zeiteinheit.

message

Der Client sendet mehr als die zulässige Anzahl von Anfragen pro Zeiteinheit.

Typ: Zeichenfolge

Erforderlich: Falsch

errorCode

429

Typ: Zeichenfolge
Erforderlich: Falsch

Version

Einzelheiten zur Anwendungsversion.

applicationId

Der Amazon-Ressourcenname (ARN) der -Anwendung

Typ: Zeichenfolge
Erforderlich: True

semanticVersion

Die semantische Version der Anwendung:

<https://semver.org/>

Typ: Zeichenfolge
Erforderlich: True

sourceCodeUrl

Ein Link zu einem öffentlichen Repository für den Quellcode Ihrer Anwendung, z. B. die URL eines bestimmten GitHub Commits.

Typ: Zeichenfolge
Erforderlich: Falsch

sourceCodeArchiveUrl

Ein Link zu dem S3-Objekt, das das ZIP-Archiv des Quellcodes für diese Version Ihrer Anwendung enthält.

Maximale Größe 50 MB

Typ: Zeichenfolge

Erforderlich: Falsch

templateUrl

Ein Link zur verpackten AWS SAM Vorlage Ihrer Anwendung.

Typ: Zeichenfolge

Erforderlich: True

creationTime

Datum und Uhrzeit der Erstellung dieser Ressource.

Typ: Zeichenfolge

Erforderlich: True

parameterDefinitions

Eine Reihe von Parametertypen, die von der Anwendung unterstützt werden.

Typ: Array des Typs [ParameterDefinition](#)

Erforderlich: True

requiredCapabilities

Eine Liste von Werten, die Sie angeben müssen, bevor Sie bestimmte Anwendungen bereitstellen können. Einige Anwendungen können Ressourcen enthalten, die sich auf die Berechtigungen in Ihrem AWS Konto auswirken können, z. B. indem neue Benutzer AWS Identity and Access Management (IAM) erstellt werden. Für diese Anwendungen müssen Sie ihre Funktionen ausdrücklich bestätigen, indem Sie diesen Parameter angeben.

Die einzig gültigen Werte sind

CAPABILITY_IAM, CAPABILITY_NAMED_IAM, CAPABILITY_RESOURCE_POLICY,
und CAPABILITY_AUTO_EXPAND.

Für die folgenden Ressourcen müssen Sie CAPABILITY_IAM oder angeben CAPABILITY_NAMED_IAM: [AWS::IAM::Group](#), [AWS::IAM::InstanceProfile](#), [AWS::IAM::Policy](#), und [AWS::IAM::Role](#). Wenn die Anwendung IAM-Ressourcen enthält, können Sie entweder

CAPABILITY_IAM oder CAPABILITY_NAMED_IAM angeben. Beinhaltet die Anwendung IAM-Ressourcen mit benutzerdefinierten Namen, müssen Sie CAPABILITY_NAMED_IAM angeben.

Für die folgenden Ressourcen müssen Sie Folgendes angeben CAPABILITY_RESOURCE_POLICY: [AWS::Lambda::Permission](#), [AWS::IAM::Policy](#), [AWS::ApplicationAutoScaling::ScalingPolicy](#), [AWS::S3::BucketPolicy](#) und [AWS::SQS::QueuePolicy](#) [AWS::SNS::TopicPolicy](#)

Bei Anwendungen, die eine oder mehrere verschachtelte Anwendungen enthalten, müssen Sie CAPABILITY_AUTO_EXPAND angeben.

Wenn Ihre Anwendungsvorlage eine der oben genannten Ressourcen enthält, empfehlen wir Ihnen, vor der Bereitstellung alle mit der Anwendung verbundenen Berechtigungen zu überprüfen. Wenn Sie diesen Parameter nicht für eine Anwendung angeben, die Funktionen benötigt, schlägt der Aufruf fehl.

Typ: Array des Typs [Capability](#)

Erforderlich: True

resourcesSupported

Ob alle in dieser Anwendung enthaltenen AWS Ressourcen in der Region unterstützt werden, in der sie abgerufen wird.

Typ: Boolesch

Erforderlich: True

Weitere Informationen finden Sie auch unter

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs sowie Referenzen finden Sie unter:

CreateApplicationVersion

- [AWS-Befehlszeilenschnittstelle](#)
- [AWS SDK für .NET](#)
- [AWS SDK für C++](#)
- [AWS SDK for Go](#)
- [AWS SDK für Java V2](#)

- [AWS-SDK für JavaScript Version 3](#)
- [AWS SDK für PHP V3](#)
- [AWS SDK für Python](#)
- [AWS SDK für Ruby V3](#)

Dokumentverlauf

- API-Version: aktuelle
- Letzte Aktualisierung der Dokumentation: 10. März 2020

In der folgenden Tabelle sind die wichtigen Änderungen in jeder Version des AWS Serverless Application Repository-Entwicklerhandbuchs. Um Benachrichtigungen über Aktualisierungen dieser Dokumentation zu erhalten, können Sie einen RSS-Feed abonnieren.

Änderung	Beschreibung	Datum
Aktualisierungen zum Teilen und Beschränken des Zugriffs auf Anwendungen	Unterstützung für die gemeinsame Nutzung von Anwendungen für Konten in einer AWS Organisation und die Beschränkung des Zugriffs auf öffentliche Anwendungen für AWS Konten und AWS Organizations wurde hinzugefügt. Weitere Beispiele für die gemeinsame Nutzung von Anwendungen für Benutzer in einer Organisation finden Sie unter Beispiele für AWS Serverless Application Repository Anwendungsrichtlinien . Beispiele für die Beschränkung des Zugriffs auf öffentliche Anwendungen finden Sie unter Identitätsbasierte AWS Serverless Application Repository-Richtlinienbeispiele .	10. März 2020
Neue unterstützte Ressourcen	Eine Reihe zusätzlicher Ressourcen wird nun unterstüt	17. Januar 2020

[Vereinfachter Prozess für die Veröffentlichung von Anwendungen](#)

Der neue Befehl `publish` in der AWS SAM-CLI vereinfacht das Veröffentlichen serverloser Anwendungen im AWS Serverless Application Repository. Ein end-to-end Tutorial zum Herunterladen und Veröffentlichen einer Beispielanwendung finden Sie unter [Quick Start: Publishing Applications](#). Anweisungen zum Veröffentlichen einer Anwendung, die Sie bereits in der AWS Cloud entwickelt und getestet haben, finden Sie unter [Veröffentlichen einer Anwendung über die AWS SAM CLI](#).

21. Dezember 2018

[Unterstützung für verschachtelte Anwendungen und Ebenen](#)

Support für verschachtelte Anwendungen und Ebenen hinzugefügt. Dazu gehören Aktualisierungen der [unterstützten AWS Ressourcen](#) und der [Bestätigung der Anwendungsfunktionen](#).

29. November 2018

[Veröffentlichen von Anwendungen mit benutzerdefinierten IAM-Rollen und Ressourcenrichtlinien](#)

Support für das Veröffentlichen von Anwendungen mit benutzerdefinierten IAM-Rollen und Ressourcenrichtlinien hinzugefügt. Dazu gehören Aktualisierungen der Workflows „[Anwendungen verwenden](#)“ und „[Anwendungen veröffentlichen](#)“ sowie Aktualisierungen der [unterstützten AWS Ressourcen](#) und der [API-Referenz](#) im AWS Serverless Application RepositoryEntwicklerhandbuch.

16. November 2018

[Aktualisierungen der Richtlinienvorlage](#)

Aktualisierungen der unterstützten [Richtlinienvorlagen](#) im AWS Serverless Application RepositoryEntwicklerhandbuch.

26. September 2018

[Aktualisierungen der Dokumentation](#)

Das Thema Authentifizierung und Zugriffskontrolle wurde dem AWS Serverless Application RepositoryEntwicklerhandbuch hinzugefügt.

2. Juli 2018

Öffentliche Veröffentlichung

Öffentliche Version der AWS Serverless Application Repository, die jetzt in 14 AWS Regionen verfügbar ist. Weitere Informationen zu den AWS Regionen, in denen der verfügbar AWS Serverless Application Repository ist, und zu den AWS Serverless Application Repository Endpunkten finden Sie unter [Regionen und Endpunkte](#) in der Allgemeine AWS-Referenz.

20. Februar 2018

Neues Handbuch

Dies ist die erste Vorschauversion des AWS Serverless Application Repository-Entwicklerhandbuches.

30. November 2017

AWS-Glossar

Die neueste AWS-Terminologie finden Sie im [AWS-Glossar](#) in der AWS-Glossar-Referenz.

Die vorliegende Übersetzung wurde maschinell erstellt. Im Falle eines Konflikts oder eines Widerspruchs zwischen dieser übersetzten Fassung und der englischen Fassung (einschließlich infolge von Verzögerungen bei der Übersetzung) ist die englische Fassung maßgeblich.