



Benutzerhandbuch

AWS Toolkit for VS Code



AWS Toolkit for VS Code: Benutzerhandbuch

Copyright © 2024 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

AWS Toolkit for Visual Studio Code	1
Was ist der AWS Toolkit for Visual Studio Code	1
Ähnliche Informationen	1
Amazon Q Developer und Amazon CodeWhisperer	2
Laden Sie das Toolkit herunter	3
Das Toolkit vom VS Code Marketplace herunterladen	3
Zusätzliche IDE-Toolkits von AWS	3
Erste Schritte	4
Das Toolkit for VS Code installieren	4
Voraussetzungen	4
Herunterladen und Installieren von AWS Toolkit for Visual Studio Code	4
Optionale Voraussetzungen	5
Verbindung herstellen zu AWS	6
Voraussetzungen	6
Öffnen Sie das Anmeldefenster	7
Über das Toolkit AWS wird eine Verbindung hergestellt	7
Authentifizierung für Amazon CodeCatalyst	8
AWS Regionen im Wandel	9
Eine Region zum AWS Explorer hinzufügen	10
Eine Region im AWS Explorer ausblenden	10
Konfigurieren einer -Umgebung	10
Konfiguration einer Toolchain für .NET Core	10
Konfigurieren einer -Umgebung Node.js	11
Konfiguration einer Toolchain für Python	11
Konfiguration einer Toolchain für Java	12
Konfigurieren Sie eine Toolchain für Go	12
Verwenden Sie Ihre Toolchain	12
Authentifizierung und Zugriff	13
IAM Identity Center	13
IAM-Anmeldeinformationen	14
Erstellen eines IAM-Benutzers	14
Erstellen einer Datei mit gemeinsamen Anmeldeinformationen aus dem AWS Toolkit for Visual Studio Code	15
Fügen Sie zusätzliche Anmeldeinformationsprofile hinzu	16

AWS Builder-ID	17
Verwenden eines Service für externe Anmeldeinformationen	17
Arbeiten mit AWS	18
Experimentelle Funktionen	19
AWS Entdecker	19
Amazon CodeCatalyst	20
Was ist Amazon CodeCatalyst?	20
Erste Schritte mit CodeCatalyst	21
Arbeiten mit CodeCatalyst-Ressourcen	21
Mit Entwicklungsumgebungen arbeiten	26
Fehlerbehebung	29
Amazon API Gateway	30
AWS App Runner	30
Voraussetzungen	31
Preisgestaltung	34
Erstellen von App-Runner-Services	35
Verwalten von App-Runner-Services	38
AWSAnwendungs-Composer	41
Mit AWS Application Composer arbeiten	42
AWS CDK	43
AWS CDK-Anwendungen	43
AWS CloudFormation-Stacks	46
Löschen einerAWS CloudFormationstapeln	46
Erstellen Sie eine CloudFormation Vorlage	47
AmazonCloudWatchLogs	49
Anzeigen von CloudWatch-Protokollgruppen und Protokoll-Streams	49
Arbeiten mit CloudWatch Protokollereignisse	50
Protokollgruppen durchsuchen	52
Amazon ECR	55
Voraussetzungen	55
Verwendung von Amazon ECR mit Toolkit for VS Code	58
Amazon ECS	67
IntelliSense für Task-Definitionsdateien verwenden	67
Amazon ECS Exec	68
Amazon EventBridge	71
Mit EventBridge Amazon-Schemas arbeiten	71

AWS IAM-Zugriffsanalysator	73
Arbeiten mit AWS IAM Access Analyzer	74
AWS IoT	78
AWS IoT-Voraussetzungen	78
AWS IoTElemente	79
AWS IoT-Zertifikate	80
AWS IoT-Richtlinien	83
AWS Lambda-Funktionen	87
Interaktion mit Remote-Lambda-Funktionen	87
Amazon Redshift	94
Arbeiten mit Amazon Redshift	94
Amazon S3	99
Arbeiten mit S3-Ressourcen	100
Mit S3-Objekten arbeiten	102
AWS-Serverless-Anwendung	106
Erste Schritte	106
Lambda-Funktionen direkt aus dem Code ausführen und debuggen	115
Ausführen und Debuggen lokaler Amazon API Gateway Ressourcen	119
Konfigurationsoptionen für das Debuggen von serverless Anwendungen	123
Fehlerbehebung	131
AWS Systems Manager	133
Annahmen und Voraussetzungen	134
IAM-Berechtigungen für Dokumente zur Automatisierung von Systems Manager	134
Erstellen eines neuen Systems-Manager-Automation-dokuments	135
Öffnen eines bestehenden Systems-Manager-Automation-dokuments	136
Bearbeiten eines Systems-Manager-Automation-Dokuments	136
Veröffentlichen eines Dokuments zur Automatisierung von Systems Manager	137
Löschen eines Systems-Manager-Automation-Dokuments	138
Ausführen eines Systems-Manager-Automation-Dokuments	138
Fehlerbehebung	139
AWS Step Functions	140
AWS Step Functions und VS Code	140
Threat Composer	150
Arbeiten mit Threat Composer	150
Ressourcen	151
IAM-Berechtigungen für den Zugriff auf Ressourcen	152

Hinzufügen und Interaktion mit bestehenden Ressourcen	153
Erstellen und Bearbeiten von Ressourcen	154
Sicherheit	157
Datenschutz	157
Dokumentverlauf	159
.....	clxiv

AWS Toolkit for Visual Studio Code

Dies ist das Benutzerhandbuch für das AWS Toolkit for VS Code. Informationen zum AWS Toolkit for Visual Studio finden Sie im [Benutzerhandbuch für das AWS Toolkit for Visual Studio](#).

Was ist der AWS Toolkit for Visual Studio Code

Das Toolkit for VS Code ist eine Open-Source-Erweiterung für den Visual Studio Code (VS Code) -Editor. Diese Erweiterung erleichtert es Entwicklern, serverlose Anwendungen, die Amazon Web Services (AWS) verwenden, zu entwickeln, lokal zu debuggen und bereitzustellen.

Themen

- [Erste Schritte mit dem AWS Toolkit for Visual Studio Code](#)
- [Arbeiten mit AWS Diensten und Tools](#)

Ähnliche Informationen

Verwenden Sie die folgenden Ressourcen für den Zugriff auf den Quellcode für das Toolkit oder zum Anzeigen derzeit offener Probleme.

- [Quellcode](#)
- [Problemverfolgung](#)

Um mehr über den Visual Studio Code Editor zu erfahren, besuchen Sie <https://code.visualstudio.com/>.

Amazon Q Developer und Amazon CodeWhisperer

Seit dem 30. April 2024 CodeWhisperer ist Amazon jetzt Teil von Amazon Q Developer. Dazu gehören Inline-Codevorschläge und Amazon Q Developer-Sicherheitschecks. Laden Sie die [Amazon Q Developer IDE-Erweiterung vom VS Code Marketplace](#) herunter, um loszulegen.

Einzelheiten zum Amazon Q Developer Service finden Sie im [Amazon Q Developer](#) User Guide. Detaillierte Informationen zu Plänen und Preisen für Amazon Q finden Sie im [Amazon Q-Preisleitfaden](#).

Das Toolkit for VS Code herunterladen

Sie können es AWS Toolkit for Visual Studio Code über den VS Code Marketplace in Ihrer IDE herunterladen, installieren und einrichten. Ausführliche Anweisungen finden Sie im Abschnitt [Herunterladen und Installieren](#) im Thema Erste Schritte dieses Benutzerhandbuchs.

Das Toolkit vom VS Code Marketplace herunterladen

Alternativ können Sie die AWS Toolkit for Visual Studio Code Installationsdateien herunterladen, indem Sie von Ihrem Webbrowser aus zum [VS Code Marketplace](#) navigieren.

Zusätzliche IDE-Toolkits von AWS

Neben dem AWS Toolkit for Visual Studio Code bietet es AWS auch IDE-Toolkits für JetBrains und Visual Studio.

AWS Toolkit for JetBrainsLinks

- Folgen Sie diesem Link, um [das AWS Toolkit for JetBrains vom JetBrains Marketplace herunterzuladen](#).
- Weitere Informationen zu AWS Toolkit for JetBrains finden Sie im [AWS Toolkit for JetBrains](#) Benutzerhandbuch.

Toolkit for Visual Studio Studio-Links

- Folgen Sie diesem Link, um [das Toolkit for Visual Studio vom Visual Studio Marketplace herunterzuladen](#).
- Weitere Informationen zum Toolkit for Visual Studio finden Sie im [Toolkit for Visual Studio Studio-Benutzerhandbuch](#).

Erste Schritte mit dem AWS Toolkit for Visual Studio Code

Das AWS Toolkit for Visual Studio Code stellt Ihre AWS Dienste und Ressourcen direkt in Ihrer integrierten VS-Code-Entwicklungsumgebung (IDE) zur Verfügung.

Um Ihnen den Einstieg zu erleichtern, wird in den folgenden Themen beschrieben, wie Sie das einrichten, installieren und konfigurieren AWS Toolkit for Visual Studio Code.

Themen

- [Installieren des AWS Toolkit for Visual Studio Code](#)
- [Verbindung herstellen zu AWS](#)
- [AWS Regionen im Wandel](#)
- [Konfigurieren einer -Umgebung](#)

Installieren des AWS Toolkit for Visual Studio Code

Voraussetzungen

Um mit der Arbeit mit AWS Toolkit for Visual Studio Code VS Code beginnen zu können, müssen die folgenden Voraussetzungen erfüllt sein. Weitere Informationen zum Zugriff auf alle AWS Dienste und Ressourcen, die über den verfügbar sind AWS Toolkit for Visual Studio Code, finden Sie im [the section called "Optionale Voraussetzungen"](#) Abschnitt dieses Handbuchs.

- VS Code erfordert ein Windows-, MacOS- oder Linux-Betriebssystem.
- AWS Toolkit for Visual Studio Code Dazu müssen Sie mit VS Code Version 1.73.0 oder einer späteren Version arbeiten.

Weitere Informationen zu VS Code oder zum Herunterladen der neuesten Version von VS Code finden Sie auf der [VS Code-Download-Website](#).

Herunterladen und Installieren von AWS Toolkit for Visual Studio Code

Sie können es AWS Toolkit for Visual Studio Code über den VS Code Marketplace in Ihrer IDE herunterladen, installieren und einrichten. Alternativ können Sie die AWS Toolkit for Visual Studio Code Installationsdateien herunterladen, indem Sie von Ihrem Webbrowser aus zum [VS Code Marketplace](#) navigieren.

Installation AWS Toolkit for Visual Studio Code von aus dem VS Code IDE Marketplace

1. Öffnen Sie die AWS Toolkit for Visual Studio Code Erweiterung in Ihrer VS Code IDE mit dem folgenden Link: [Öffnen Sie den VS Code Marketplace](#).

Note

Wenn VS Code noch nicht auf Ihrem Computer ausgeführt wird, kann dieser Vorgang einige Augenblicke dauern, während VS Code geladen wird.

2. Wählen Sie in der AWS Toolkit for Visual Studio Code Erweiterung im VS Code Marketplace die Option Installieren aus, um den Installationsvorgang zu starten.
3. Wenn Sie dazu aufgefordert werden, wählen Sie, VS Code neu zu starten, um den Installationsvorgang abzuschließen.

Optionale Voraussetzungen

Um bestimmte Funktionen des AWS Toolkit for Visual Studio Code verwenden zu können, benötigen Sie Folgendes:

- Amazon Web Services (AWS) -Konto: Ein AWS Konto ist nicht erforderlich, um das zu verwenden AWS Toolkit for Visual Studio Code, aber die Funktionalität ist ohne dieses Konto erheblich eingeschränkt. Um ein AWS Konto zu erhalten, gehen Sie zur [AWS Startseite](#). Wählen Sie AWS Konto erstellen oder Registrierung abschließen (falls Sie die Website schon einmal besucht haben).
- Codeentwicklung — Das relevante SDK für die Sprache, die Sie verwenden möchten. Sie können von den folgenden Links herunterladen oder Ihren bevorzugten Paketmanager verwenden:
 - .NET SDK: <https://dotnet.microsoft.com/download>
 - Node.js SDK: <https://nodejs.org/en/download>
 - Python SDK: <https://www.python.org/downloads>
 - Java-SDK: <https://aws.amazon.com/corretto/>
 - Gehe zum SDK: <https://golang.org/doc/install>
- AWS SAM CLI — Dies ist ein AWS CLI-Tool, mit dem Sie Ihre serverlosen Anwendungen lokal entwickeln, testen und analysieren können. Dies ist für die Installation des Toolkits nicht erforderlich. Wir empfehlen jedoch, dass Sie es (und Docker, wie nachfolgend beschrieben)

installieren, da es für alle AWS Serverless Application Model (AWS SAM)-Funktionen erforderlich ist, z. B. [Eine neue serverlose Anwendung \(lokal\) erstellen](#).

Weitere Informationen finden Sie unter [Installieren der AWS SAM-CLI](#) im [AWS Serverless Application Model Entwicklerhandbuch](#).

- Docker — Die AWS SAM CLI benötigt diese Open-Source-Software-Container-Plattform. Weitere Informationen und Anweisungen zum Herunterladen finden Sie unter [Docker](#).
- Package Manager — Ein Paketmanager, mit dem Sie Anwendungscode herunterladen und teilen können.
 - .NET: [NuGet](#)
 - Node.js: [npm](#)
 - Python: [pip](#)
 - Java: [Gradle oder Maven](#)

Verbindung herstellen zu AWS

Die meisten Ressourcen von Amazon Web Services (AWS) werden über ein AWS Konto verwaltet. Für die Nutzung von ist kein AWS Konto erforderlich. Ohne Verbindung sind die AWS Toolkit for Visual Studio Code Funktionen des Toolkit jedoch eingeschränkt.

Wenn Sie zuvor ein AWS Konto eingerichtet und sich über einen anderen AWS Dienst (z. B. den AWS Command Line Interface) authentifiziert haben, erkennt der Ihre Anmeldeinformationen AWS Toolkit for Visual Studio Code automatisch.

Voraussetzungen

Wenn du noch kein Konto hast AWS oder noch kein Konto erstellt hast, gibt es drei Hauptschritte, um es AWS Toolkit for Visual Studio Code mit deinem AWS Konto zu verbinden:

1. Registrierung für ein AWS Konto: Du kannst dich über das [Anmeldeportal für ein AWS Konto AWS registrieren](#). Ausführliche Informationen zur Einrichtung eines neuen AWS Kontos finden Sie unter dem Thema „[Übersicht](#)“ im AWS Setup-Benutzerhandbuch.
2. Authentifizierung einrichten: Es gibt drei Hauptmethoden, um sich mit Ihrem AWS Konto über den AWS Toolkit for Visual Studio Code zu authentifizieren. Weitere Informationen zu den einzelnen Methoden finden Sie im Thema [Authentifizierung und Zugriff](#) in diesem Benutzerhandbuch.

3. Authentifizierung über das Toolkit: Sie können über das Toolkit eine Verbindung mit Ihrem AWS Konto herstellen, indem Sie die Verfahren in den folgenden Abschnitten dieses Benutzerhandbuchs ausführen. AWS

Öffnen Sie das Anmeldefenster

Führen Sie eines der folgenden Verfahren aus, um das AWS Toolkit-Anmeldefenster zu öffnen.

So öffnen Sie das AWS Toolkit-Anmeldefenster vom AWS Explorer aus:

1. Erweitern Sie im AWS Toolkit for Visual Studio Code EXPLORER.
2. Erweitern Sie die Option Weitere Aktionen... Menü durch Auswahl der... Symbol.
3. Aus den Weitere Aktionen... Wählen Sie im Menü Connect, AWS um das AWS Toolkit-Anmeldefenster zu öffnen.

So öffnen Sie das AWS Toolkit-Anmeldefenster mithilfe der VS Code-Befehlspalette:

1. Öffnen Sie die Befehlspalette, indem Sie **Shift+Command+P (Ctrl+Shift+P Windows)** drücken.
2. Geben Sie **AWS: Add a New Connection** in das Suchfeld ein.
3. Wählen Sie **AWS: Add a New Connection** diese Option, um das AWS Toolkit-Anmeldefenster zu öffnen.

Über das Toolkit AWS wird eine Verbindung hergestellt

Authentifizieren Sie sich und stellen Sie eine Verbindung mit SSO her

Gehen Sie wie folgt vor AWS IAM Identity Center, um sich zu authentifizieren und eine Verbindung AWS mit Using herzustellen.

Note

Durch die Authentifizierung mit AWS Builder ID oder IAM Identity Center wird das AWS Autorisierungsportal in Ihrem Standard-Webbrowser gestartet. Jedes Mal, wenn Ihre Anmeldeinformationen ablaufen, muss dieser Vorgang wiederholt werden, um die Verbindung zwischen Ihrem AWS Konto und dem AWS Toolkit for Visual Studio Code zu erneuern.

Authentifizieren Sie sich und stellen Sie eine Verbindung mit dem AWS IAM Identity Center her

1. Wählen Sie im AWS Toolkit-Anmeldebereich die Registerkarte Workforce und anschließend die Schaltfläche Weiter, um fortzufahren.
2. Geben Sie im Bereich Mit IAM Identity Center anmelden die Start-URL für Ihre Organisation ein. Diese URL wird Ihnen von einem Administrator oder Helpdesk in Ihrem Unternehmen zur Verfügung gestellt.
3. Wählen Sie Ihre AWS Region aus dem Drop-down-Menü aus. Dies ist die AWS Region, in der Ihr Identitätsverzeichnis gehostet wird.
4. Wählen Sie die Schaltfläche Weiter und bestätigen Sie, dass Sie die Website für AWS Autorisierungsanfragen in Ihrem Standard-Webbrowser öffnen möchten.
5. Folgen Sie den Anweisungen in Ihrem Standard-Webbrowser. Sie werden benachrichtigt, wenn der Autorisierungsvorgang abgeschlossen ist. Sie können Ihren Browser sicher schließen und zu VS Code zurückkehren.

Authentifizieren Sie sich und stellen Sie eine Verbindung mit IAM-Anmeldeinformationen her

Gehen Sie wie folgt vor, um sich AWS mithilfe von IAM-Anmeldeinformationen zu authentifizieren und eine Verbindung herzustellen.

Authentifizieren Sie sich und stellen Sie eine Verbindung mit IAM-Anmeldeinformationen her

1. Wählen Sie im AWS Toolkit-Anmeldebereich die Option IAM Credential aus und klicken Sie dann auf die Schaltfläche Weiter, um fortzufahren.
2. Geben Sie die **Profile Name**, **Access Key**, und **Secret Key** Ihres AWS Kontos in die dafür vorgesehenen Felder ein und klicken Sie dann auf die Schaltfläche Weiter, um das Profil zu Ihrer Konfigurationsdatei hinzuzufügen und das Toolkit mit Ihrem Konto zu verbinden. AWS
3. Der AWS Toolkit-Explorer wird aktualisiert und zeigt Ihre AWS Dienste und Ressourcen an, sobald die Authentifizierung abgeschlossen ist und eine Verbindung hergestellt wurde.

Authentifizierung für Amazon CodeCatalyst

Um mit CodeCatalyst der Arbeit über das Toolkit zu beginnen, authentifizieren Sie sich und stellen Sie eine Verbindung entweder mit Ihrer AWS Builder-ID oder Ihren IAM Identity Center-Anmeldeinformationen her.

In den folgenden Verfahren wird beschrieben, wie Sie das Toolkit authentifizieren und mit Ihrem Konto verbinden. AWS

Authentifizieren Sie sich und stellen Sie eine Verbindung mit einer Builder-ID her AWS

1. Wählen Sie im AWS Toolkit-Anmeldebereich die Registerkarte Workforce und dann die Schaltfläche Weiter, um fortzufahren.
2. Wählen Sie oben im Bereich Mit SSO anmelden den Link Direkt zur Anmeldung aus.
3. Folgen Sie den Anweisungen in Ihrem Standard-Webbrowser. Sie werden benachrichtigt, wenn der Autorisierungsvorgang abgeschlossen ist. Sie können Ihren Browser sicher schließen und zu VS Code zurückkehren.

Authentifizieren Sie sich und stellen Sie eine Verbindung mit IAM Identity Center her

1. Wählen Sie im AWS Toolkit-Anmeldebereich die Registerkarte Workforce und anschließend die Schaltfläche Weiter, um fortzufahren.
2. Geben Sie im Bereich Mit IAM Identity Center anmelden die Start-URL für Ihre Organisation ein. Diese URL wird Ihnen von einem Administrator oder Helpdesk in Ihrem Unternehmen zur Verfügung gestellt.
3. Wählen Sie Ihre AWS Region aus dem Drop-down-Menü aus. Dies ist die AWS Region, in der Ihr Identitätsverzeichnis gehostet wird.
4. Wählen Sie die Schaltfläche Weiter und bestätigen Sie, dass Sie die Website für AWS Autorisierungsanfragen in Ihrem Standard-Webbrowser öffnen möchten.
5. Folgen Sie den Anweisungen in Ihrem Standard-Webbrowser. Sie werden benachrichtigt, wenn der Autorisierungsvorgang abgeschlossen ist. Sie können Ihren Browser sicher schließen und zu VS Code zurückkehren.

AWS Regionen im Wandel

Eine AWS Region gibt an, wo Ihre AWS Ressourcen verwaltet werden. Ihre AWS Standardregion wird erkannt, wenn Sie über den eine Verbindung zu Ihrem AWS Konto herstellen. Dies wird automatisch im AWS Explorer angezeigt. AWS Toolkit for Visual Studio Code

In den folgenden Abschnitten wird beschrieben, wie Sie eine Region im AWS Explorer hinzufügen oder ausblenden.

Eine Region zum AWS Explorer hinzufügen

Gehen Sie wie folgt vor, um dem AWS Explorer eine Region hinzuzufügen.

1. Öffnen Sie in VS Code die Befehlspalette, indem Sie im Hauptmenü Ansicht erweitern und Befehlspalette auswählen. Oder verwenden Sie die folgenden Tastenkombinationen:
 - Windows und Linux — Drücken Sie **Ctrl+Shift+P**.
 - macOS — Drücken Sie **Shift+Command+P**.
2. Suchen Sie in der Befehlspalette nach Regionen ein **AWS: Show or Hide Regions** - AWS oder ausblenden und wählen Sie diese aus, um eine Liste der verfügbaren Regionen anzuzeigen.
3. Wählen Sie aus der Liste die AWS Regionen aus, die Sie dem AWS Explorer hinzufügen möchten.
4. Wählen Sie die Schaltfläche OK, um Ihre Auswahl zu bestätigen und den AWS Explorer zu aktualisieren.

Eine Region im AWS Explorer ausblenden

Gehen Sie wie folgt vor, um eine Region in der AWS Explorer-Ansicht auszublenden.

1. Suchen Sie im AWS Explorer nach der AWS Region, die Sie ausblenden möchten.
2. Öffnen Sie das Kontextmenü für die Region, die Sie ausblenden möchten (klicken Sie mit der rechten Maustaste darauf).
3. Wählen Sie Regionen ein- oder ausblenden, um die Optionen AWS: Regionen anzeigen oder ausblenden in VS Code zu öffnen.
4. Deaktivieren Sie in der AWS Explorer-Ansicht die Regionen, die Sie ausblenden möchten.

Konfigurieren einer -Umgebung

Das AWS Toolkit for Visual Studio Code unterstützt mehrere Sprachen für alle AWS Dienste. In den folgenden Abschnitten wird beschrieben, wie Sie Toolchain für verschiedene Sprachen

Konfiguration einer Toolchain für .NET Core

1. Stellen Sie sicher, dass Sie das AWS Toolkit for VS Code [installiert haben](#).

2. Installieren Sie die [C # -Erweiterung](#). Mit dieser -Umgebung
3. Öffnen Sie eine AWS Serverless Application Model-(AWS SAM)-Anwendung, oder [erstellen Sie eine](#).
4. Kopieren Sie den Ordner, der `template.yaml` enthält.

Konfigurieren einer -Umgebung Node.js

1. Stellen Sie sicher, dass Sie dasAWS Toolkit for VS Code [installiert haben](#).
2. Öffnen Sie eine AWS SAM-Anwendung, oder [erstellen Sie eine](#).
3. Kopieren Sie den Ordner, der `template.yaml` enthält.

Note

Wenn Sie eine TypeScript Lambda-Funktion direkt aus dem Quellcode heraus debuggen (Startkonfiguration hat `"target": "code"`), muss der TypeScript Compiler entweder global oder in dem Ihres Projekts installiert werden `package.json`.

Konfiguration einer Toolchain für Python

1. Stellen Sie sicher, dass Sie dasAWS Toolkit for VS Code [installiert haben](#).
2. Installieren Sie die [Python-Erweiterung für Visual Studio-Code](#). Diese Erweiterung ermöglicht es VS Code, Python-Anwendungen zu debuggen.
3. Öffnen Sie eine AWS SAM-Anwendung, oder [erstellen Sie eine](#).
4. Kopieren Sie den Ordner, der `template.yaml` enthält.
5. Öffnen Sie ein Terminal im Stammverzeichnis Ihrer Anwendung, und konfigurieren Sie `virtualenv`, indem Sie `python -m venv ./venv` ausführen.

Note

Sie müssen `virtualenv` nur einmal pro System konfigurieren.

6. Aktivieren Sie `virtualenv`, indem Sie einen der folgenden Codes ausführen:
 - Bash-Shell: `./venv/Scripts/activate`

- PowerShell: `./venv/Scripts/Activate.ps1`

Konfiguration einer Toolchain für Java

1. Stellen Sie sicher, dass Sie das AWS Toolkit for VS Code [installiert haben](#).
2. Installieren Sie die [Java-Erweiterung und Java 11](#). Diese Erweiterung ermöglicht es VS Code, Java-Funktionen zu erkennen.
3. Installieren Sie die [Java-Debugger-Erweiterung](#). Diese Erweiterung ermöglicht es VS Code, Java-Anwendungen zu debuggen.
4. Öffnen Sie eine AWS SAM-Anwendung, oder [erstellen Sie eine](#).
5. Kopieren Sie den Ordner, der `template.yaml` enthält.

Konfigurieren Sie eine Toolchain für Go

1. Stellen Sie sicher, dass Sie das AWS Toolkit for VS Code [installiert haben](#).
2. Go 1.14 oder höher ist für das Debuggen von Go Lambda-Funktionen erforderlich.
3. Installieren Sie die [Go-Erweiterung](#).

Note

Version 0.25.0 oder höher ist für das Debuggen von Go 1.15+-Laufzeiten erforderlich.

4. Installieren Sie die Go-Tools mithilfe der [Befehlspalette](#):
 - a. Wählen Sie in der Befehlspalette `Go: Install/Update Tools`.
 - b. Wählen Sie aus der Gruppe der Kontrollkästchen die Option `dlv` und `ausgopl`.
5. Öffnen Sie eine AWS SAM-Anwendung, oder [erstellen Sie eine](#).
6. Kopieren Sie den Ordner, der `template.yaml` enthält.

Verwenden Sie Ihre Toolchain

Sobald Sie Ihre Toolchain eingerichtet haben, können Sie sie verwenden, um die AWS SAM Anwendung [auszuführen oder zu debuggen](#).

Authentifizierung und Zugriff für das AWS Toolkit for Visual Studio Code

Sie müssen sich nicht bei authentifizieren, AWS um mit der Arbeit mit dem AWS Toolkit for Visual Studio Code zu beginnen. Die meisten AWS Ressourcen werden jedoch über ein AWS Konto verwaltet. Um auf alle AWS Toolkit for Visual Studio Code Dienste und Funktionen zugreifen zu können, müssen Sie sich mit einer AWS IAM Identity Center AWS Builder-ID oder IAM-Anmeldeinformationen authentifizieren.

Die folgenden Themen enthalten zusätzliche Informationen zu den einzelnen Anmeldeinformationstypen.

Einzelheiten dazu, wie Sie AWS Toolkit for Visual Studio Code mit Ihren vorhandenen Anmeldeinformationen eine Verbindung herstellen können, finden Sie im AWS Thema [Herstellen einer Verbindung zu](#) in diesem Benutzerhandbuch. [AWS](#)

Themen

- [AWS IAM-Identitätszentrum](#)
- [AWS IAM-Anmeldeinformationen](#)
- [AWS Builder-ID für Entwickler](#)
- [Verwenden eines Service für externe Anmeldeinformationen](#)

AWS IAM-Identitätszentrum

AWS IAM Identity Center ist die empfohlene bewährte Methode für die Verwaltung Ihrer AWS Kontoauthentifizierung.

Detaillierte Anweisungen zur Einrichtung von IAM Identity Center for Software Development Kits (SDKs) finden Sie im Abschnitt zur [IAM Identity Center-Authentifizierung im Referenzhandbuch](#) für AWS SDKs und Tools.

Einzelheiten dazu, wie Connect das AWS Toolkit authentifizieren und mit Ihren vorhandenen IAM Identity Center-Anmeldeinformationen verbinden können, finden Sie im AWS Thema [Verbinden mit](#) in diesem Benutzerhandbuch.

AWS IAM-Anmeldeinformationen

AWS Authentifizierung der IAM-Anmeldeinformationen mit Ihrem AWS Konto über lokal gespeicherte Zugriffsschlüssel.

Einzelheiten dazu, wie Connect das AWS Toolkit authentifizieren und mit Ihren vorhandenen AWS IAM-Anmeldeinformationen verbinden, finden Sie im AWS Thema [Verbinden](#) mit in diesem Benutzerhandbuch.

In den folgenden Abschnitten wird beschrieben, wie Sie IAM-Anmeldeinformationen einrichten, um sich mit Ihrem Konto über den zu authentifizieren. AWS AWS Toolkit for Visual Studio Code

Important

Bevor Sie IAM-Anmeldeinformationen für die Authentifizierung mit Ihrem AWS Konto einrichten, beachten Sie Folgendes:

- Wenn Sie IAM-Anmeldeinformationen bereits über einen anderen AWS Dienst (z. B. den AWS CLI) eingerichtet haben, erkennt der diese Anmeldeinformationen AWS Toolkit for Visual Studio Code automatisch und stellt sie in VS Code zur Verfügung.
- AWS empfiehlt die Verwendung der IAM Identity Center-Authentifizierung. Weitere Informationen zu Best Practices für AWS IAM finden Sie im Abschnitt [Bewährte Sicherheitsmethoden in IAM](#) im AWS Identity and Access Management-Benutzerhandbuch.
- Um Sicherheitsrisiken zu vermeiden, sollten Sie IAM-Benutzer nicht zur Authentifizierung verwenden, wenn Sie speziell entwickelte Software entwickeln oder mit echten Daten arbeiten. Verwenden Sie stattdessen den Verbund mit einem Identitätsanbieter wie [Was ist IAM Identity Center?](#) im AWS IAM Identity Center -Benutzerhandbuch.

Erstellen eines IAM-Benutzers

Bevor Sie das für die AWS Toolkit for Visual Studio Code Authentifizierung mit Ihrem AWS Konto einrichten können, müssen Sie Schritt 1: Erstellen Sie Ihren IAM-Benutzer und Schritt 2: Abrufen Ihrer Zugangsschlüssel im Thema [Authentifizieren mit langfristigen Anmeldeinformationen](#) im Referenzhandbuch für AWS SDKs und Tools abschließen.

Note

Schritt 3: Die Aktualisierung der Datei mit den gemeinsamen Anmeldeinformationen im Referenzhandbuch für AWS SDKs und Tools ist optional.

Wenn Sie Schritt 3 abgeschlossen haben, erkennt der Ihre Anmeldeinformationen AWS Toolkit for Visual Studio Code automatisch während der unten aufgeführten [the section called “Erstellen einer Datei mit gemeinsamen Anmeldeinformationen aus dem AWS Toolkit for Visual Studio Code”](#) Schritte.

Wenn Sie Schritt 3 noch nicht abgeschlossen haben, AWS Toolkit for Visual Studio Code werden Sie durch den Prozess der Erstellung eines `credentials` file geführt, wie im [the section called “Erstellen einer Datei mit gemeinsamen Anmeldeinformationen aus dem AWS Toolkit for Visual Studio Code”](#) Folgenden beschrieben.

Erstellen einer Datei mit gemeinsamen Anmeldeinformationen aus dem AWS Toolkit for Visual Studio Code

In Ihrer gemeinsamen Konfigurationsdatei und in der Datei mit gemeinsamen Anmeldeinformationen werden die Konfiguration und die Anmeldeinformationen für Ihre AWS Konten gespeichert. Weitere Informationen zu freigegebenen Konfigurationen und Anmeldeinformationen finden Sie im AWS Command Line Interface -Benutzerhandbuch im Abschnitt [Wo werden Konfigurationseinstellungen gespeichert?](#).

Erstellen einer gemeinsamen Anmeldeinformationsdatei über AWS Toolkit for Visual Studio Code

1. Öffnen Sie die Befehlspalette, indem Sie **Shift+Command+P (Ctrl+Shift+P)** drücken.
2. Geben Sie **AWS: Add a New Connection** in das Suchfeld ein.
3. Wählen Sie **AWS: Add a New Connection** diese Option, um das AWS Toolkit-Anmeldefenster zu öffnen.
4. Wählen Sie im Bereich AWS „Toolkit-Anmeldung“ die Option „IAM Credential“ und anschließend auf die Schaltfläche „Fortfahren“, um fortzufahren.
5. Geben Sie die **Profile Name**, **Access Key**, und **Secret Key** Ihres AWS Kontos in die dafür vorgesehenen Felder ein und klicken Sie dann auf die Schaltfläche Weiter, um das Profil zu Ihrer Konfigurationsdatei hinzuzufügen und das Toolkit mit Ihrem Konto zu verbinden. AWS

- Der AWS Toolkit-Explorer wird aktualisiert und zeigt Ihre AWS Dienste und Ressourcen an, sobald die Authentifizierung abgeschlossen ist und eine Verbindung hergestellt wurde.

Note

Gehen Sie in diesem Beispiel davon aus, dass `[Profile_Name]` Syntaxfehler enthält und die Authentifizierung fehlschlägt.

```
[Profile_Name]
xaws_access_key_id= AKIAI44QH8DHBEXAMPLE
xaws_secret_access_key= wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
```

Im Folgenden finden Sie ein Beispiel für eine Protokollnachricht, die als Reaktion auf einen fehlgeschlagenen Authentifizierungsversuch generiert wurde.

```
2022-11-02 22:01:54 [ERROR]: Profile [Profile_Name] is not a valid Credential
Profile: not supported by the Toolkit
2022-11-02 22:01:54 [WARN]: Shared Credentials Profile [Profile_Name] is not
valid. It will not be used by the toolkit.
```

Fügen Sie zusätzliche Anmeldeinformationsprofile hinzu

Sie können Ihren Konfigurationsdateien mehrere Anmeldeinformationen hinzufügen. Öffnen Sie dazu die Befehlspalette und wählen Sie AWS Toolkit Create Credentials Profile. Dadurch wird die Datei mit den Anmeldeinformationen geöffnet. Auf dieser Seite können Sie unter Ihrem ersten Profil ein neues Profil hinzufügen, wie im folgenden Beispiel gezeigt:

```
# Amazon Web Services Credentials File used by AWS CLI, SDKs, and tools
# This file was created by the AWS Toolkit for Visual Studio Code extension.
#
# Your AWS credentials are represented by access keys associated with IAM users.
# For information about how to create and manage AWS access keys for a user, see:
# https://docs.aws.amazon.com/IAM/latest/UserGuide/id_credentials_access-keys.html
#
# This credential file can store multiple access keys by placing each one in a
# named "profile". For information about how to change the access keys in a
```

```
# profile or to add a new profile with a different access key, see:
# https://docs.aws.amazon.com/cli/latest/userguide/cli-config-files.html
#
[Profile1_Name]
# The access key and secret key pair identify your account and grant access to AWS.
aws_access_key_id = AKIAIOSFODNN7EXAMPLE
# Treat your secret key like a password. Never share your secret key with anyone. Do
# not post it in online forums, or store it in a source control system. If your secret
# key is ever disclosed, immediately use IAM to delete the access key and secret key
# and create a new key pair. Then, update this file with the replacement key details.
aws_secret_access_key = wJalrXUtnFEMI/K7MDENG/bPxrFiCYEXAMPLEKEY
[Profile2_Name]
aws_access_key_id = AKIAI44QH8DHBEXAMPLE
aws_secret_access_key = je7MtGbClwBF/2Zp9Utk/h3yCo8nvbEXAMPLEKEY
```

AWS Builder-ID für Entwickler

Eine AWS Builder-ID ist ein zusätzliches AWS Konto, das für bestimmte AWS Dienste optional oder erforderlich ist. Ausführliche Informationen zur AWS Builder-ID-Authentifizierungsmethode finden Sie unter dem Thema [Mit AWS Builder-ID anmelden im AWS Anmelde-Benutzerhandbuch](#).

Einzelheiten dazu, wie Connect das AWS Toolkit authentifizieren und mit Ihrer vorhandenen AWS Builder-ID verbinden, finden Sie im AWS Thema [Verbinden](#) mit in diesem Benutzerhandbuch.

Verwenden eines Service für externe Anmeldeinformationen

Sie können das AWS Toolkit for Visual Studio Code für Anmeldeinformationsprozesse konfigurieren, die nicht direkt von unterstützt werden AWS, indem Sie Ihre ändern `shared config file`.

Das Ändern Ihrer Prozess `shared config file` für Anmeldeinformationen ist sowohl für die als auch für AWS Toolkit for Visual Studio Code die identisch AWS Command Line Interface. Ausführliche Informationen zum Einrichten externer Anmeldeinformationen finden Sie im AWS Command Line Interface Benutzerhandbuch unter dem Thema [Beschaffung von Anmeldeinformationen mit einem externen Prozess](#).

Arbeiten mit AWS Diensten und Tools

Das AWS Toolkit for Visual Studio Code stellt Ihnen AWS Dienste, Tools und Ressourcen direkt in VS Code zur Verfügung. Im Folgenden finden Sie eine Liste mit Leitfäden zu den einzelnen Toolkit for VS Code Code-Diensten und ihren Funktionen. Wählen Sie einen Dienst oder ein Tool aus, um weitere Informationen darüber zu erhalten, wie es funktioniert, wie es eingerichtet wird und wie Sie mit den grundlegenden Funktionen arbeiten.

Themen

- [Arbeiten mit experimentellen Funktionen](#)
- [Mit AWS Diensten im AWS Explorer arbeiten](#)
- [Amazon CodeCatalyst für VS Code](#)
- [Arbeiten mit Amazon API Gateway](#)
- [Verwenden AWS App Runner mit AWS Toolkit for Visual Studio Code](#)
- [AWSAnwendungs-Composer](#)
- [AWS CDKfür VS-Code](#)
- [Arbeiten mitAWS CloudFormationStapel](#)
- [Arbeiten mitCloudWatchLoggt mit demAWS Toolkit for Visual Studio Code](#)
- [Arbeiten mit Amazon Elastic Container Registry](#)
- [Arbeiten mit Amazon Elastic Container Service](#)
- [Arbeiten mit Amazon EventBridge](#)
- [AWS IAM-Zugriffsanalysator](#)
- [Arbeiten mit AWS IoT in der AWS Toolkit for Visual Studio Code](#)
- [Arbeiten mit AWS Lambda-Funktionen](#)
- [Amazon Redshift im Toolkit for VS Code](#)
- [Arbeiten mit Amazon S3](#)
- [Arbeiten mit serverlosen Anwendungen](#)
- [Arbeiten mit Systems-Manager-Automation-Dokumenten](#)
- [Arbeiten mit AWS Step Functions](#)
- [Mit Threat Composer arbeiten](#)

- [Arbeiten mit -Ressourcen](#)

Arbeiten mit experimentellen Funktionen

Experimentelle Funktionen bieten frühen Zugriff auf Funktionen im AWS Toolkit for Visual Studio Code bevor sie offiziell veröffentlicht werden.

Warning

Da experimentelle Funktionen weiterhin getestet und aktualisiert werden, können sie Probleme mit der Benutzerfreundlichkeit haben. Und experimentelle Funktionen können aus dem AWS Toolkit for Visual Studio Code ohne vorherige Ankündigung.

Sie können experimentelle Funktionen für bestimmte AWS-Services in der AWS Toolkit-Abschnitt im Einstellungen in Ihrer VS-Code-IDE.

1. So bearbeiten Sie AWS-Einstellungen in VS-Code, wählen Sie Datei, Präferenzen, Einstellungen.
2. In der Einstellungen-Fenster, erweitern Erweiterungen und wähle AWS Toolkit.
3. Under AWS: -Experimente die Kontrollkästchen für die experimentellen Funktionen, auf die Sie vor der Veröffentlichung zugreifen möchten. Wenn Sie eine experimentelle Funktion ausschalten möchten, deaktivieren Sie das entsprechende Kontrollkästchen.

Mit AWS Diensten im AWS Explorer arbeiten

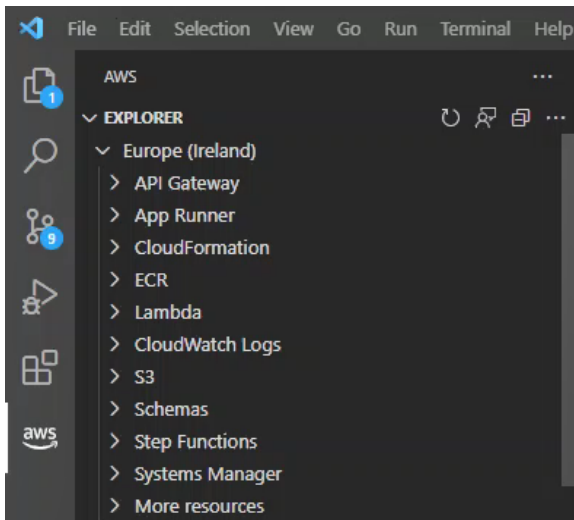
Der AWS Explorer bietet Ihnen eine Übersicht über einige AWS Dienste, mit denen Sie arbeiten können, wenn Sie den verwenden AWS Toolkit for Visual Studio Code.

Dieser Abschnitt enthält Informationen zum Zugriff auf und zur Verwendung des AWS Explorers in VS Code. Es wird davon ausgegangen, dass Sie das Toolkit for VS Code bereits auf Ihrem System [installiert und konfiguriert](#) haben.

Einige wichtige Punkte:

- Wenn das Toolkit korrekt installiert und konfiguriert ist, sollten Sie Elemente im Explorer sehen. Um den AWS Explorer zu öffnen, wählen Sie das AWS-Symbol in der Aktivitätsleiste.

Beispielsweise:



- Für bestimmte Funktionen sind bestimmte AWS Berechtigungen erforderlich. Um beispielsweise die AWS Lambda Funktionen in Ihrem AWS Konto zu sehen, [Authentifizierung und Zugriff](#) müssen die Anmeldeinformationen, in denen Sie konfiguriert haben, mindestens schreibgeschützte Lambda-Berechtigungen enthalten. Weitere Informationen zu den Berechtigungen, die die einzelnen Features benötigen, finden Sie in den folgenden Themen.
- Wenn Sie mit AWS Diensten interagieren möchten, die im AWS Explorer nicht sofort sichtbar sind, können Sie unter Weitere Ressourcen aus Hunderten von Ressourcen auswählen, die der Benutzeroberfläche hinzugefügt werden können.

Sie können beispielsweise AWS Toolkit:CodeArtifact: :Repository aus der Auswahl der verfügbaren Ressourcentypen auswählen. Nachdem dieser Ressourcentyp zu Weitere Ressourcen hinzugefügt wurde, können Sie den Eintrag erweitern, um eine Liste von Ressourcen anzuzeigen, die unterschiedliche CodeArtifact Repositorys mit ihren eigenen Eigenschaften und Attributen erstellen. Darüber hinaus können Sie die Eigenschaften und Attribute von Ressourcen in Vorlagen im JSON-Format beschreiben, die gespeichert werden können, um neue Ressourcen in der Cloud zu erstellen. AWS

Amazon CodeCatalyst für VS Code

Was ist Amazon CodeCatalyst?

Amazon CodeCatalyst ist ein cloudbasierter Ort für die Zusammenarbeit von Softwareentwicklungsteams. Über die AWS Toolkit for Visual Studio Code können Sie Ihre CodeCatalyst Ressourcen direkt von VS Code aus anzeigen und verwalten. Sie können auch direkt in der Cloud arbeiten, indem Sie das AWS Toolkit verwenden, um virtuelle Computerumgebungen von

Dev Environments zu starten, auf denen VS Code ausgeführt wird. Weitere Informationen zu diesem CodeCatalyst Service finden Sie im [CodeCatalystAmazon-Benutzerhandbuch](#).

In den folgenden Themen wird beschrieben CodeCatalyst, wie Sie VS Code mit dem Toolkit for VS Code verbinden und wie Sie damit arbeiten. CodeCatalyst

Themen

- [Erste Schritte mit CodeCatalyst und dem Toolkit for VS Code](#)
- [Arbeiten mit CodeCatalyst Amazon-Ressourcen in VS Code](#)
- [Arbeiten mit dem Toolkit in einer Entwicklungsumgebung](#)
- [Fehlerbehebung bei Amazon CodeCatalyst und VS Code](#)

Erste Schritte mit CodeCatalyst und dem Toolkit for VS Code

Gehen Sie wie folgt vor, um mit CodeCatalyst der Arbeit in VS Code zu beginnen.

Themen

- [Ein CodeCatalyst Konto erstellen](#)
- [Das AWS Toolkit verbinden mit CodeCatalyst](#)

Ein CodeCatalyst Konto erstellen

Sie benötigen eine aktive AWS Builder-ID oder AWS IAM Identity Center Anmeldeinformationen, um CodeCatalyst über das Toolkit for VS Code eine Verbindung herzustellen. Weitere Informationen zur AWS Builder-ID, zum IAM Identity Center und zu den CodeCatalyst Anmeldeinformationen finden Sie im CodeCatalyst Abschnitt [Einrichtung mit](#) im CodeCatalystBenutzerhandbuch.

Das AWS Toolkit verbinden mit CodeCatalyst

Informationen dazu, wie Sie das AWS Toolkit mit Ihrem CodeCatalyst Konto verbinden, finden Sie im CodeCatalyst Abschnitt [Authentifizierung für Amazon](#) im AWS Thema Verbindung mit diesem Benutzerhandbuch.

Arbeiten mit CodeCatalyst Amazon-Ressourcen in VS Code

Die folgenden Abschnitte bieten einen Überblick über die CodeCatalyst Amazon-Ressourcenverwaltungsfunktionen, die im Toolkit for VS Code verfügbar sind.

Weitere Informationen zu Entwicklungsumgebungen und dazu, wie Sie von dort aus darauf zugreifen können CodeCatalyst, finden Sie im Abschnitt [Entwicklungsumgebungen](#) im CodeCatalystAmazon-Benutzerhandbuch.

In den folgenden Abschnitten wird beschrieben, wie Sie Entwicklungsumgebungen aus VS Code erstellen, öffnen und mit ihnen arbeiten.

Themen

- [Klonen Sie ein Repository](#)
- [Öffnen einer Entwicklungsumgebung](#)
- [Eine CodeCatalyst Entwicklungsumgebung erstellen](#)
- [Erstellen einer Entwicklungsumgebung aus einem Repository eines Drittanbieters](#)
- [CodeCatalyst Befehle in VS Code](#)

Klonen Sie ein Repository

CodeCatalyst ist ein cloudbasierter Dienst, bei dem Sie mit der Cloud verbunden sein müssen, um an CodeCatalyst Projekten arbeiten zu können. Wenn Sie lieber lokal an einem Projekt arbeiten möchten, können Sie Ihre CodeCatalyst Repositories auf Ihren lokalen Computer klonen und sie online mit Ihrem CodeCatalyst Projekt synchronisieren, wenn Sie das nächste Mal mit der Cloud verbunden sind.

Gehen Sie wie folgt vor, um mit dem AWS Toolkit ein Repository von Ihrem CodeCatalyst Konto nach VS Code zu klonen:

Note

Wenn Sie ein Repository von einem Drittanbieterdienst klonen, werden Sie möglicherweise aufgefordert, sich mit den Anmeldeinformationen dieses Dienstes zu authentifizieren. Während das Repository geklont wird, zeigt VS Code den Fortschritt im Statusfenster zum Klonen des Repositories an. Nachdem das Repository geklont wurde, wird die Option *Möchten Sie das geklonte Repository öffnen?* eine Meldung wird angezeigt.

1. Erweitern Sie im Toolkit for VS Code den DEVELOPER TOOLS Explorer.
2. Erweitern Sie CodeCatalyst und wählen Sie Clone Repository.

3. Suchen Sie im Dialogfeld „CodeCatalyst Repository auswählen“ nach dem Repository, das Sie klonen möchten, und wählen Sie es dann aus, um das Dialogfeld „Ordner zum Klonen auswählen“ zu öffnen.
4. Wählen Sie „Repository-Speicherort auswählen“, um die Aufforderung zu schließen und mit dem Klonen des Repositories zu beginnen.
5. Wählen Sie im Dialogfenster eine der folgenden Optionen, um den Klonvorgang abzuschließen:
 - Um Ihr Repository in Ihrem aktuellen VS Code-Fenster zu öffnen, wählen Sie Öffnen.
 - Um Ihr Repository in einem neuen VS Code-Fenster zu öffnen, wählen Sie In neuem Fenster öffnen.
 - Um den Klonvorgang abzuschließen, ohne Ihr Repository zu öffnen, schließen Sie das Dialogfenster.

Öffnen einer Entwicklungsumgebung

Gehen Sie wie folgt vor, um eine vorhandene Entwicklungsumgebung in VS Code zu öffnen.

Note

Wenn Sie die Entwicklungsumgebung auswählen, wird der Vorgang zum Herstellen einer Verbindung mit VS Code gestartet, CodeCatalyst indem Sie Ihre Entwicklungsumgebung öffnen. Während dieses Vorgangs zeigt VS Code Fortschrittsaktualisierungen in einem CodeCatalyst Statusfenster an. Das Statusfenster wird aktualisiert, wenn der Vorgang abgeschlossen ist.

- Wenn die Entwicklungsumgebung nicht geöffnet werden kann, wird der Status mit Informationen darüber, warum der Prozess fehlgeschlagen ist, und einem Link zum Öffnen der Prozessprotokolle aktualisiert.
- Wenn der Vorgang erfolgreich ist, wird Ihre Entwicklungsumgebung in einem neuen Fenster von VS Code aus geöffnet.

1. Erweitern Sie im Toolkit for VS Code den DEVELOPER TOOLS Explorer.
2. Erweitern Sie CodeCatalyst und wählen Sie Entwicklungsumgebung öffnen, um das Dialogfeld CodeCatalystEntwicklungsumgebung auswählen in VS Code zu öffnen.

3. Wählen Sie im Dialogfeld „CodeCatalyst Entwicklungsumgebung auswählen“ die Entwicklungsumgebung aus, die Sie öffnen möchten.

Eine CodeCatalyst Entwicklungsumgebung erstellen

Gehen Sie wie folgt vor, um eine neue Entwicklungsumgebung zu erstellen:

Note

Beachten Sie beim Erstellen einer neuen Entwicklungsumgebung Folgendes:

- AWS empfiehlt die Angabe eines Alias, da dies die Organisation vereinfacht und die Suchmöglichkeiten für Entwicklungsumgebungen verbessert.
- Dev Environments speichert Ihre Arbeit dauerhaft. Das bedeutet, dass Ihre Entwicklungsumgebung gestoppt werden kann, ohne dass Ihre Arbeit verloren geht. Das Stoppen Ihrer Entwicklungsumgebung reduziert die Kosten, die erforderlich sind, um Ihre Entwicklungsumgebung am Laufen zu halten.
- Speicher ist die einzige Einstellung, die nach der Erstellung Ihrer Entwicklungsumgebung nicht geändert werden kann.
- VS Code zeigt den Fortschritt der Erstellung Ihrer Entwicklungsumgebung in einem Statusfenster an. Nachdem die Entwicklungsumgebung erstellt wurde, öffnet VS Code die Entwicklungsumgebung in einem neuen Fenster und die Option Vertrauen Sie den Autoren der Dateien in diesem Ordner? Eine Eingabeaufforderung erscheint ebenfalls. Stimmen Sie den Allgemeinen Geschäftsbedingungen zu, um weiterhin in Ihrer Entwicklungsumgebung arbeiten zu können.

1. Erweitern Sie im Toolkit for VS Code den DEVELOPER TOOLS Explorer.
2. Erweitern CodeCatalyst Sie und wählen Sie die Option Entwicklungsumgebung erstellen, um das Menü CodeCatalyst Entwicklungsumgebung erstellen in VS Code zu öffnen.
3. Wählen Sie im Abschnitt Quellcode eine der folgenden Optionen aus:
 - Ein vorhandenes CodeCatalyst Repository verwenden: Erstellt eine Entwicklungsumgebung aus einem vorhandenen CodeCatalyst Repository. Sie müssen das CodeCatalyst Projekt und den Zweig auswählen.
 - Eine leere Entwicklungsumgebung erstellen: Erstellt eine leere Entwicklungsumgebung.

4. (Optional) Geben Sie im Abschnitt Alias einen alternativen Namen für Ihre Entwicklungsumgebung ein.
5. (Optional) Ändern Sie im Abschnitt Konfiguration der Entwicklungsumgebungen die folgenden Einstellungen, um sie an Ihre spezifischen Anforderungen anzupassen.
 - Compute: Wählen Sie Edit Compute, um die Menge an Rechenleistung und RAM zu ändern, die Ihrem System zugewiesen sind.
 - Timeout: Wählen Sie „Timeout bearbeiten“, um die zulässige Leerlaufzeit des Systems zu ändern, bevor Ihre Entwicklungsumgebung gestoppt wird.
 - Speicher: Wählen Sie „Speichergröße bearbeiten“, um die Menge an Speicherplatz zu ändern, die Ihrem System zugewiesen ist.
6. Wählen Sie Create Dev Environment, um Ihre neue Cloud-Entwicklungsumgebung zu erstellen.

Erstellen einer Entwicklungsumgebung aus einem Repository eines Drittanbieters

Sie können Entwicklungsumgebungen aus einem Repository eines Drittanbieters erstellen, indem Sie auf das Repository als Quelle verlinken.

Das Verknüpfen mit einem Drittanbieter-Repository als Quelle erfolgt auf Projektebene in CodeCatalyst. Anweisungen und weitere Informationen dazu, wie Sie ein Drittanbieter-Repository mit Ihrer Entwicklungsumgebung verbinden, finden Sie im CodeCatalystAmazon-Benutzerhandbuch im Thema [Verknüpfen eines Quell-Repositorys](#).

CodeCatalyst Befehle in VS Code

Es gibt zusätzliche VS Code-Befehle, die verwandten Funktionen CodeCatalyst zugewiesen sind, die nicht direkt im AWS Toolkit angezeigt werden.

Gehen Sie wie folgt vor, um eine Liste der Befehle anzuzeigen, denen CodeCatalyst von der Befehlspalette aus zugewiesen wurden:

1. Erweitern Sie im Toolkit for VS Code den DEVELOPER TOOLS Explorer.
2. Wählen Sie CodeCatalyst Befehle anzeigen, um die Befehlspalette mit einer vorausgefüllten Suche nach zu öffnen. CodeCatalyst
3. Wählen Sie einen CodeCatalyst Befehl aus der Liste aus, um ihn zu aktivieren.

Arbeiten mit dem Toolkit in einer Entwicklungsumgebung

Entwicklungsumgebungen sind virtuelle Computerumgebungen für Amazon CodeCatalyst. In den folgenden Abschnitten wird beschrieben, wie Sie Entwicklungsumgebungen mithilfe von erstellen, starten und von dort aus arbeitenAWS Toolkit for Visual Studio Code.

Ausführliche Informationen zu Entwicklungsumgebungen finden Sie unter dem Thema [Entwicklungsumgebungen](#) im CodeCatalystAmazon-Benutzerhandbuch.

Konfiguration Ihrer Entwicklungsumgebung mit Devfiles

Die `devfile` Spezifikation ist ein offenes Standardformat für YAML, das verwendet werden kann, um Konfigurationen für Entwicklungsumgebungen zu definieren. Jede Entwicklungsumgebung hat eine Dev-Datei. Wenn Sie eine Entwicklungsumgebung ohne Projektarchiv oder aus einem Projektarchiv erstellen, das keine Entwicklungsdatei enthält, wird automatisch ein Standard auf die Quelle angewendet. Devfiles können von CodeCatalyst oder von Ihrer IDE aus aktualisiert werden. Die Prozesse zum Aktualisieren einer Dev-Datei in einer lokalen oder Remote-Instanz von VS Code sind identisch, aber wenn Sie eine Dev-Datei lokal aktualisieren, müssen Sie die Updates in Ihr Quell-Repository übertragen, bevor die Updates wirksam werden.

Ausführliche Informationen zur Konfiguration von Entwicklungsumgebungen mit Devfiles finden Sie unter dem Thema [Konfiguration Ihrer Entwicklungsumgebung](#) im CodeCatalystAmazon-Benutzerhandbuch.

Das folgende Verfahren beschreibt, wie Sie Ihre Dev-Datei von einer Remote-Instanz des Toolkits aus bearbeiten können, während es in einer Entwicklungsumgebung läuft.

Important

Beachten Sie Folgendes, wenn Sie den `Devfile` From VS Code bearbeiten:

- Wenn Sie den Namen der Dev-Datei oder den Namen der Devfile-Komponente ändern, wird der Inhalt Ihres Stammverzeichnisses ersetzt. Der gesamte vorherige Inhalt ist verloren gegangen und kann nicht wiederhergestellt werden.
- Wenn Sie eine Entwicklungsumgebung ohne eine Dev-Datei im Stammordner oder eine Entwicklungsumgebung erstellen, die keinem Quell-Repository zugeordnet ist, wird bei der Erstellung eine Dev-Datei mit den Standardkonfigurationseinstellungen für Ihre Entwicklungsumgebung generiert.

- Anweisungen zur Definition und Konfiguration Ihrer `Devfile` finden Sie in der Dokumentation zum [Hinzufügen von Befehlen](#) auf der [devfile.io-Website](#).

1. Erweitern Sie im Toolkit for VS Code den DEVELOPER TOOLS Explorer.
2. Erweitern Sie CodeCatalyst und wählen Sie Devfile öffnen, um es `devfile.yaml` in einem neuen Editorfenster in Ihrer aktuellen Entwicklungsumgebung zu öffnen.
3. Aktualisieren Sie im VS Code-Editor Ihre Dev-Datei und speichern Sie dann Ihre Änderungen.
4. Wenn Sie Ihre Entwicklungsumgebung das nächste Mal starten, wird die Konfiguration aktualisiert, sodass sie den Spezifikationen entspricht, die in Ihrer `Devfile` definiert sind.

Authentifizierung und Verbindung zu AWS Ihrer Entwicklungsumgebung

Um von Ihrer Entwicklungsumgebung aus auf all Ihre AWS Ressourcen zuzugreifen, müssen Sie sich authentifizieren und Ihre Remote-Instanz des Toolkits mit Ihrem Konto verbinden. AWS Die Remote-Instanz des Toolkits authentifiziert sich automatisch mit den Anmeldeinformationen, die Sie von Ihrer lokalen Instanz des Toolkits übernommen haben, wenn Ihre Entwicklungsumgebung gestartet wird.

Die Verfahren zur Aktualisierung Ihrer Anmeldeinformationen für eine Remote-Instanz des Toolkits sind identisch mit der Authentifizierung in Ihrer lokalen Instanz des Toolkits. Ausführliche Anweisungen zum Aktualisieren der Anmeldeinformationen, zur Authentifizierung und zum AWS Herstellen einer Verbindung über das Toolkit finden Sie im AWS Abschnitt Herstellen einer [Verbindung zu](#) im Thema Erste Schritte in diesem Benutzerhandbuch.

Weitere Informationen zu den einzelnen AWS Authentifizierungsmethoden, die mit dem kompatibel sind AWS Toolkit for Visual Studio Code, finden Sie im Thema [Authentifizierung und Zugriff](#) in diesem Benutzerhandbuch.

Arbeiten mit dem Toolkit for VS Code in Entwicklungsumgebungen

Nachdem Sie eine Entwicklungsumgebung in VS Code geöffnet oder erstellt haben, können Sie mit dem Toolkit for VS Code arbeiten, ähnlich wie Sie es von einer lokalen Instanz von VS Code aus tun können. Entwicklungsumgebungen, auf denen VS Code ausgeführt wird, sind so konfiguriert, dass das AWS Toolkit automatisch installiert und eine Verbindung mit Ihrer AWS Builder-ID hergestellt wird.

Anhalten einer Entwicklungsumgebung

So beenden Sie Ihre aktuelle Entwicklungsumgebung:

1. Erweitern Sie im Toolkit for VS Code den DEVELOPER TOOLS Explorer.
2. Erweitern Sie CodeCatalyst und wählen Sie Stop Dev Environment.
3. Wenn Sie von VS Code dazu aufgefordert werden, bestätigen Sie, dass Sie Ihre Entwicklungsumgebung beenden möchten.
4. Ihre Entwicklungsumgebung wurde erfolgreich gestoppt, wenn VS Code die Remoteverbindung schließt und zu einer lokalen Entwicklungsinstantz zurückkehrt.

Öffnen der Einstellungen für die Entwicklungsumgebung

Gehen Sie wie folgt vor, um die Einstellungen für Ihre aktuelle Entwicklungsumgebung zu öffnen:

Note

Sie können den Speicherplatz, der Ihrer Entwicklungsumgebung zugewiesen wurde, nicht ändern, nachdem sie erstellt wurde.

1. Erweitern Sie im Toolkit for VS Code den DEVELOPER TOOLS Explorer.
2. Erweitern Sie CodeCatalyst und wählen Sie Einstellungen öffnen, um die Ansicht mit den Einstellungen für die Entwicklungsumgebung für Ihre aktuelle Entwicklungsumgebung zu öffnen.
3. Die folgenden Abschnitte in der Ansicht Dev Environment Settings (Entwicklungsumgebungseinstellungen) enthalten Optionen für Ihre Entwicklungsumgebung:
 - Alias: Zeigen Sie den Alias an, der Ihrer Entwicklungsumgebung zugewiesen ist, und ändern Sie ihn.
 - Status: Sehen Sie sich Ihren aktuellen Status der Entwicklungsumgebung und das Projekt an, dem sie zugewiesen ist, und beenden Sie Ihre Umgebung.
 - Devfile: Zeigt den Namen und den Speicherort von `Devfile` für Ihre Entwicklungsumgebung an. Öffnen Sie Ihre `Devfile` indem Sie auf die Schaltfläche Im Editor öffnen klicken.
 - Compute Settings (Einstellungen berechnen): Ändern Sie die Größe und den Standardwert für Timeout Length (Timeout-Länge) für Ihre Entwicklungsumgebung.

Fehlerbehebung bei Amazon CodeCatalyst und VS Code

Die folgenden Themen behandeln mögliche technische Probleme bei der Arbeit mit Amazon CodeCatalyst und VS Code.

Themen

- [VS-Code-Version](#)
- [Berechtigungen für Amazon CodeCatalyst](#)
- [Über das Toolkit for VS Code eine Verbindung zu einer Entwicklungsumgebung herstellen](#)

VS-Code-Version

Es wird erwartet, dass Ihre Version von VS Code einen Handler für `vscode://` URIs auf Ihrem System einrichtet. Ohne diesen Handler können Sie nicht auf alle CodeCatalyst Funktionen des AWS Toolkits zugreifen. Beispielsweise tritt beim Starten einer Entwicklungsumgebung von VS Code Insiders ein Fehler auf. Dies liegt daran, dass VS Code Insiders `vscode-insiders://` URIs verarbeitet und keine URIs. `vscode://`

Berechtigungen für Amazon CodeCatalyst

Im Folgenden sind die Anforderungen an Dateiberechtigungen für die Arbeit mit CodeCatalyst folgenden Dateien aufgeführtAWS Toolkit for Visual Studio Code:

- Legen Sie Ihre eigenen Zugriffsberechtigungen für Ihre `~/ .ssh/config` Datei auf `read` und `festwrite`. Beschränken `write` Sie die Berechtigungen für alle anderen Benutzer.
- Legen Sie Ihre Zugriffsberechtigungen für die `~/ .ssh/id_rsa` Dateien `~/ .ssh/id_dsa` und auf „`readNur`“ fest. Einschränken `read write` und `execute` Berechtigungen für alle anderen Benutzer.
- Ihre `globals.context.globalStorageUri.fsPath` Datei muss sich an einem beschreibbaren Speicherort befinden.

Über das Toolkit for VS Code eine Verbindung zu einer Entwicklungsumgebung herstellen

Wenn Sie beim Versuch, eine Verbindung zu einer Entwicklungsumgebung herzustellen, die AWS Toolkit for Visual Studio Code folgende Fehlermeldung erhalten:

Ihr `~/.ssh/config` hat einen `aws-devenv-*` Abschnitt, der möglicherweise veraltet ist.

- Wählen Sie die Konfiguration öffnen. Schaltfläche, um Ihre `~/.ssh/config` Datei im VS Code Editor zu öffnen.
- Wählen Sie im Editor den Inhalt des Host `aws-devenv-*` Abschnitts aus und löschen Sie ihn.
- Speichern Sie die Änderungen, die Sie am Host `aws-devenv-*` von vorgenommen haben `~/.ssh/config`. Schließen Sie dann die Datei.
- Versuchen Sie erneut, über das Toolkit for VS Code eine Verbindung zu einer Entwicklungsumgebung herzustellen.

Arbeiten mit Amazon API Gateway

Sie können Remote-API-Gateway-Ressourcen in Ihrem verbundenen AWS-Konto unter Verwendung des AWS Toolkit for Visual Studio Code aus.

Note

Diese Funktion unterstützt das Debuggen nicht.

So durchsuchen und führen Sie Remote-API-Gateway-Ressourcen aus

1. In der AWS Explorer, wählen API Gateway um das Menü zu erweitern. Die Remote-API-Gateway-Ressourcen werden aufgeführt.
2. Suchen Sie die API Gateway Gateway-Ressource, die Sie aufrufen möchten, öffnen Sie das Kontextmenü (rechte Maustaste) und wählen Sie dann Aufrufen von am AWS aus.
3. Geben Sie im Parameterformular die Aufnahmeparameter an.
4. Um die Remote-API-Gateway-Ressource auszuführen, wählen Sie Aufrufen aus. Die Ergebnisse werden im VS-Code-AusgabeAnzeigen von n

Verwenden AWS App Runner mit AWS Toolkit for Visual Studio Code

[AWS App Runner](#) bietet eine schnelle, einfache und kostengünstige Möglichkeit, Quellcode oder ein Container-Image direkt in einer skalierbaren und sicheren Webanwendung in der AWS Cloud

bereitzustellen. Damit müssen Sie sich nicht mit neuen Technologien vertraut machen, entscheiden, welchen Rechen dienst Sie verwenden möchten, oder wissen, wie AWS Ressourcen bereitgestellt und konfiguriert werden.

Sie können AWS App Runner es verwenden, um Dienste auf der Grundlage eines Quellimages oder Quellcodes zu erstellen und zu verwalten. Wenn Sie ein Quellimage verwenden, können Sie ein öffentliches oder privates Container-Image auswählen, das in einem Image-Repository gespeichert ist. App Runner unterstützt die folgenden Image-Repository-Anbieter:

- Amazon Elastic Container Registry (Amazon ECR): Speichert private Bilder in Ihrem AWS Konto.
- Amazon Elastic Container Registry Öffentlich (Amazon ECR öffentlich): Speichert öffentlich lesbare Images.

Wenn Sie die Option Quellcode wählen, können Sie von einem Quellcode-Repository aus bereitstellen, das von einem unterstützten Repository-Anbieter verwaltet wird. Derzeit wird App Runner [GitHub](#) als Quellcode-Repository-Anbieter unterstützt.

Voraussetzungen

Für die Interaktion mit App Runner mithilfe von AWS Toolkit for Visual Studio Code ist Folgendes erforderlich:

- Ein AWS Konto
- Eine Version AWS Toolkit for Visual Studio Code dieser Funktionen AWS App Runner

Stellen Sie zusätzlich zu diesen grundlegenden Anforderungen sicher, dass alle relevanten IAM-Benutzer über Berechtigungen zur Interaktion mit dem App-Runner-Service verfügen. Außerdem benötigen Sie spezifische Informationen über Ihre Servicequelle, z. B. die Container-Image-URI oder die Verbindung zum GitHub Repository. Diese Informationen benötigen Sie beim Erstellen Ihres App-Runner-Services.

Konfigurieren von IAM-Berechtigungen für App Runner

Der einfachste Weg, die für App Runner erforderlichen Berechtigungen zu gewähren, besteht darin, der entsprechenden AWS Identity and Access Management (IAM-) Entität, insbesondere einem Benutzer oder einer Gruppe, eine bestehende AWS verwaltete Richtlinie zuzuweisen. App Runner bietet zwei verwaltete Richtlinien, die Sie an Ihre IAM-Benutzer anfügen können:

- `AWSAppRunnerFullAccess`: Ermöglicht Benutzern, alle App-Runner-Aktionen auszuführen.
- `AWSAppRunnerReadOnlyAccess`: Ermöglicht Benutzern das Auflisten und Anzeigen von Details zu App-Runner-Ressourcen.

Wenn Sie ein privates Repository aus der Amazon Elastic Container Registry (Amazon ECR) als Servicequelle auswählen, müssen Sie außerdem die folgende Zugriffsrolle für Ihren App-Runner-Service erstellen:

- `AWSAppRunnerServicePolicyForECRAccess`: Ermöglicht App Runner den Zugriff auf Amazon Elastic Container Registry (Amazon ECR)-Images in Ihrem Konto.

Sie können diese Rolle automatisch erstellen, wenn Sie Ihre Dienstinstantz mit der Befehlspalette von VS Code konfigurieren.

Note

Die mit dem `AWSAppRunnerServiceRoleForAppRunnerDienst` verknüpfte Rolle `AWS App Runner` ermöglicht die Ausführung der folgenden Aufgaben:

- Übertragen Sie Protokolle an Amazon CloudWatch Logs-Protokollgruppen.
- Erstellen Sie Amazon CloudWatch Events-Regeln, um Amazon Elastic Container Registry (Amazon ECR) Image-Push zu abonnieren.

Sie müssen die serviceverknüpfte Rolle nicht manuell erstellen. Wenn Sie eine `AWS App Runner` in der `AWS Management Console` oder mithilfe von `API-Operationen` erstellen, die von aufgerufen werden `AWS Toolkit for Visual Studio Code`, `AWS App Runner` erstellt diese serviceverknüpfte Rolle für Sie.

Weitere Informationen finden Sie unter [Identity and Access Management für App Runner](#) im `AWS App Runner -Entwicklerhandbuch`.

Abrufen von Servicequellen für App Runner

Sie können `AWS App Runner` verwenden, um Dienste aus einem Quellbild oder Quellcode bereitzustellen.

Source image

Wenn Sie die Bereitstellung von einem Quell-Image aus durchführen, können Sie einen Link zum Repository für dieses Image aus einer privaten oder öffentlichen AWS Image-Registry abrufen.

- Private Amazon-ECR-Registrierung: Kopieren Sie den URI für ein privates Repository, das die Amazon-ECR-Konsole verwendet, unter <https://console.aws.amazon.com/ecr/repositories>.
- Öffentliche Amazon-ECR-Registrierung: Kopieren Sie den URI für ein öffentliches Repository, das die Amazon ECR Public Gallery verwendet, unter <https://gallery.ecr.aws/>.

Note

Sie können den URI für ein privates Amazon ECR-Repository auch direkt vom AWS Explorer im Toolkit for VS Code abrufen:

- Öffnen Sie den AWS Explorer und erweitern Sie den ECR-Knoten, um die Liste der Repositories für diese Region anzuzeigen. AWS
- Klicken Sie mit der rechten Maustaste auf das Repository und wählen Sie Copy Repository URI (Repository-URI kopieren), um den Link in die Zwischenablage zu kopieren.

Sie geben den URI für das Image-Repository an, wenn Sie Ihre Service-Instance mit der Befehlspalette von VS Code konfigurieren

Weitere Informationen finden Sie unter [App Runner service based on a source image](#) (App-Runner-Service basierend auf einem Quellimage) im AWS App Runner -Entwicklerhandbuch.

Source code

Damit Ihr Quellcode für einen AWS App Runner Service bereitgestellt werden kann, muss dieser Code in einem Git-Repository gespeichert werden, das von einem unterstützten Repository-Anbieter verwaltet wird. App Runner unterstützt einen Quellcode-Repository-Anbieter: [GitHub](#).

Informationen zum Einrichten eines GitHub Repositories finden Sie in der [Dokumentation Erste Schritte](#) unter GitHub.

Um Ihren Quellcode aus einem GitHub Repository für einen App Runner-Dienst bereitzustellen, stellt App Runner eine Verbindung zu her GitHub. Wenn Ihr Repository privat ist (das heißt, es ist

nicht öffentlich zugänglich GitHub), müssen Sie App Runner Verbindungsdetails zur Verfügung stellen.

Important

Um GitHub Verbindungen herzustellen, müssen Sie die App Runner-Konsole (<https://console.aws.amazon.com/apprunner>) verwenden, um eine Verbindung herzustellen, die GitHub auf verweist AWS. Sie können die Verbindungen auswählen, die auf der GitHub Verbindungsseite verfügbar sind, wenn Sie Ihre Dienstinanz mit der Befehlspalette von VS Code konfigurieren.

Weitere Informationen finden Sie unter [Verwalten von App-Runner-Verbindungen](#) im AWS App Runner -Entwicklerhandbuch.

Die App Runner-Dienstinanz bietet eine verwaltete Laufzeit, mit der Ihr Code erstellt und ausgeführt werden kann. AWS App Runner unterstützt derzeit die folgenden Laufzeiten:

- Mit Python verwaltete Laufzeit
- Mit Node.js verwaltete Laufzeit

Im Rahmen Ihrer Servicekonfiguration geben Sie Informationen darüber an, wie der App-Runner-Service Ihren Service aufbaut und startet. Sie können diese Informationen über die Command Palette (Befehlspalette) eingeben oder eine YAML-formatierte [App Runner configuration file](#) (App-Runner-Konfigurationsdatei) angeben. Werte in dieser Datei weisen App Runner an, wie Sie Ihren Service erstellen und starten und einen Laufzeitkontext bereitstellen. Dies umfasst relevante Netzwerkeinstellungen und Umgebungsvariablen. Die Konfigurationsdatei hat den Namen `apprunner.yaml`. Es wird automatisch zum Stammverzeichnis des Repositorys Ihrer Anwendung hinzugefügt.

Preisgestaltung

Ihnen werden die Computing- und Speicherressourcen in Rechnung gestellt, die Ihre Anwendung verwendet. Wenn Sie Ihre Bereitstellungen automatisieren, zahlen Sie außerdem eine festgelegte monatliche Gebühr für jede Anwendung, die alle automatisierten Bereitstellungen für diesen Monat abdeckt. Wenn Sie sich für die Bereitstellung aus Quellcode entscheiden, zahlen Sie zusätzlich eine

Entwicklungsgebühr für die Zeit, die App Runner benötigt, um einen Container aus Ihrem Quellcode zu entwickeln.

Weitere Informationen finden Sie unter [AWS App Runner -Preisgestaltung](#).

Themen

- [Erstellen von App-Runner-Services](#)
- [Verwalten von App-Runner-Services](#)

Erstellen von App-Runner-Services

Sie können einen App Runner-Dienst in Toolkit for VS Code erstellen, indem Sie den AWS Explorer und die Befehlspalette von VS Code verwenden. Nachdem Sie sich entschieden haben, einen Dienst in einer bestimmten AWS Region zu erstellen, führen Sie die nummerierten Schritte in der Befehlspalette durch den Prozess der Konfiguration der Dienstinstanz, auf der Ihre Anwendung ausgeführt wird.

Stellen Sie vor dem Erstellen eines App-Runner-Services sicher, dass Sie die [Voraussetzungen](#) erfüllen. Dies umfasst die Bereitstellung der relevanten IAM-Berechtigungen und die Bestätigung des spezifischen Quell-Repositorys, das Sie bereitstellen möchten.

So erstellen Sie einen App-Runner-Service

1. Öffnen Sie den AWS Explorer, falls er nicht bereits geöffnet ist.
2. Klicken Sie mit der rechten Maustaste auf den Knoten App Runner und wählen Sie Create Service (Service erstellen) aus.

Die Befehlspalette wird angezeigt.

3. Wählen Sie für Select a source code location type (Quellcode-Speicherorttyp auswählen) die Option ECR oder Repository aus.

Wenn Sie ECR wählen, geben Sie ein Container-Image in einem Repository an, das von Amazon Elastic Container Registry verwaltet wird. Wenn Sie Repository wählen, geben Sie ein Quellcode-Repository an, das von einem unterstützten Repository-Anbieter verwaltet wird. Derzeit wird App Runner [GitHub](#) als Quellcode-Repository-Anbieter unterstützt.

Bereitstellen von ECR

1. Wählen Sie für **Select or enter an image repository** (Image-Repository auswählen oder eingeben) die URL des Image-Repositorys aus, das von Ihrer privaten Amazon-ECR-Registrierung oder der Amazon ECR Public Gallery verwaltet wird, oder geben Sie sie ein.

Note

Wenn Sie ein Repository aus der Amazon ECR Public Gallery angeben, stellen Sie sicher, dass automatische Bereitstellungen deaktiviert sind, da App Runner keine automatischen Bereitstellungen für ein Image in einem öffentlichen ECR-Repository unterstützt.

Automatische Bereitstellungen sind standardmäßig ausgeschaltet. Dies wird angezeigt, wenn das Symbol in der Kopfzeile der Befehlspalette von einer diagonalen Linie durchzogen ist. Wenn Sie sich für das Einschalten automatischer Bereitstellungen entschieden haben, werden Sie in einer Meldung darüber informiert, dass diese Option zusätzliche Kosten verursachen kann.

2. Wenn im Schritt „Befehlspalette“ die Meldung **Keine Tags gefunden** angezeigt wird, müssen Sie einen Schritt zurückgehen und ein Repository auswählen, das ein mit Tags versehenes Container-Image enthält.
3. Wenn Sie eine private Amazon ECR-Registrierung verwenden, benötigen Sie die ECR-Zugriffsrolle **ECR, AppRunner** die App Runner den Zugriff auf Amazon Elastic Container Registry (Amazon ECR) -Images in Ihrem Konto ermöglicht. **AccessRole** Wählen Sie das „+“ -Symbol in der Kopfzeile der Befehlspalette, um diese Rolle automatisch zu erstellen. (Eine Zugriffsrolle ist nicht erforderlich, wenn Ihr Image in Amazon ECR Public gespeichert wird, wo Images öffentlich verfügbar sind.)
4. Geben Sie für **Port** den IP-Port ein, der vom Service verwendet wird (z. B. **Port 8000**).
5. Für **Configure environment variables** (Umgebungsvariablen konfigurieren) können Sie eine Datei angeben, die Umgebungsvariablen enthält, die zum Anpassen des Verhaltens in Ihrer Service-Instance verwendet werden. Oder Sie können diesen Schritt überspringen.
6. Geben Sie für **Name your service** (Ihren Service benennen) einen eindeutigen Namen ohne Leerzeichen ein und drücken Sie **Enter** (Eingabe).
7. Wählen Sie für **Select instance configuration** (Instance-Konfiguration auswählen) eine Kombination aus CPU-Einheiten und Arbeitsspeicher in GB für Ihre Service-Instance aus.

Wenn Ihr Service erstellt wird, ändert sich sein Status von Creating (Wird erstellt) in Running (Wird ausgeführt).

8. Nachdem Ihr Service gestartet wurde, klicken Sie mit der rechten Maustaste darauf und wählen Sie Copy Service URL (Service-URL kopieren).
9. Fügen Sie für den Zugriff auf die bereitgestellte Anwendung die kopierte URL in die Adressleiste Ihres Webbrowsers ein.

Bereitstellen von einem Remote-Repository

1. Wählen Sie unter Verbindung auswählen eine Verbindung aus, GitHub zu der eine Verknüpfung besteht AWS. Die Verbindungen, die zur Auswahl stehen, sind auf der GitHub Verbindungsseite der App Runner-Konsole aufgeführt.
2. Wählen Sie unter Wählen Sie ein GitHub Remote-Repository aus, oder geben Sie eine URL für das Remote-Repository ein.

Remote-Repositorys, die bereits mit dem Source Control Management (SCM) von Visual Studio Code konfiguriert sind, stehen zur Auswahl. Sie können auch einen Link zum Repository einfügen, wenn er nicht aufgeführt ist.

3. Wählen Sie für Select a branch (Eine Verzweigung auswählen), welche Git-Verzweigung Ihres Quellcodes Sie bereitstellen möchten.
4. Geben Sie für Choose configuration source (Konfigurationsquelle auswählen) an, wie Sie die Laufzeitkonfiguration definieren möchten.

Wenn Sie Use configuration file (Konfigurationsdatei verwenden) angeben, wird Ihre Service-Instance durch Einstellungen konfiguriert, die durch die `apprunner.yaml`-Konfigurationsdatei definiert sind. Diese Datei befindet sich im Stammverzeichnis des Repositorys Ihrer Anwendung.

Wenn Sie Alle Einstellungen hier konfigurieren wählen, verwenden Sie die Befehlspalette, um Folgendes anzugeben:

- Runtime (Laufzeit): Wählen Sie Python 3 oder Nodejs 12 aus.
 - Build command (Entwicklungsbefehl): Geben Sie den Befehl ein, um Ihre Anwendung in der Laufzeitumgebung Ihrer Service-Instance zu erstellen.
 - Start command (Startbefehl): Geben Sie den Befehl ein, um Ihre Anwendung in der Laufzeitumgebung Ihrer Service-Instance zu starten.
5. Geben Sie für Port den IP-Port ein, der vom Service verwendet wird (z. B. Port 8000).

6. Für Configure environment variables (Umgebungsvariablen konfigurieren) können Sie eine Datei angeben, die Umgebungsvariablen enthält, die zum Anpassen des Verhaltens in Ihrer Service-Instance verwendet werden. Oder Sie können diesen Schritt überspringen.
7. Geben Sie für Name your service (Ihren Service benennen) einen eindeutigen Namen ohne Leerzeichen ein und drücken Sie Enter (Eingabe).
8. Wählen Sie für Select instance configuration (Instance-Konfiguration auswählen) eine Kombination aus CPU-Einheiten und Arbeitsspeicher in GB für Ihre Service-Instance aus.

Wenn Ihr Service erstellt wird, ändert sich sein Status von Creating (Wird erstellt) in Running (Wird ausgeführt).

9. Nachdem Ihr Service gestartet wurde, klicken Sie mit der rechten Maustaste darauf und wählen Sie Copy Service URL (Service-URL kopieren).
10. Fügen Sie für den Zugriff auf die bereitgestellte Anwendung die kopierte URL in die Adressleiste Ihres Webbrowsers ein.

Note

Wenn Ihr Versuch zum Erstellen eines App-Runner-Service fehlschlägt, zeigt der Service im AWS -Explorer den Status Create failed (Erstellen fehlgeschlagen) an. Tipps zur Fehlerbehebung finden Sie unter [Wenn die Service-Erstellung fehlschlägt](#) im Entwicklerhandbuch zu App Runner.

Verwalten von App-Runner-Services

Nachdem Sie einen App Runner-Dienst erstellt haben, können Sie ihn verwalten, indem Sie den AWS Explorer-Bereich verwenden, um die folgenden Aktivitäten auszuführen:

- [Anhalten und Fortsetzen von App-Runner-Services](#)
- [Bereitstellen von App-Runner-Services](#)
- [Anzeigen von Protokoll-Streams für App Runner](#)
- [Löschen von App-Runner-Services](#)

Anhalten und Fortsetzen von App-Runner-Services

Wenn Sie Ihre Webanwendung vorübergehend deaktivieren und die Ausführung des Codes beenden müssen, können Sie Ihren AWS App Runner-Dienst anhalten. App Runner reduziert die Rechenkapazität für den Service auf Null. Wenn Sie bereit sind, Ihre Anwendung erneut auszuführen, setzen Sie Ihren App-Runner-Service fort. App Runner stellt neue Rechenkapazitäten bereit, stellt Ihre Anwendung bereit und führt die Anwendung aus.

Important

App Runner wird Ihnen nur in Rechnung gestellt, wenn er ausgeführt wird. Daher können Sie Ihre Bewerbung nach Bedarf anhalten und fortsetzen, um die Kosten zu verwalten. Dies ist besonders hilfreich bei Entwicklungs- und Testszenarien.

So unterbrechen Sie Ihren App-Runner-Service

1. Öffnen Sie den AWS Explorer, falls er nicht bereits geöffnet ist.
2. Erweitern Sie App Runner, um die Liste der Services anzuzeigen.
3. Klicken Sie mit der rechten Maustaste auf Ihren Service und wählen Sie Pause aus.
4. Wählen Sie im daraufhin angezeigten Dialogfeld die Option Confirm (Bestätigen) aus.

Während der Service pausiert, ändert sich der Servicestatus von Running (Wird ausgeführt) zu Pausing (Wird pausiert) und dann zu Paused (Pausiert).

So setzen Sie Ihren App-Runner-Service wieder fort

1. Öffnen Sie den AWS Explorer, falls er nicht bereits geöffnet ist.
2. Erweitern Sie App Runner, um die Liste der Services anzuzeigen.
3. Klicken Sie mit der rechten Maustaste auf Ihren Service und wählen Sie Resume (Fortsetzen) aus.

Während der Service wieder aufgenommen wird, ändert sich der Servicestatus von Resuming (Wird fortgesetzt) zu Running (Wird ausgeführt).

Bereitstellen von App-Runner-Services

Wenn Sie die Option für die manuelle Bereitstellung für Ihren Service wählen, müssen Sie jede Bereitstellung explizit für Ihren Service initiieren.

1. Öffnen Sie den AWS Explorer, falls er nicht bereits geöffnet ist.
2. Erweitern Sie App Runner, um die Liste der Services anzuzeigen.
3. Klicken Sie mit der rechten Maustaste auf Ihren Service und wählen Sie Start Deployment (Bereitstellung starten) aus.
4. Während Ihre Anwendung bereitgestellt wird, ändert sich der Servicestatus von Deploying (Wird bereitgestellt) zu Running (Wird ausgeführt).
5. Um zu bestätigen, dass Ihre Anwendung erfolgreich bereitgestellt wurde, klicken Sie mit der rechten Maustaste auf denselben Service und wählen Sie Copy Service URL (Service-URL kopieren) aus.
6. Fügen Sie für den Zugriff auf die bereitgestellte Webanwendung die kopierte URL in die Adressleiste Ihres Webbrowsers ein.

Anzeigen von Protokoll-Streams für App Runner

Verwenden Sie CloudWatch Logs, um Ihre Log-Streams für Dienste wie App Runner zu überwachen, zu speichern und darauf zuzugreifen. Ein Protokollstream ist eine Abfolge von Protokollereignissen, die dieselbe Quelle nutzen.

1. Erweitern Sie App Runner, um die Liste der Service-Instances anzuzeigen.
2. Erweitern Sie eine bestimmte Service-Instance, um die Liste der Protokollgruppen anzuzeigen. (Eine Protokollgruppe ist eine Gruppe von Protokollstreams, die dieselben Einstellungen für die Aufbewahrung, Überwachung und Zugriffskontrolle besitzen.)
3. Klicken Sie mit der rechten Maustaste auf eine Protokollgruppe und wählen Sie View Log Streams (Protokollstreams anzeigen) aus.
4. Wählen Sie in der Befehlspalette einen Log-Stream aus der Gruppe aus.

Der VS Code-Editor zeigt die Liste der Protokollereignisse an, aus denen der Stream besteht. Sie können wählen, ob ältere oder neuere Ereignisse in den Editor geladen werden sollen.

Löschen von App-Runner-Services

Important

Wenn Sie Ihren App-Runner-Service löschen, wird er dauerhaft entfernt und Ihre gespeicherten Daten werden gelöscht. Wenn Sie den Service neu erstellen müssen, muss App Runner Ihre Quelle erneut abrufen und erstellen, wenn es sich um ein Code-Repository handelt. Ihre Webanwendung erhält eine neue App-Runner-Domain.

1. Öffnen Sie den AWS Explorer, falls er nicht bereits geöffnet ist.
2. Erweitern Sie App Runner, um die Liste der Services anzuzeigen.
3. Klicken Sie mit der rechten Maustaste auf einen Service und wählen Sie Delete Service (Service löschen) aus.
4. Geben Sie in der Befehlspalette Löschen ein und drücken Sie dann zur Bestätigung die Eingabetaste.

Der gelöschte Service zeigt den Status Deleting (Wird gelöscht). Anschließend verschwindet der Service aus der Liste.

AWSAnwendungs-Composer

Sie können den verwendenAWS Toolkit for Visual Studio Code, um mit dem AWS Application Composer-Dienst zu arbeiten. AWS Application Composer ist ein Visual Builder für AWS Anwendungen, der Sie beim Entwerfen Ihrer Anwendungsarchitektur und bei der Visualisierung Ihrer AWS CloudFormation Infrastruktur unterstützt.

Ausführliche Informationen zum AWS Application Composer-Dienst finden Sie im [AWSApplication Composer-Benutzerhandbuch](#).

In den folgenden Themen wird beschrieben, wie Sie mit AWS Application Composer aus dem arbeitenAWS Toolkit for Visual Studio Code.

Themen

- [Arbeiten mit AWS Application Composer aus dem Toolkit](#)

Arbeiten mit AWS Application Composer aus dem Toolkit

Mit Application Composer für AWS Toolkit for Visual Studio Code können Sie Anwendungen über eine interaktive Leinwand visuell entwerfen. Sie können Application Composer auch verwenden, um AM-Templates zu visualisieren, AWS CloudFormation und AWS Serverless Application Model zu ändern. Während der Arbeit mit Application Composer werden Ihre Änderungen dauerhaft gespeichert, sodass Sie nahtlos zwischen der Bearbeitung von Dateien direkt im VS Code-Editor oder mithilfe der interaktiven Leinwand wechseln können.

Ausführliche Informationen zum AWS Application Composer-Dienst, Informationen zu den ersten Schritten und Tutorials finden Sie im [AWS Application Composer-Dienst-Benutzerhandbuch](#).

In den folgenden Abschnitten wird beschrieben, wie Sie über den auf den AWS Application Composer-Dienst zugreifen können AWS Toolkit for Visual Studio Code.

Zugriff auf AWS Application Composer über das Toolkit

Es gibt drei Hauptmethoden, wie Sie vom Toolkit aus auf AWS Application Composer zugreifen können.

Von einer vorhandenen Vorlage aus auf AWS Application Composer zugreifen

1. Öffnen Sie in VS Code eine vorhandene Vorlagendatei im VS Code-Editor.
2. Klicken Sie im Editorfenster auf die Schaltfläche AWS Application Composer in der oberen rechten Ecke des Editorfensters.
3. AWS Application Composer öffnet und visualisiert Ihre Vorlagendatei im VS Code Editor-Fenster.

Zugriff auf AWS Application Composer über das Kontextmenü (Rechtsklick)

1. Klicken Sie in VS Code mit der rechten Maustaste auf die Vorlagendatei, die Sie mit AWS Application Composer öffnen möchten.
2. Wählen Sie im Kontextmenü die Option Mit App Composer öffnen.
3. AWS Application Composer öffnet und visualisiert Ihre Vorlagendatei in einem neuen VS Code-Editor-Fenster.

Über die Befehlspalette auf AWS Application Composer zugreifen

1. Öffnen Sie in VS Code die Befehlspalette, indem Sie **Cmd + Shift + P** oder **Ctrl + Shift + P** (Windows) drücken
2. Geben Sie im Suchfeld AWSApplication Composer ein **AWS Application Composer** und wählen Sie es aus, wenn es in den Ergebnissen angezeigt wird.
3. Wählen Sie die Vorlagendatei aus, die Sie öffnen möchten. AWS Application Composer öffnet und visualisiert Ihre Vorlagendatei in einem neuen VS Code-Editor-Fenster.

AWS CDK für VS-Code

Dies ist die Vorabdokumentation einer Funktion, die als Vorversion vorliegt. Änderungen sind vorbehalten.

Die AWS CDK Bedienung ermöglicht Ihnen die Arbeit mit [AWS Cloud Development Kit \(AWS CDK\)](#) Anwendungen, oder Apps aus. Ausführliche Informationen zum AWS CDK im [AWS Cloud Development Kit \(AWS CDK\) Entwicklerhandbuch](#) aus.

AWS CDK-Apps bestehen aus Bausteinen, den so genannten [Konstrukten](#), die Definitionen für Ihre AWS CloudFormation-Stacks und die AWS-Ressourcen darin enthalten. Mit dem AWS CDK-Explorer können Sie die [Stacks](#) und [Ressourcen](#) visualisieren, die in AWS CDK-Konstrukten definiert sind. Diese Visualisierung wird in einer Baumstruktur-Ansicht im Visual Studio Code (VS-Code) -Editor.

In diesem Abschnitt finden Sie Informationen dazu, wie Sie zugreifen und diese verwenden AWS CDK im VS-Code-Editor. Es wird davon ausgegangen, dass Sie bereits [installiert und konfiguriert](#) das Toolkit for VS Code für Ihre lokale IDE.

Themen

- [Arbeiten mit AWS CDK Anwendungen](#)

Arbeiten mit AWS CDK Anwendungen

Dies ist die Vorabdokumentation einer Funktion, die als Vorversion vorliegt. Änderungen sind vorbehalten.

Verwenden der AWS CDK Explorer im AWS Toolkit for VS Code zum Visualisieren und Arbeiten mit AWS CDK-Anwendungen.

Voraussetzungen

- Stellen Sie sicher, dass Ihr System die Voraussetzungen erfüllt, die unter angegeben sind [Installieren des Toolkit for VS Code](#) aus.
- Installieren des AWS CDK-Befehlszeilenschnittstelle, wie in den ersten Abschnitten von [Erste Schritte mit der AWS CDK](#) im AWS Cloud Development Kit (AWS CDK) Entwicklerhandbuch aus.

Important

Ihre AWS CDK-Version muss 1.17.0 oder höher sein. Verwenden Sie auf der Befehlszeile `cdk --version`, um zu sehen, welche Version Sie ausführen.

Visualisieren einer AWS CDK-Anwendung

Verwendung der AWS Toolkit for VS Code AWS CDK Explorer können Sie das verwalten [Stapel](#) und [Ressourcen](#) die in den CDK-Konstrukten Ihrer Apps gespeichert sind. Die AWS CDK Der Explorer zeigt Ihre Ressourcen in einer Baumansicht mit den Informationen an, die im `tree.json`-Datei, die erstellt wird, wenn Sie `cdk synth` befehlen. Die `tree.json` befindet sich unter `cdk.out`-Verzeichnis, standardmäßig.

Einstieg in die Verwendung des Toolkit AWS CDK Explorer, müssen Sie eine CDK-Anwendung erstellen.

1. Führen Sie die ersten Schritte des [Hello World-Tutorial](#) befindet sich unter [AWS CDK Entwicklerhandbuch](#) aus.

Important

Wenn Sie den Tutorialschritt erreicht haben Bereitstellen des Stacks, hören Sie auf und kehren Sie zu dieser Anleitung zurück.

Note

Sie können die Befehle, die im Lernprogramm bereitgestellt werden, z. B. **mkdir** und **cdk init**, auf einer Betriebssystembefehlszeile oder in einer Terminal-Fenster im VS-Code-Editor.

2. Nachdem Sie die erforderlichen Schritte des CDK-Tutorials abgeschlossen haben, öffnen Sie den CDK-Inhalt, den Sie im VS-Code-Editor erstellt haben.
3. Aus AWS erweitern Sie im Navigationsbereich CDK (Vorschau) personalisieren. Ihre CDK-Anwendungen und die zugehörigen Ressourcen werden jetzt in der CDK Explorer-Strukturansicht angezeigt.

Wichtige Hinweise

- Wenn Sie CDK-Apps in den VS Code -Editor laden, können Sie mehrere Ordner gleichzeitig laden. Jeder Ordner kann wie im vorherigen Bild dargestellt mehrere CDK-Apps enthalten. Der AWS CDK-Explorer findet Apps im Projektstammverzeichnis und dessen direkten Unterverzeichnissen.
- Wenn Sie die ersten Schritte des Lernprogramms ausführen, stellen Sie möglicherweise fest, dass der letzte Befehl, den Sie ausführen, **cdk synth** ist, der die `tree.json`-Datei generiert. Wenn Sie Aspekte einer CDK-App ändern, z. B. weitere Ressourcen hinzufügen, müssen Sie diesen Befehl erneut ausführen, um die Änderungen in der Strukturansicht anzuzeigen.

Führen Sie andere Operationen auf einem AWS CDK App

Sie können den VS Code -Editor verwenden, um andere Operationen in einer CDK-App auszuführen, so wie Sie es mit der Befehlszeile Ihres Betriebssystems oder anderen Tools tun würden. Beispielsweise können Sie die Code-Dateien im -Editor aktualisieren und die -App mithilfe eines VS Code bereitstellen Terminal-Fenster.

Um diese Arten von Aktionen auszuprobieren, verwenden Sie den VS Code -Editor, um [Hello World-Tutorial](#) im AWS CDK Entwicklerhandbuch aus. Stellen Sie sicher, dass Sie den letzten Schritt ausführen, Die Ressourcen der App zerstören, damit Sie keine unerwarteten Kosten für Ihre AWS-Konto.

Arbeiten mit AWS CloudFormation Stapel

Das AWS Toolkit for Visual Studio Code bietet Unterstützung für [AWS CloudFormation](#)-Stacks. Mit dem Toolkit for VS Code können Sie bestimmte Aufgaben mit AWS CloudFormation-Stacks, z. B. das Löschen dieser Stacks.

Themen

- [Löschen einer AWS CloudFormation Stapeln](#)
- [Erstellen Sie eine AWS CloudFormation Vorlage mit dem AWS Toolkit for Visual Studio Code](#)

Löschen einer AWS CloudFormation Stapeln

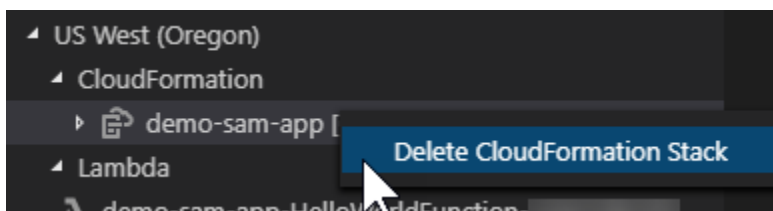
Sie können AWS CloudFormation-Stacks mit AWS Toolkit for Visual Studio Code löschen.

Voraussetzungen

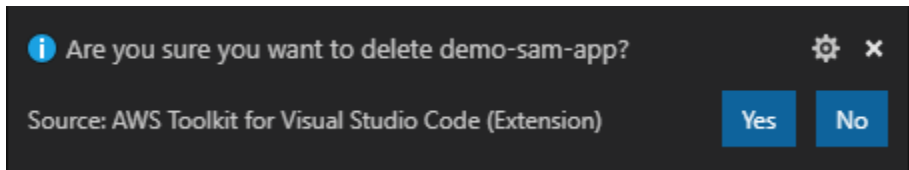
- Stellen Sie sicher, dass Ihr System die Voraussetzungen erfüllt, die unter [Installieren des Toolkit for VS Code](#) aus.
- Stellen Sie sicher, dass die Anmeldeinformationen, die Sie in [Authentifizierung und Zugriff](#) konfiguriert haben, den entsprechenden Lese-/Schreibzugriff auf den AWS CloudFormation-Service enthalten. Wenn im AWS Explorer unter CloudFormation als Meldung in etwa „Fehler beim Laden von CloudFormation-Ressourcen“ angezeigt wird, überprüfen Sie die Berechtigungen, die diesen Anmeldeinformationen zugeordnet sind. Änderungen, die Sie an Berechtigungen vornehmen, dauern einige Minuten, bis sie sich auf AWS Explorer im VS-Code.

Löschen eines CloudFormation-Stacks

1. Öffnen Sie im AWS Explorer das Kontextmenü des AWS CloudFormation-Stacks, den Sie löschen möchten.



2. Wählen Sie Delete CloudFormation-Stack (CloudFormation Stack löschen).
3. Wählen Sie in der angezeigten Meldung Yes (Ja) aus, um das Löschen zu bestätigen.



Nachdem der Stack gelöscht wurde, wird er nicht mehr im AWS Explorer aufgeführt.

Erstellen Sie eine AWS CloudFormation Vorlage mit dem AWS Toolkit for Visual Studio Code

Sie AWS Toolkit for Visual Studio Code können Ihnen beim Schreiben AWS CloudFormation und bei SAM-Vorlagen behilflich sein.

Voraussetzungen

Toolkit for VS Code und Voraussetzungen für Anmeldeinformationen

- Bevor Sie über das Toolkit for VS Code auf den CloudFormation Service zugreifen können, müssen Sie die in der Bedienungsanleitung zur [Installation des Toolkit for VS Code](#) beschriebenen Anforderungen erfüllen.
- Die Anmeldeinformationen, die Sie in erstellt haben, [Authentifizierung und Zugriff](#) müssen den entsprechenden Lese-/Schreibzugriff auf den Dienst beinhalten. AWS CloudFormation

Note

Wenn der CloudFormationDienst die Meldung Fehler beim Laden der CloudFormation Ressourcen anzeigt, überprüfen Sie die Berechtigungen, die Sie diesen Anmeldeinformationen zugewiesen haben. Beachten Sie auch, dass es einige Minuten dauern kann, bis Änderungen an Berechtigungen im AWS Explorer aktualisiert werden.

CloudFormation Voraussetzungen für Vorlagen

- Installieren und aktivieren Sie die [Redhat Developer YAML VS Code-Erweiterung](#).
- Sie müssen mit dem Internet verbunden sein, wenn Sie die Redhat Developer YAML VS Code-Erweiterung verwenden, da sie zum Herunterladen und Einlösen von JSON-Schemas auf Ihren Computer verwendet wird.

Eine CloudFormation Vorlage mit YAML Schema Support schreiben

Das Toolkit verwendet YAML-Sprachunterstützung und JSON-Schemas, um den Schreibprozess und SAM-Vorlagen zu optimieren. CloudFormation Funktionen wie Syntaxvalidierung und Autovervollständigung beschleunigen nicht nur den Prozess, sondern tragen auch zur Verbesserung der Qualität Ihrer Vorlage bei. Bei der Auswahl eines Schemas für Ihre Vorlage werden die folgenden bewährten Methoden empfohlen.

CloudFormation Vorlage

- Die Datei hat die Erweiterung .yaml oder .yml.
- Die Datei hat einen Knoten der obersten Ebene oder einen Ressourcenknoten **AWSTemplateFormatVersion**.

SAM-Vorlage

- Alle bereits beschriebenen Kriterien für CloudFormation
- Die Datei hat einen Transform-Knoten der obersten Ebene, der einen Wert enthält, der mit `AWS::Serverless` beginnt.

Das Schema wird bei der Änderung der Datei angewendet. Beispielsweise wird ein SAM-Vorlagenschema angewendet, nachdem einer CloudFormation Vorlage eine serverlose Transformation hinzugefügt und die Datei gespeichert wurde.

Überprüfung der Syntax

Die YAML-Erweiterung wendet automatisch die Typvalidierung auf Ihre Vorlage an. Dadurch werden Einträge mit ungültigen Typen für eine bestimmte Eigenschaft hervorgehoben. Wenn Sie den Mauszeiger über einen markierten Eintrag bewegen, zeigt die Erweiterung Korrekturmaßnahmen an.

Automatische Vervollständigung

Wenn Sie neue Felder, Aufzählungswerte oder andere [Ressourcentypen](#) hinzufügen, können Sie die Autovervollständigungsfunktion der YAML-Erweiterung starten, indem Sie Strg + Leertaste eingeben.

Arbeiten mit CloudWatch Logs mit dem AWS Toolkit for Visual Studio Code

Amazon CloudWatch Logs ermöglicht es Ihnen, die Protokolle aller Ihrer Systeme, Anwendungen und AWS-Dienste, die Sie verwenden, in einem einzigen, hoch skalierbaren Service. Sie können sie dann einfach anzeigen, nach bestimmten Fehlercodes oder Mustern suchen, sie anhand bestimmter Felder filtern oder sicher für zukünftige Analysen archivieren. Weitere Informationen finden Sie unter [Was ist Amazon CloudWatch Protokolle?](#) in der Amazon CloudWatch Benutzerleitfaden.

Die folgenden Themen beschreiben die Verwendung von AWS Toolkit for Visual Studio Code um damit zu arbeiten CloudWatch Logs sich ein AWS-Konto.

Themen

- [Anzeigen von CloudWatch-Protokollgruppen und Protokoll-Streams mithilfe der AWS Toolkit for Visual Studio Code](#)
- [Arbeiten mit CloudWatch Protokollieren von -Protokollstreams mithilfe des AWS Toolkit for Visual Studio Code](#)
- [CloudWatch Log-Gruppen durchsuchen](#)

Anzeigen von CloudWatch-Protokollgruppen und Protokoll-Streams mithilfe der AWS Toolkit for Visual Studio Code

Ein Protokollstream ist eine Abfolge von Protokollereignissen, die dieselbe Quelle nutzen. Jede separate Quelle für CloudWatch Logs bildet einen separaten Protokollstream.

Eine Protokollgruppe ist eine Gruppe von Protokollstreams, die dieselben Einstellungen für die Aufbewahrung, Überwachung und Zugriffskontrolle besitzen. Sie können Protokollgruppen definieren und angeben, welche Streams in welche Gruppe geschickt werden sollen. Es gibt keine Begrenzung dazu, wie viele Protokoll-Streams zu einer Protokollgruppe gehören können.

Weitere Informationen finden Sie unter [Arbeiten mit Protokollgruppen und Protokoll-Streams](#) im Amazon CloudWatch Benutzerhandbuch.

Themen

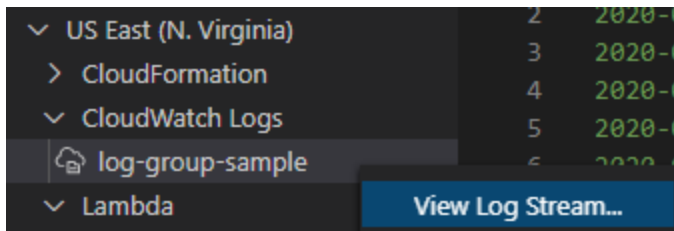
- [Anzeigen von Protokollgruppen und Protokoll-Streams mit dem CloudWatch-Logs-Knoten](#)

Anzeigen von Protokollgruppen und Protokoll-Streams mit dem CloudWatch-Logs-Knoten

1. Wählen Sie im VS-CodeAnzeigen,ExplorerSo öffnen Sie denAWS-Explorer.
2. Klicken Sie auf das Symbol CloudWatch-Logs-Knoten, um die Liste der Protokollgruppen zu erweitern.

Die Protokollgruppen für die aktuelle AWS-Region wird unter der CloudWatch-Logs-Knoten.

3. Um die Protokollströme in einer Protokollgruppe anzuzeigen, klicken Sie mit der rechten Maustaste auf den Namen der Protokollgruppe, und wählen SieAnzeigen Protokoll-Streamaus.



4. Aus-BefehlspaletteWählen Sie einen Protokoll-Stream aus der anzuzeigenden Gruppe aus.

Note

Die-Befehlspalettezeigt einen Zeitstempel für das letzte Ereignis in jedem Stream an.

Die[Log Stream \(Protokollstream\)Editor](#)startet, um die Protokollereignisse des Streams anzuzeigen.

Arbeiten mit CloudWatch Protokollieren von -Protokollstreams mithilfe desAWS Toolkit for Visual Studio Code

Nachdem Sie das geöffnet habenProtokollieren Steam-Editor können Sie auf die Protokollereignisse in jedem Stream zugreifen. Ein Protokollereignis ist ein Datensatz von einigen Aktivitäten, der von der überwachten Anwendung oder Ressource aufgezeichnet wird.

Themen

- [Log-Stream-Informationen anzeigen und kopieren](#)
- [Speichern des Inhalts des Protokolldatenstam-Editors in einer lokalen Datei](#)

Log-Stream-Informationen anzeigen und kopieren

Wenn Sie einen Log-Stream öffnen, wird der Log Stream (Protokollstream)-Editor zeigt die Sequenz der Protokollereignisse dieses Streams an.

1. Öffnen Sie den Protokollstream, der angezeigt werden soll, den Log Stream (Protokollstream) - Editor (siehe [Anzeigen von CloudWatch-Protokollgruppen und Protokoll-Streams](#)) enthalten.

Jede Zeile, die ein Ereignis auflistet, wird mit einem Zeitstempel versehen, um anzuzeigen, wann es protokolliert wurde.

2. Verwenden Sie die folgenden Optionen, um Informationen zu den Ereignissen des Streams anzuzeigen und zu kopieren:
 - Ereignisse nach Zeit anzeigen: Zeigen Sie die neuesten und älteren Protokollereignisse an, indem Sie **Laden Sie Neuere Ereignisse** oder **Ältere Ereignisse laden**.

Note

Die Log Stream (Protokollstream) lädt zunächst einen Stapel der letzten 10.000 Zeilen von Protokollereignissen oder 1 MB Protokolldaten (je nachdem, was kleiner ist). Wenn Sie **Laden Sie neuere Ereignisse** wählen, zeigt der Editor Ereignisse an, die nach dem Laden des letzten Stapels protokolliert wurden. Wenn Sie **Ältere Ereignisse laden** wählen, zeigt der Editor einen Stapel von Ereignissen an, die vor den aktuell angezeigten Ereignissen aufgetreten sind.

- Protokollereignisse kopieren: Wählen Sie die zu kopierenden Ereignisse aus, klicken Sie mit der rechten Maustaste **Kopieren** aus der Speisekarte.
- Kopieren Sie den Namen des Log-Streams: Klicken Sie mit der rechten Maustaste **Log Stream (Protokollstream)editor** und wähle **Protokollstreamname**.

Note

Sie können auch die verwenden-Befehlspalette führen Sie aus **AWSToolkit Protokollstreamname**.

Speichern des Inhalts des Protokolldatenstam-Editors in einer lokalen Datei

Sie können den Inhalt des CloudWatch Log-Stream-Editor zu einem Log-Datei auf Ihrem lokalen Computer.

Note

Mit dieser Option speichern Sie nur die Protokollereignisse, die derzeit im Log Stream Editor angezeigt werden. Wenn beispielsweise die Gesamtgröße eines Protokolldatenstroms 5 MB beträgt und nur 2 MB im Editor geladen werden, enthält die gespeicherte Datei auch nur 2 MB Protokolldaten. Um weitere zu speichernde Daten anzuzeigen, wählen Sie Laden Sie neuere Ereignisse oder Ältere Ereignisse laden im Editor.

1. Öffnen Sie den Protokollstream, der kopiert werden soll, den ProtokollstreamsHerausgeber (siehe [Anzeigen von CloudWatch-Protokollgruppen und Protokoll-Streams](#)) enthalten.
2. Wählen Sie das SymbolSave-Symbol neben der Registerkarte, die den Namen des Log-Streams anzeigt.

Note

Sie können auch die verwenden-Befehlspaletteführen Sie ausAWSToolkit Aktuellen Logstream-Inhalt speichern.

3. Verwenden Sie das Dialogfeld, um einen Downloadordner für die Protokolldatei auszuwählen oder zu erstellen, und klicken Sie auf Save.

CloudWatch Log-Gruppen durchsuchen

Sie können die Suchprotokollgruppe verwenden, um alle Protokolldatenströme in einer Protokollgruppe zu durchsuchen.

Ausführliche Informationen zum Amazon CloudWatch Logs-Service finden Sie im Thema [Arbeiten mit Protokollgruppen und Protokollstreams](#) im CloudWatch Amazon-Benutzerhandbuch.

Durchsuchen von Protokollgruppen in der VS Code-Befehlspalette

Gehen Sie wie folgt vor, um Protokollgruppen in der VS Code-Befehlspalette zu suchen.

Ausführliche Informationen zu den Filtern und Mustern von Amazon CloudWatch Logs finden Sie im Abschnitt [Filter- und Mustersyntax](#) im CloudWatch Amazon-Benutzerhandbuch.

1. Öffnen Sie in VS Code die Befehlspalette, indem Sie **cmd+shift+p** (windows:**ctrl+shift+p**) drücken.
2. Geben Sie in der Befehlspalette den Befehl ein und wählen Sie ihn aus **AWS: Search Log Group**, um das Dialogfeld „Suchprotokollgruppe“ in VS Code zu öffnen, und folgen Sie den Anweisungen, um fortzufahren.

Note

Ab der ersten Eingabeaufforderung haben Sie die Möglichkeit, Ihre AWS Region zu wechseln, bevor Sie mit den nächsten Schritten fortfahren.

3. Wählen Sie in der Eingabeaufforderung Protokollgruppe auswählen (1/3) die Protokollgruppe aus, die Sie durchsuchen möchten.
4. Wählen Sie in der Eingabeaufforderung Zeitfilter auswählen (2/3) den Zeitfilter aus, der auf Ihre Suche angewendet werden soll.
5. Geben Sie in der Eingabeaufforderung Search Log Group... (3/3) Ihr Suchmuster in das dafür vorgesehene Feld ein und drücken Sie dann die **Enter** Taste, um die Suche fortzusetzen, oder die **ESC** Taste, um die Suche abubrechen.
6. Ihre Suchergebnisse werden im VS Code-Editor geöffnet, wenn die Suche abgeschlossen ist.

Suchen nach Protokollgruppen im AWS Explorer

Gehen Sie wie folgt vor, um im AWS Toolkit for Visual Studio Code Explorer nach Protokollgruppen zu suchen.

1. Erweitern Sie im AWS Toolkit for Visual Studio Code Explorer CloudWatch.
2. Öffnen Sie das Kontextmenü für die Suchprotokollgruppe, die Sie durchsuchen möchten (klicken Sie mit der rechten Maustaste darauf), und wählen Sie dann Suchprotokollgruppe aus, um die Suchaufforderung zu öffnen.
3. Folgen Sie den Anweisungen, indem Sie einen Zeitraum auswählen, um fortzufahren.
4. Wenn Sie dazu aufgefordert werden, geben Sie Ihr Suchmuster in das dafür vorgesehene Feld ein und drücken Sie dann die **Enter** Taste, um fortzufahren, oder die **ESC** Taste, um die Suche abubrechen.

5. Ihre Suchergebnisse werden im VS Code-Editor geöffnet, wenn die Suche abgeschlossen ist.

Mit Ergebnissen des Suchprotokolls arbeiten

Nach Abschluss einer erfolgreichen CloudWatch Protokollgruppensuche werden Ihre Suchergebnisse im VS Code-Editor geöffnet. In den folgenden Verfahren wird beschrieben, wie Sie mit Suchprotokollergebnissen arbeiten.

Note

Bei der Anzeige eines einzelnen Protokolldatenstroms sind die folgenden Funktionen auf die Ergebnisse in Ihrem aktuell aktiven Protokollstream beschränkt.

Speichern der Gruppenergebnisse Ihres Suchprotokolls

Gehen Sie wie folgt vor, um die Ergebnisse Ihrer Suchprotokollgruppe lokal zu speichern.

1. Wählen Sie in den Ergebnissen Ihrer Suchprotokollgruppe die Symbolschaltfläche Protokoll in Datei speichern, die sich in der oberen rechten Ecke des VS Code-Editors befindet.
2. Geben Sie in der Eingabeaufforderung Speichern unter den Namen und den Speicherort an, an dem Sie die Datei speichern möchten.
3. Wenn Sie OK wählen, wird die Datei auf Ihrem lokalen Computer gespeichert.

Ändern des Zeitbereichs, des Zeitbereichs

Gehen Sie wie folgt vor, um den Zeitraum zu ändern, der in den Ergebnissen Ihrer Suchprotokollgruppe aktiv ist.

1. Wählen Sie in den Gruppenergebnissen Ihres Suchprotokolls die Option Nach Datum suchen... Symbolschaltfläche in der oberen rechten Ecke des VS Code-Editors.
2. Wählen Sie in der Aufforderung „Zeitfilter auswählen“ einen neuen Zeitraum für Ihre Suchprotokollergebnisse aus.
3. Ihre Ergebnisse werden aktualisiert, wenn die Aufforderung Zeitfilter auswählen geschlossen wird.

Das Suchmuster ändern

Gehen Sie wie folgt vor, um das Suchmuster zu ändern, das in den Ergebnissen Ihrer Suchprotokollgruppe aktiv ist.

1. Wählen Sie in den Gruppenergebnissen Ihres Suchprotokolls die Option Nach Muster suchen... Symbolschaltfläche in der oberen rechten Ecke des VS Code-Editors.
2. Geben Sie an der Eingabeaufforderung Search Log Group das neue Suchmuster in das dafür vorgesehene Feld ein.
3. Drücken Sie die **Enter** Eingabetaste, um die Eingabeaufforderung zu schließen und Ihre Ergebnisse mit dem neuen Suchmuster zu aktualisieren.

Arbeiten mit Amazon Elastic Container Registry

Amazon Elastic Container Registry (Amazon ECR) ist ein von AWS verwalteter Container-Registry-Service, der sicher und skalierbar ist. Über das Toolkit für den VS Code Explorer können Sie auf mehrere Amazon-ECR-Servicefunktionen zugreifen.

- Erstellen eines Repositorys.
- Erstellen eines AWS App Runner-Service für Ihr Repository oder ein markiertes Bild.
- Zugreifen auf Bild-Tags und Repository-URIs oder ARNs.
- Löschen von Bild-Tags und Repositorys.

Sie können auch über die VS-Code-Konsole auf alle Amazon-ECR-Funktionen zugreifen, indem Sie die AWS -CLI und andere Plattformen in VS Code integrieren.

Weitere Informationen zu Amazon ECR finden Sie unter [Was ist Amazon Elastic Container Registry?](#) im Benutzerhandbuch von Amazon Elastic Container Registry.

Voraussetzungen

Sie müssen diese Schritte ausführen, um über den VS Code Explorer auf den Amazon-ECR-Service zugreifen zu können.

Erstellen eines IAM-Benutzers

Bevor Sie auf einen AWS Service wie Amazon ECR zugreifen können, müssen Sie Anmeldeinformationen angeben. Auf diese Weise kann der Service bestimmen, ob Sie über die

Berechtigung für den Zugriff auf dessen Ressourcen verfügen. Wir raten Ihnen davon ab, AWS direkt über die Anmeldeinformationen für Ihr AWS Root-Konto auf zuzugreifen. Verwenden Sie stattdessen AWS Identity and Access Management (IAM), um einen IAM-Benutzer zu erstellen und fügen Sie diesen Benutzer zu einer IAM-Gruppe mit Administrator-Berechtigungen hinzu. Sie können dann mithilfe einer speziellen URL und mit den Anmeldeinformationen für den IAM-Benutzer auf AWS zugreifen.

Wenn Sie sich AWS zwar bei angemeldet, aber für sich selbst keinen IAM-Benutzer erstellt haben, können Sie mithilfe der IAM-Konsole einen Benutzer erstellen.

Wählen Sie zum Erstellen eines Administratorbenutzers eine der folgenden Optionen aus.

Wählen Sie eine Möglichkeit zur Verwaltung Ihres Administrators aus.	Bis	Von	Sie können auch
Im IAM Identity Center (Empfohlen)	<p>Verwendung von kurzfristigen Anmeldeinformationen für den Zugriff auf AWS.</p> <p>Dies steht im Einklang mit den bewährten Methoden für die Sicherheit. Weitere Informationen zu bewährten Methoden finden Sie unter Bewährte Methoden für die Sicherheit in</p>	Beachtung der Anweisungen unter Erste Schritte im AWS IAM Identity Center-Benutzerhandbuch.	Programmgesteuerten Zugriff unter Berücksichtigung der Informationen im Abschnitt Konfigurieren von AWS CLI für die Verwendung von AWS IAM Identity Center im AWS Command Line Interface-Benutzerhandbuch konfigurieren.

Wählen Sie eine Möglichkeit zur Verwaltung Ihres Administrators aus.	Bis	Von	Sie können auch
	IAM im IAM-Benutzerhandbuch.		
In IAM (Nicht empfohlen)	Verwendung von langfristigen Anmeldeinformationen für den Zugriff auf AWS.	Beachtung der Anweisungen unter Erstellen Ihres ersten IAM-Administrators und Ihrer ersten Benutzergruppe im IAM-Benutzerhandbuch.	Programmgesteuerten Zugriff unter Verwendung der Informationen unter Verwalten der Zugriffsschlüssel für IAM-Benutzer im IAM-Benutzerhandbuch konfigurieren.

Um sich als diesen neuen IAM-Benutzer anzumelden, melden Sie sich zunächst aus der AWS Konsole ab. Verwenden Sie dann die folgende URL. In der folgenden URL, wobei `your_aws_account_id` ohne Bindestriche Ihre AWS Konto-ID darstellt. (Wenn beispielsweise Ihre AWS Konto-ID lautet `1234-5678-9012`, ist Ihre AWS Konto-ID `123456789012`):

```
https://your_aws_account_id.signin.aws.amazon.com/console/
```

Geben Sie den IAM-Benutzernamen und das von Ihnen soeben erstellte Passwort ein. Nachdem Sie sich angemeldet haben, wird in der Navigationsleiste „`your_user_name @ your_aws_account_id`“ angezeigt.

Wenn Sie nicht möchten, dass die URL für Ihre Anmeldeseite Ihre AWS-Konto-ID enthält, können Sie einen Konto-Alias erstellen. Wählen Sie im IAM-Dashboard die Option `Customize` und geben Sie ein -Konto-Alias ein. Dies kann Ihr Firmenname sein. Weitere Informationen finden Sie unter [Ihre AWS Konto-ID und ihr Alias](#) im IAM-Benutzerhandbuch.

Nach dem Erstellen eines Konto-Alias verwenden Sie die folgende URL, um sich anzumelden:

```
https://your_account_alias.signin.aws.amazon.com/console/
```

Um den Anmeldelink der IAM-Benutzer Ihres Kontos zu verifizieren, öffnen Sie die IAM-Konsole und prüfen dies im Dashboard unter IAM users sign-in link.

Weitere Informationen über IAM finden Sie im [AWS Identity and Access Management Benutzerhandbuch](#).

Installieren und konfigurieren Sie Docker

Sie können Docker installieren und konfigurieren, indem Sie Ihr bevorzugtes Betriebssystem aus dem [Install Docker Engine-Benutzerhandbuch](#) auswählen und den Anweisungen folgen.

Installieren und konfigurieren Sie AWS CLI Version 2

Installieren und konfigurieren Sie AWS CLI Version 2, indem Sie Ihr bevorzugtes Betriebssystem aus dem Benutzerhandbuch zur [Installation, Aktualisierung und Deinstallation der AWS CLI Version 2](#) auswählen.

Themen

- [Arbeiten mit Amazon Elastic Container Registry in VS Code](#)

Arbeiten mit Amazon Elastic Container Registry in VS Code

Sie können den Amazon-Elastic-Container-Registry- (Amazon-ECR) -Dienst direkt aus dem AWS Explorer in VS Code verwenden, um ein Programmabbild in ein Amazon ECR-Repository zu übertragen. Um zu beginnen, müssen Sie die folgenden Schritte ausführen:

1. Erstellen Sie eine Dockerfile, die die zum Erstellen eines Images erforderlichen Informationen enthält.
2. Erstellen Sie ein Image aus diesem Dockerfile und kennzeichnen Sie das Bild zur Verarbeitung.
3. Erstellen Sie ein Repository in Ihrer Amazon ECR-Instanz.
4. Verschieben Sie das Image mit Tags, um Ihr Repository.

Abschnitte

- [Voraussetzungen](#)

- [1. Erstellen eines Docker-Datei](#)
- [2. Erstellen Sie Ihr Image aus der Docker-Datei](#)
- [3. Erstellen Sie ein neues Repository](#)
- [4. Push, Pull-Übertragung und Löschen von Images](#)

Voraussetzungen

Bevor Sie die Amazon ECR-Servicefunktion des Toolkit for VS Code verwenden können, müssen Sie diese erfüllen [Voraussetzungen](#).

1. Erstellen eines Docker-Datei

Docker verwendet eine Datei namens Dockerfile, um ein Image zu definieren, das gepusht und in einem Remote-Repository gespeichert werden kann. Bevor Sie ein Image in ein ECR-Repository hochladen können, müssen Sie eine Dockerfile erstellen und dann ein Image aus dieser Dockerfile erstellen.

Erstellen eines Docker-Datei

1. Navigieren Sie mit dem Toolkit for VS Code -Explorer zu dem Verzeichnis, in dem Sie Ihre Dockerfile speichern möchten.
2. Erstellen Sie eine neue Datei mit dem Namen Dockerfile.

Note

VS Code könnte Sie auffordern, einen Dateityp oder eine Dateiendung auszuwählen. Wenn dies der Fall ist, wählen Sie Nur-Text. Vs Code hat eine „Dockerfile“-Erweiterung. Wir empfehlen Ihnen jedoch nicht, es zu verwenden. Dies liegt daran, dass die Erweiterung Konflikte mit bestimmten Versionen von Docker oder anderen zugehörigen Anwendungen verursachen kann.

Bearbeiten Sie Ihre Dockerfile mit VS Code

Wenn Ihr Dockerfile eine Dateierweiterung hat, öffnen Sie das Kontextmenü (rechte Maustaste) für die Datei und entfernen Sie die Dateierweiterung.

Nachdem die Dateierweiterung aus Ihrem Dockerfile entfernt wurde:

1. Öffnen Sie das leere Dockerfile direkt in VS Code.
2. Kopieren Sie den Inhalt des folgenden Beispiels in Ihr Dockerfile:

Example Dockerfile Image Vorlage

```
FROM ubuntu:18.04

# Install dependencies
RUN apt-get update && \
    apt-get -y install apache2

# Install apache and write hello world message
RUN echo 'Hello World!' > /var/www/html/index.html

# Configure apache
RUN echo '. /etc/apache2/envvars' > /root/run_apache.sh && \
    echo 'mkdir -p /var/run/apache2' >> /root/run_apache.sh && \
    echo 'mkdir -p /var/lock/apache2' >> /root/run_apache.sh && \
    echo '/usr/sbin/apache2 -D FOREGROUND' >> /root/run_apache.sh && \
    chmod 755 /root/run_apache.sh

EXPOSE 80

CMD /root/run_apache.sh
```

Dies ist eine Dockerfile, die ein Ubuntu-18.04-Image verwendet. Die `LAUF`Anweisungen aktualisieren die Paket-Caches. Installieren Sie Softwarepakete für den Webserver und schreiben Sie dann den Inhalt „Hello World!“ in das Dokumenten-Stammverzeichnis des Webserver. Die `AUSSETZEN`-Befehl stellt Port 80 auf dem Container bereit und die `CMD`Anweisung startet den Webserver.

3. Speichern Sie die Docker-Datei.

Important

Stellen Sie sicher, dass Ihre Dockerfile keine Erweiterung an den Namen angehängt hat. Eine Dockerfile mit Erweiterungen kann Konflikte mit bestimmten Versionen von Docker oder anderen zugehörigen Anwendungen verursachen.

2. Erstellen Sie Ihr Image aus der Docker-Datei

Die von Ihnen erstellte Dockerfile enthält die Informationen, die zum Erstellen eines Images für ein Programm erforderlich sind. Bevor Sie dieses Image an Ihre Amazon ECR-Instanz übertragen können, müssen Sie zuerst das Image erstellen.

Erstellen Sie ein Image aus der Docker-Datei

1. Verwenden Sie die Docker CLI oder eine CLI, die in Ihre Docker-Instanz integriert ist, um in das Verzeichnis zu navigieren, das Ihre Dockerfile enthält.
2. Ausführen des `sDocker-Build` das Image zu erstellen, das in Ihrer Dockerfile definiert ist.

```
docker build -t hello-world .
```

3. Ausführen des `sDocker-Images`-Befehl, um zu überprüfen, ob das Abbild korrekt erstellt wurde.

```
docker images --filter reference=hello-world
```

Example Beispielausgabe:

REPOSITORY	TAG	IMAGE ID	CREATED
hello-world	latest	e9ffedc8c286	4 minutes ago
SIZE			
241MB			

4.

Note

Dieser Schritt ist nicht erforderlich, um ein Image zu erstellen oder zu pushen, aber Sie können sehen, wie das Programm-Image funktioniert, wenn es ausgeführt wird.

Um das neu erstellte Image auszuführen, verwenden Sie den Befehl `DockerausgabeBefehl`.

```
docker run -t -i -p 80:80 hello-world
```

Die `-p`-Option, die im vorherigen Beispiel angegeben wurde, ordnet das exponierte Port 80 auf dem Behälter zu Port 80 des Hostsystems. Wenn Sie Docker lokal ausführen, navigieren Sie zu <http://localhost:80> Verwenden des Webbrowsers. Wenn das Programm korrekt lief, wird ein „Hello World!“ -Anweisung wird angezeigt.

Weitere Informationen zu Docker-Ausgabe-Befehl, siehe [Docker-Ausgabe](#) auf der Docker-Website.

3. Erstellen Sie ein neues Repository

Um Ihr Bild in Ihre Amazon ECR-Instanz hochzuladen, erstellen Sie ein neues Repository, in dem es gespeichert werden kann.

Erstellen Sie ein neues Amazon-ECR-Repository

1. Aus dem VS-Code-Aktivitätsleiste, wähle das AWS-Toolkit-Symbol.
2. Expand erweitern AWS-Explorer-Menü.
3. Suchen Sie die Standardeinstellung AWS-Region, die mit Ihrem verknüpft ist AWS-Konto. Wählen Sie es dann aus, um eine Liste der Dienste anzuzeigen, die über das Toolkit for VS Code verfügbar sind.
4. Wählen Sie das Symbol ECR + Option, um mit dem Erstellen Sie ein neues -Prozess.
5. Folgen Sie den Anweisungen, um den Vorgang abzuschließen.
6. Nach Abschluss können Sie Ihr neues Repository über das ECR-Abschnitt im AWS-Explorer-Menü.

4. Push, Pull-Übertragung und Löschen von Images

Nachdem Sie ein Image aus Ihrem Dockerfile erstellt und ein Repository erstellt haben, können Sie Ihr Image in Ihr Amazon ECR-Repository pushen. Verwenden Sie zusätzlich das AWS-Explorer mit Docker und dem AWS CLI können Sie folgende Aktionen ausführen:

- Rufen Sie ein Image aus dem -Repository ab.
- Lösche ein Bild, das in deinem Repository gespeichert ist.
- Lösche dein Repository.

Authentifizieren Sie Docker mit Ihrer Standardregistrierung

Für den Datenaustausch zwischen Amazon ECR- und Docker-Instances ist eine Authentifizierung erforderlich. So authentifizieren Sie Docker bei Ihrer Registrierung:

1. Öffnen Sie ein Befehlszeilenbetriebssystem, das mit Ihrer Instanz von verbunden istAWSCLI.
2. Verwenden derget-login-passwordMethode zur Authentifizierung bei Ihrer privaten ECR-Registrierung.

```
aws ecr get-login-password --region region | docker login --username AWS --password-stdin AWS_account_id.dkr.ecr.region.amazonaws.com
```

Important

Im vorhergehenden Befehl müssen Sie beider**region**und die**AWS_account_id**zu den Informationen, die spezifisch für Ihre sindAWSKonto.

Versehen und pushen Sie ein Image in Ihr Repository

Nachdem Sie Docker mit Ihrer Instanz von authentifiziert habenAWS, pushen Sie ein Image in Ihr Repository.

1. Verwenden derDocker-Images, um die lokal gespeicherten Bilder anzuzeigen und das zu identifizieren, das Sie markieren möchten.

```
docker images
```

Example Beispielausgabe:

REPOSITORY	TAG	IMAGE ID	CREATED
hello-world	latest	e9ffedc8c286	4 minutes ago
SIZE			
241MB			

2. Kennzeichnen Sie Ihr Bild mit demDocker-TagBefehl.

```
docker tag hello-world:latest AWS_account_id.dkr.ecr.region.amazonaws.com/hello-world:latest
```

3. Verschieben Sie das markierte Image in Ihr Repository mit dem Docker-Tag-Befehl.

```
docker push AWS_account_id.dkr.ecr.region.amazonaws.com/hello-world:latest
```

Example Beispielausgabe:

```
The push refers to a repository [AWS_account_id.dkr.ecr.region.amazonaws.com/hello-world] (len: 1)
e9ae3c220b23: Pushed
a6785352b25c: Pushed
0998bf8fb9e9: Pushed
0a85502c06c9: Pushed
latest: digest:
sha256:215d7e4121b30157d8839e81c4e0912606fca105775bb0636b95aed25f52c89b size: 6774
```

Nachdem dein markiertes Bild erfolgreich in dein Repository hochgeladen wurde, ist es in der AWS Explorer-Menü.

Abruf eines Abbilds von Amazon ECR

- Sie können ein Image auf Ihre lokale Instanz von Docker-Tag-Befehl.

```
docker pull AWS_account_id.dkr.ecr.region.amazonaws.com/hello-world:latest
```

Example Beispielausgabe:

```
The push refers to a repository [AWS_account_id.dkr.ecr.region.amazonaws.com/hello-world] (len: 1)
e9ae3c220b23: Pushed
a6785352b25c: Pushed
0998bf8fb9e9: Pushed
0a85502c06c9: Pushed
```

```
latest: digest:  
  sha256:215d7e4121b30157d8839e81c4e0912606fca105775bb0636b95aed25f52c89b size: 6774
```

Löschen Sie ein Image aus Ihrem Amazon ECR-Repository

Es gibt zwei Methoden zum Löschen eines Images aus VS Code. Die erste Methode ist die Verwendung des AWS-Explorers.

1. Von der AWS Explorer, erweitern Sie das ECR Speisekarte
2. Erweitern Sie das Repository, aus dem Sie ein Bild löschen möchten
3. Wählen Sie das Image-Tag aus, das Sie löschen möchten, indem Sie das Kontextmenü öffnen (rechte Maustaste).
4. Wählen Sie das Symbol Löschen von Tags... Option, um alle gespeicherten Bilder zu löschen, die mit diesem Tag verknüpft sind

Löschen eines Abbilds mit dem AWS CLI

- Löschen können Sie ein Image auch mit dem Inhalt AWS CLI `batch-delete-image` Befehl.

```
AWS ecr batch-delete-image \  
  --repository-name hello-world \  
  --image-ids imageTag=latest
```

Example Beispielausgabe:

```
{  
  "failures": [],  
  "imageIds": [  
    {  
      "imageTag": "latest",  
      "imageDigest":  
      "sha256:215d7e4121b30157d8839e81c4e0912606fca105775bb0636b95aed25f52c89b"  
    }  
  ]  
}
```

Löschen Sie ein Repository aus Ihrer Amazon ECR-Instanz

Es gibt zwei Methoden, um ein Repository aus VS Code zu löschen. Die erste Methode ist die Verwendung des AWS-Explorers.

1. Von der AWS Explorer, erweitern Sie das ECR Speisekarte
2. Wählen Sie das Repository aus, das Sie löschen möchten, indem Sie das Kontextmenü öffnen (rechte Maustaste).
3. Wählen Sie das Symbol Repository... Option zum gewählten -Repository

Löschen eines Amazon ECR-Repositorys aus AWS CLI

- Du kannst ein Repository mit dem `aws ecr delete-repository` Befehl.

Note

Standardmäßig können Sie kein Repository löschen, das Images enthält. Allerdings ist der `--force` Flag erlaubt dies.

```
aws ecr delete-repository \  
--repository-name hello-world \  
--force
```

Example Beispielausgabe:

```
{  
  "failures": [],  
  "imageIds": [  
    {  
      "imageTag": "latest",  
      "imageDigest":  
        "sha256:215d7e4121b30157d8839e81c4e0912606fca105775bb0636b95aed25f52c89b"    }  
  ]  
}
```



```
}  
  ]  
}
```

Arbeiten mit Amazon Elastic Container Service

Die AWS Toolkit for Visual Studio Code bietet Unterstützung für [Amazon Elastic Container Service \(Amazon ECS\)](#) aus. Das Toolkit for VS Code unterstützt Sie bei bestimmten Amazon ECS-bezogenen Arbeiten, z. B. beim Erstellen von Aufgabendefinitionen.

Themen

- [IntelliSense für Amazon ECS-Task-Definitionsdateien verwenden](#)
- [Amazon Elastic Container Service Exec in AWS Toolkit for Visual Studio Code](#)

IntelliSense für Amazon ECS-Task-Definitionsdateien verwenden

Eines der Dinge, die Sie bei der Arbeit mit Amazon Elastic Container Service (Amazon ECS) tun können, besteht darin, Aufgabendefinitionen zu erstellen, wie unter [Erstellen einer Aufgabendefinition](#) aus Amazon Elastic Container Service Entwicklerhandbuch aus. Bei der Installation des AWS Toolkit for Visual Studio Code enthält die Installation IntelliSense-Funktionalität für Amazon ECS -Aufgabendefinitionsdateien.

Voraussetzungen

- Stellen Sie sicher, dass Ihr System die Voraussetzungen erfüllt, die unter [Installieren des Toolkit for VS Code](#) aus.

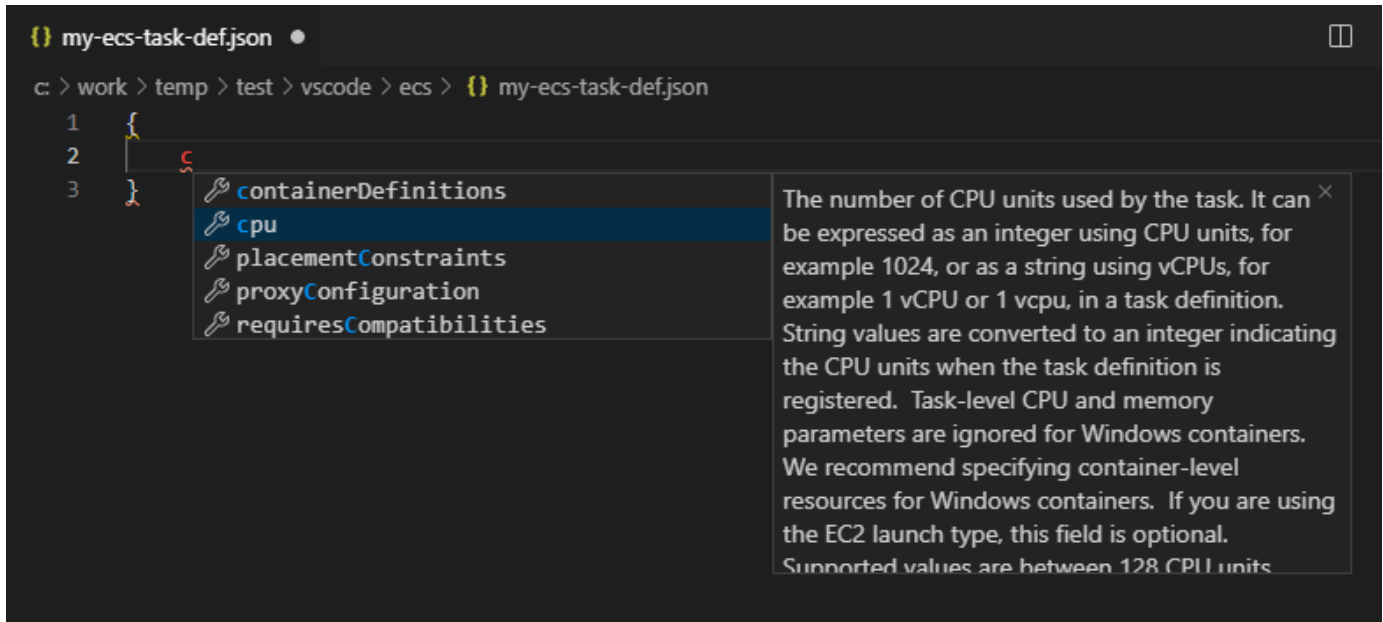
Verwenden Sie IntelliSense in Amazon ECS -Aufgabendefinitionsdateien

Das folgende Beispiel zeigt Ihnen, wie Sie IntelliSense in Amazon ECS -Aufgabendefinitionsdateien nutzen können.

1. Erstellen Sie eine JSON-Datei für die Amazon ECS -Aufgabendefinition. Der Dateiname muss `ecs-task-def.json` am Ende haben, kann aber am Anfang zusätzliche Zeichen haben.

Erstellen Sie in diesem Beispiel eine Datei mit dem Namen `my-ecs-task-def.json`.

2. Öffnen Sie die Datei in einem VS Code-Editor, und geben Sie die anfänglichen geschweiften Klammern ein.
3. Geben Sie den Buchstaben „c“ ein, als ob Sie der Definition `cpu` hinzufügen wollten. Beachten Sie das angezeigte IntelliSense-Dialogfeld, das dem folgenden ähnelt.



The screenshot shows a VS Code editor window with a file named `my-ecs-task-def.json`. The editor content is a JSON object with an opening curly brace on line 1. On line 2, the letter 'c' is typed, and an IntelliSense dropdown menu is visible. The menu lists several properties: `containerDefinitions`, `cpu` (highlighted), `placementConstraints`, `proxyConfiguration`, and `requiresCompatibilities`. A tooltip for the `cpu` property is displayed to the right of the dropdown. The tooltip text reads: "The number of CPU units used by the task. It can be expressed as an integer using CPU units, for example 1024, or as a string using vCPUs, for example 1 vCPU or 1 vcpu, in a task definition. String values are converted to an integer indicating the CPU units when the task definition is registered. Task-level CPU and memory parameters are ignored for Windows containers. We recommend specifying container-level resources for Windows containers. If you are using the EC2 launch type, this field is optional. Supported values are between 128 CPU units..."

Amazon Elastic Container Service Exec in AWS Toolkit for Visual Studio Code

Mithilfe der Amazon ECS Exec-Funktion können Sie einzelne Befehle in einem Amazon Elastic Container Service (Amazon ECS) -Container mit dem ausgeben. AWS Toolkit for Visual Studio Code

Important

Durch die Aktivierung und Deaktivierung von Amazon ECS Exec ändert sich der Status der Ressourcen in Ihrem AWS Konto. Dies beinhaltet das Beenden und Neustarten des Service. Das Ändern des Ressourcenzustands bei aktiviertem Amazon ECS Exec kann zu unvorhersehbaren Ergebnissen führen. Weitere Informationen zu Amazon ECS finden Sie im Entwicklerhandbuch [Using Amazon ECS Exec for Debugging](#).

Voraussetzungen für Amazon ECS Exec

Bevor Sie die Amazon ECS Exec-Funktion verwenden können, müssen einige Voraussetzungen erfüllt sein.

Amazon-ECS-Anforderungen

Je nachdem, ob Ihre Aufgaben auf Amazon EC2 oder Amazon ECS Exec gehostet werden AWS Fargate (Fargate), gelten für Amazon ECS Exec unterschiedliche Versionsanforderungen.

- Wenn Sie Amazon EC2 verwenden, müssen Sie ein für Amazon ECS optimiertes AMI verwenden, das nach dem 20. Januar 2021 mit einer Agent-Version von 1.50.2 oder höher veröffentlicht wurde. Weitere Informationen finden Sie im Entwicklerhandbuch unter [Für Amazon ECS optimierte AMIs](#).
- Wenn Sie verwenden AWS Fargate, müssen Sie die Plattformversion 1.4.0 oder höher verwenden. Weitere Informationen zu Fargate-Anforderungen finden Sie im Entwicklerhandbuch unter [AWS Fargate -Plattformversionen](#).

AWS Kontokonfiguration und IAM-Berechtigungen

Um die Amazon ECS Exec-Funktion nutzen zu können, muss ein vorhandener Amazon ECS-Cluster mit Ihrem AWS Konto verknüpft sein. Amazon ECS Exec verwendet Systems Manager, um eine Verbindung mit den Containern in Ihrem Cluster herzustellen, und benötigt spezifische Berechtigungen vom Typ „IAM-Rolle für Aufgabe“, um mit dem SSM-Service zu kommunizieren.

Spezifische Informationen zu IAM-Rollen und Richtlinien für Amazon ECS Exec finden Sie im Entwicklerhandbuch unter [Erforderliche IAM-Berechtigungen für ECS Exec](#).

Arbeiten mit Amazon ECS Exec

Sie können Amazon ECS Exec direkt im AWS Explorer im Toolkit for VS Code aktivieren oder deaktivieren. Wenn Sie Amazon ECS Exec aktiviert haben, können Sie Container aus dem Amazon ECS-Menü auswählen und dann Befehle für sie ausführen.

Aktivieren von Amazon ECS Exec

1. Suchen Sie im AWS Explorer das Amazon ECS-Menü und erweitern Sie es.
2. Erweitern Sie den Cluster mit dem Service, den Sie ändern möchten.
3. Öffnen Sie das Kontextmenü (rechte Maustaste) für den Service und wählen Sie Befehlsausführung aktivieren aus.

⚠ Important

Dadurch wird eine neue Bereitstellung Ihres Service gestartet, was einige Minuten dauern kann. Weitere Informationen finden Sie im Hinweis am Anfang dieses Abschnitts.

Deaktivieren von Amazon ECS Exec

1. Suchen Sie im AWS Explorer das Amazon ECS-Menü und erweitern Sie es.
2. Erweitern Sie den Cluster, der den gewünschten Service enthält.
3. Öffnen Sie das Kontextmenü (rechte Maustaste) für den Service und wählen Sie Befehlsausführung deaktivieren aus.

⚠ Important

Dadurch wird eine neue Bereitstellung Ihres Service gestartet, was einige Minuten dauern kann. Weitere Informationen finden Sie im Hinweis am Anfang dieses Abschnitts.

Ausführen von Befehlen für einen Container

Um mithilfe des AWS Explorers Befehle für einen Container auszuführen, muss Amazon ECS Exec aktiviert sein. Wenn es nicht aktiviert ist, lesen Sie das Verfahren zur Aktivierung von ECS Exec in diesem Abschnitt.

1. Suchen Sie im AWS Explorer das Amazon ECS-Menü und erweitern Sie es.
2. Erweitern Sie den Cluster, der den gewünschten Service enthält.
3. Erweitern Sie den Service, um die zugehörigen Container aufzulisten.
4. Öffnen Sie das Kontextmenü für den Container (Rechtsklick) und wählen Sie Befehl im Container ausführen aus.
5. Eine Eingabeaufforderung mit einer Liste der laufenden Aufgaben wird geöffnet. Wählen Sie den gewünschten Task-ARN aus.

Note

Wenn nur ein Task für diesen Service ausgeführt wird, wird er automatisch ausgewählt und dieser Schritt wird übersprungen.

6. Wenn Sie dazu aufgefordert werden, geben Sie den Befehl ein, den Sie ausführen möchten, und drücken Sie die Eingabetaste, um den Vorgang auszuführen.

Arbeiten mit Amazon EventBridge

Die AWS Toolkit for Visual Studio Code (VS Code) bietet Unterstützung für [Amazon EventBridge](#) aus. Mit dem Toolkit for VS Code können Sie mit bestimmten Aspekten von EventBridge arbeiten, etwa mit Schemata.

Themen

- [Mit EventBridge Amazon-Schemas arbeiten](#)

Mit EventBridge Amazon-Schemas arbeiten

Sie können den AWS Toolkit for Visual Studio Code (VS-Code) verwenden, um verschiedene Operationen mit [EventBridge Amazon-Schemas](#) durchzuführen.

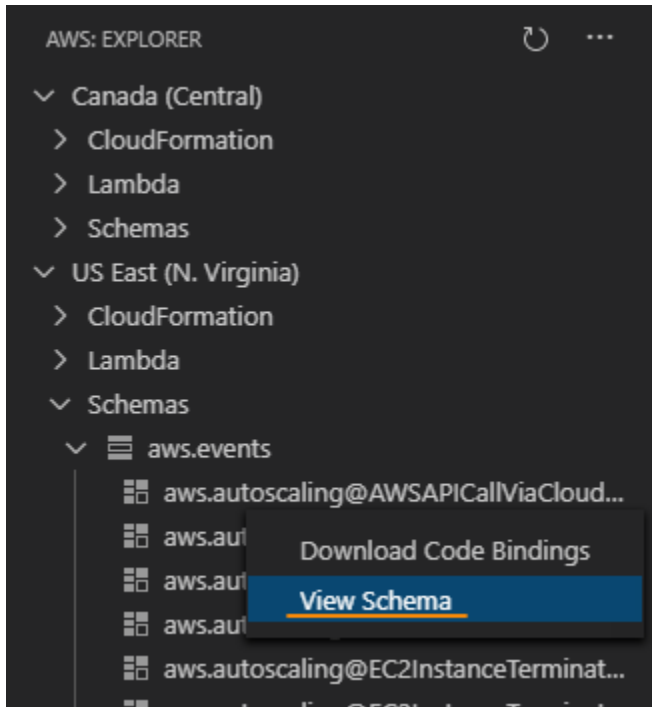
Voraussetzungen

- Stellen Sie sicher, dass Ihr System die unter [Installation des Toolkit for VS Code](#) angegebenen Voraussetzungen erfüllt.
- Das EventBridge Schema, mit dem Sie arbeiten möchten, muss in Ihrem AWS Konto verfügbar sein. Wenn dies nicht der Fall ist, erstellen Sie es, oder laden Sie es hoch. Weitere Informationen finden Sie im [EventBridge Amazon-Benutzerhandbuch unter EventBridge Amazon-Schemas](#).

Anzeigen eines verfügbaren Schemas

1. Erweitern Sie im AWS -Explorer Schemas (Schemata).
2. Erweitern Sie den Namen der Registrierung, die das Schema enthält, das Sie anzeigen möchten. Beispielsweise befinden sich viele der bereitgestellten Schemas in der AWS Registrierung `aws.events`.

- Um ein Schema im Editor anzuzeigen, öffnen Sie das Kontextmenü des Schemas, und wählen Sie dann View Schema (Schema anzeigen).

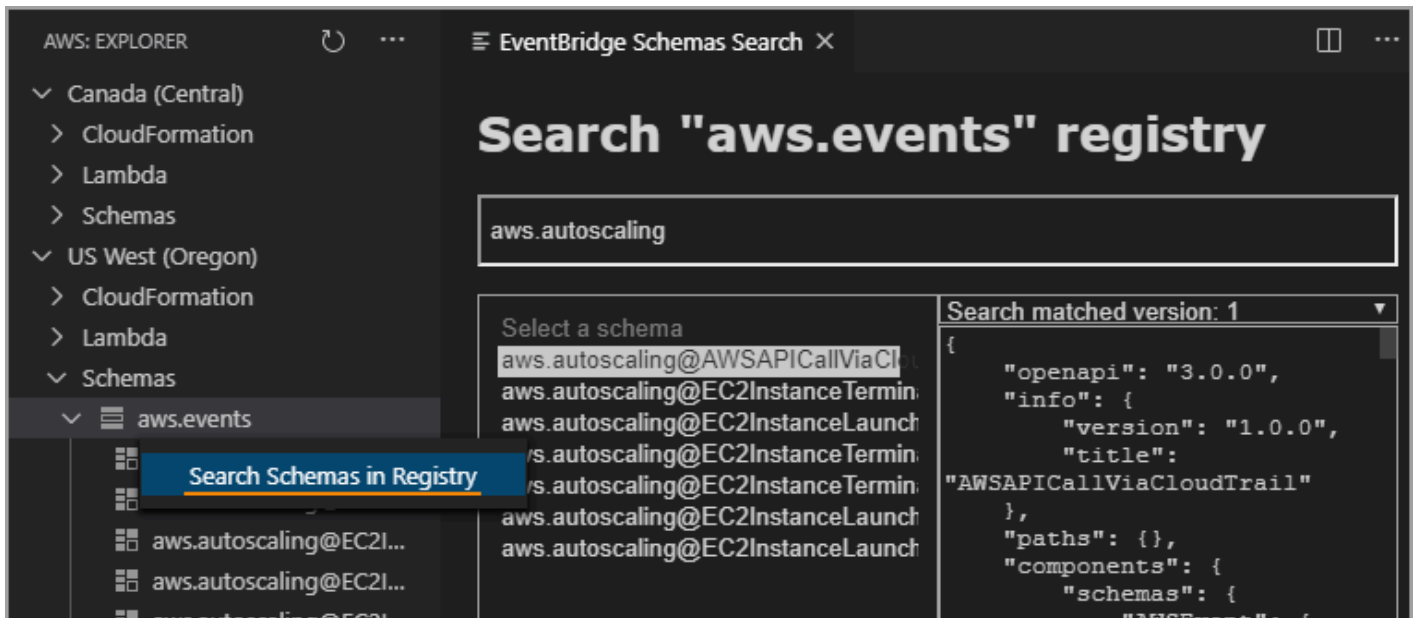


Suchen eines verfügbaren Schemas

Führen Sie im AWS -Explorer eine oder mehrere der folgenden Aktionen aus:

- Beginnen Sie mit der Eingabe des Titels des Schemas, das Sie suchen möchten. Der AWS -Explorer hebt die Schema-Titel hervor, die eine Übereinstimmung enthalten. (Eine Registrierung muss erweitert werden, damit Sie die hervorgehobenen Titel sehen können.)
- Öffnen Sie das Kontextmenü für Schemas (Schemata), und wählen Sie Search Schemas (Schemata suchen). Oder erweitern Sie Schemas (Schemata), öffnen Sie das Kontextmenü für die Registrierung, die das gewünschte Schema enthält, und wählen Sie dann Search Schemas in Registry (Schemata in der Registrierung suchen). Geben Sie im Dialogfeld EventBridge Schemasuche den Titel des Schemas ein, nach dem Sie suchen möchten. Im Dialogfeld werden die Schematitel angezeigt, die eine Übereinstimmung enthalten.

Um das Schema im Dialogfeld anzuzeigen, wählen Sie den Titel des Schemas aus.



Generieren des Codes für ein verfügbares Schema

1. Erweitern Sie im AWS -Explorer Schemas (Schemata).
2. Erweitern Sie den Namen der Registrierung, die das Schema enthält, für das Sie Code generieren möchten.
3. Klicken Sie mit der rechten Maustaste auf den Titel des Schemas, und wählen Sie dann Download code bindings (Codebindungen herunterladen).
4. Wählen Sie auf den resultierenden Assistentenseiten Folgendes aus:
 - Die Version des Schemas.
 - Die Code-Bindungssprache
 - Den Arbeitsbereichordner, in dem Sie den generierten Code auf Ihrem lokalen Entwicklungscomputer speichern möchten

AWS IAM-Zugriffsanalysator

Mit dem IAM [Access Analyzer in der können Sie AWS Identity and Access Management \(IAM\) Access Analyzer-Richtlinienprüfungen](#) für Ihre in AWS CloudFormation Vorlagen, Terraform-Plänen und JSON-Richtliniendokumenten verfassten IAM-Richtlinien ausführen. AWS Toolkit for Visual Studio Code

Zu den Richtlinienprüfungen von IAM Access Analyzer gehören die Richtlinienvvalidierung und benutzerdefinierte Richtlinienprüfungen. Die Richtlinienvvalidierung hilft Ihnen dabei, Ihre IAM-Richtlinien gemäß den Standards zu validieren, die in den Themen [Grammatik der IAM-JSON-Richtliniensprache](#) und [Bewährte AWS Sicherheitsmethoden in IAM](#) im Benutzerhandbuch beschrieben sind. AWS Identity and Access Management Ihre Ergebnisse bei der Richtlinienvvalidierung umfassen Sicherheitswarnungen, Fehler, allgemeine Warnungen und Richtlinienvorschläge.

Sie können auch benutzerdefinierte Richtlinienprüfungen für neuen Zugriff durchführen, die auf Ihren Sicherheitsstandards basieren. Für jede benutzerdefinierte Überprüfung der Richtlinien für einen neuen Zugriff wird eine Gebühr erhoben. Detaillierte Informationen zu den Preisen finden Sie auf der [Preisseite für AWS IAM Access Analyzer](#). Einzelheiten zu den IAM Access Analyzer-Richtlinienprüfungen finden Sie unter dem Thema [Prüfungen zur Überprüfung von Richtlinien](#) im AWS Identity and Access Management Benutzerhandbuch.

In den folgenden Themen wird beschrieben, wie Sie mit IAM Access Analyzer-Richtlinienprüfungen arbeiten. AWS Toolkit for Visual Studio Code

Themen

- [Arbeiten mit AWS IAM Access Analyzer](#)

Arbeiten mit AWS IAM Access Analyzer

In den folgenden Abschnitten wird beschrieben, wie Sie die IAM-Richtlinienvvalidierung und benutzerdefinierte Richtlinienprüfungen in der durchführen. AWS Toolkit for Visual Studio Code Weitere Informationen finden Sie in den folgenden Themen im AWS Identity and Access Management Benutzerhandbuch: Überprüfung der [IAM Access Analyzer-Richtlinien und Überprüfung benutzerdefinierter IAM Access Analyzer-Richtlinien](#).

Voraussetzungen

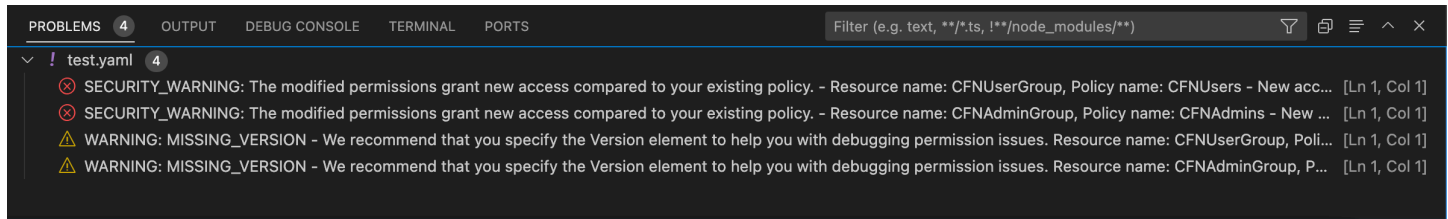
Die folgenden Voraussetzungen müssen erfüllt sein, bevor Sie mit den IAM Access Analyzer-Richtlinienprüfungen aus dem Toolkit arbeiten können.

- Installieren Sie Python Version 3.6 oder höher.
- Installieren Sie entweder den [IAM-Richtlinienvvalidator für oder den AWS CloudFormation IAM-Richtlinienvvalidator für Terraform, der für](#) die Python-CLI-Tools erforderlich und im Fenster IAM-Richtlinienprüfungen angegeben ist.

- Konfigurieren Sie Ihre Rollenmeldedaten. AWS

IAM Access Analyzer-Richtlinienüberprüfungen

Mit dem können Sie Richtlinienprüfungen für AWS CloudFormation Vorlagen, Terraform-Pläne und JSON-Richtliniendokumente durchführen. AWS Toolkit for Visual Studio Code Ihre Prüfungsergebnisse können im VS Code Problems Panel eingesehen werden. Die folgende Abbildung zeigt das VS Code Problems Panel.



IAM Access Analyzer bietet drei Arten von Prüfungen:

- Richtlinie validieren
- CheckAccessNotGranted
- CheckNoNewAccess

In den folgenden Abschnitten wird beschrieben, wie die einzelnen Prüfungstypen ausgeführt werden.

Note

Konfigurieren Sie Ihre AWS Rollenmeldedaten, bevor Sie eine Prüfung durchführen. Zu den unterstützten Dateien gehören die folgenden Dokumententypen: AWS CloudFormation Vorlagen, Terraform-Pläne und JSON-Richtliniendokumente

Validate Policy wird ausgeführt

Die Überprüfung „Richtlinie validieren“, auch als Richtlinienvvalidierung bezeichnet, validiert Ihre Richtlinie anhand der Grammatik und der AWS Best Practices der IAM-Richtlinien. Weitere Informationen finden Sie in den Themen [Grammatik der IAM-JSON-Richtliniensprache](#) und [Bewährte AWS Sicherheitsmethoden in IAM](#) im Benutzerhandbuch. AWS Identity and Access Management

1. Öffnen Sie in VS Code eine unterstützte Datei, die AWS IAM-Richtlinien enthält, im VS Code-Editor.

2. Um die IAM Access Analyzer-Richtlinienprüfungen zu öffnen, öffnen Sie die VS Code-Befehlspalette **CTRL+Shift+P**, indem Sie auf, suchen und dann klicken **IAM Policy Checks**, um den Bereich IAM-Richtlinienprüfungen im VS Code-Editor zu öffnen.
3. Wählen Sie im Bereich IAM Policy Checks Ihren Dokumenttyp aus dem Dropdownmenü aus.
4. Wählen Sie im Abschnitt „Richtlinien validieren“ die Schaltfläche „Richtlinienvvalidierung ausführen“, um die Überprüfung „Richtlinie validieren“ auszuführen.
5. Überprüfen Sie im Problembereich in VS Code die Ergebnisse Ihrer Richtlinienüberprüfung.
6. Aktualisieren Sie Ihre Richtlinie und wiederholen Sie dieses Verfahren. Führen Sie die Überprüfung „Richtlinie validieren“ erneut aus, bis Ihre Ergebnisse der Richtlinienüberprüfung keine Sicherheitswarnungen oder Fehler mehr anzeigen.

Wird ausgeführt CheckAccessNotGranted

CheckAccessNotGranted ist eine benutzerdefinierte Richtlinienüberprüfung, um sicherzustellen, dass bestimmte IAM-Aktionen gemäß Ihrer Richtlinie nicht zulässig sind.

Note

Mit jeder Überprüfung der benutzerdefinierten Richtlinien ist eine Gebühr verbunden. Einzelheiten zu den Preisen für benutzerdefinierte Policy-Checks finden Sie im Preisleitfaden für [AWS IAM Access Analyzer](#).

1. Öffnen Sie in VS Code eine unterstützte Datei, die AWS IAM-Richtlinien enthält, im VS Code-Editor.
2. Um die IAM Access Analyzer-Richtlinienprüfungen zu öffnen, öffnen Sie die VS Code-Befehlspalette **CTRL+Shift+P**, indem Sie auf, suchen und dann klicken **IAM Policy Checks**, um den Bereich IAM-Richtlinienprüfungen im VS Code-Editor zu öffnen.
3. Wählen Sie im Bereich IAM Policy Checks Ihren Dokumenttyp aus dem Dropdownmenü aus.
4. Wählen Sie im Abschnitt Benutzerdefinierte Richtlinienprüfungen die Option aus CheckAccessNotGranted.
5. Geben Sie in das Texteingabefeld eine durch Kommas getrennte Liste von Aktionen ein. Alternativ können Sie einen Dateipfad angeben, der auf eine Liste von Aktionen verweist.

Note

Der Dateipfad wird in der Regel von Ihrem Administrator oder Sicherheitsteam bereitgestellt und kann ein Systemdateipfad oder ein Amazon S3 S3-Bucket-URI sein. Um eine Amazon S3 S3-Bucket-URI verwenden zu können, muss Ihre aktuelle Rolle Zugriff auf den Amazon S3 S3-Bucket haben.

6. Wählen Sie die Schaltfläche „Benutzerdefinierte Richtlinienüberprüfung ausführen“.
7. Überprüfen Sie im Problembereich in VS Code die Ergebnisse Ihrer Richtlinienüberprüfung. Benutzerdefinierte Richtlinienprüfungen geben ein FAIL Ergebnis PASS oder ein Ergebnis zurück.
8. Aktualisieren Sie Ihre Richtlinie und wiederholen Sie diesen Vorgang. Führen Sie die CheckAccessNotGranted Prüfung erneut durch, bis sie zurückgegeben PASS wird.

Wird ausgeführt CheckNoNewAccess


CheckNoNewAccess ist eine benutzerdefinierte Richtlinienüberprüfung, mit der überprüft wird, ob Ihre Richtlinie im Vergleich zu einer Referenzrichtlinie neuen Zugriff gewährt.

Note

Mit jeder Überprüfung der benutzerdefinierten Richtlinien ist eine Gebühr verbunden. Einzelheiten zu den Preisen für benutzerdefinierte Policy-Checks finden Sie im [Preisleitfaden für AWS IAM Access Analyzer](#).

1. Öffnen Sie in VS Code eine unterstützte Datei, die AWS IAM-Richtlinien enthält, im VS Code-Editor.
2. Um die IAM Access Analyzer-Richtlinienprüfungen zu öffnen, öffnen Sie die VS Code-Befehlspalette **CTRL+Shift+P**, indem Sie auf, suchen und dann klicken **IAM Policy Checks**, um den Bereich IAM-Richtlinienprüfungen im VS Code-Editor zu öffnen.
3. Wählen Sie im Bereich IAM Policy Checks Ihren Dokumenttyp aus dem Dropdownmenü aus.
4. Wählen Sie im Abschnitt Benutzerdefinierte Richtlinienprüfungen die Option aus CheckNoNewAccess.

5. Geben Sie ein Referenzdokument für die JSON-Richtlinie ein. Alternativ können Sie einen Dateipfad angeben, der auf ein JSON-Richtliniendokument verweist.

 Note

Der Dateipfad wird in der Regel von Ihrem Administrator oder Sicherheitsteam bereitgestellt und kann ein Systemdateipfad oder ein Amazon S3 S3-Bucket-URI sein. Um eine Amazon S3 S3-Bucket-URI verwenden zu können, muss Ihre aktuelle Rolle Zugriff auf den Amazon S3 S3-Bucket haben.

6. Wählen Sie den Referenzrichtlinientyp aus, der dem Typ Ihres Referenzdokuments entspricht.
7. Wählen Sie die Schaltfläche „Benutzerdefinierte Richtlinienüberprüfung ausführen“.
8. Überprüfen Sie im Problembereich in VS Code die Ergebnisse Ihrer Richtlinienüberprüfung. Benutzerdefinierte Richtlinienprüfungen geben ein FAIL Ergebnis PASS oder ein Ergebnis zurück.
9. Aktualisieren Sie Ihre Richtlinie und wiederholen Sie diesen Vorgang. Führen Sie die CheckNoNewAccess Prüfung erneut durch, bis sie zurückgegeben PASS wird.

Arbeiten mit AWS IoT in der AWS Toolkit for Visual Studio Code

AWS IoT in der AWS Toolkit for Visual Studio Code ermöglicht Ihnen die Interaktion mit dem AWS IoT Service, während Unterbrechungen Ihres Arbeitsablaufs in VS Code minimiert werden. Dieses Benutzerhandbuch soll Ihnen den Einstieg in die Verwendung von AWS IoT Servicefunktionen, die in der AWS Toolkit for Visual Studio Code. Weitere Informationen zum AWS IoT-Service finden Sie im Entwicklerhandbuch [Was ist AWS IoT?](#)

AWS IoT-Voraussetzungen

Einstieg in die Verwendung von AWS IoT in der AWS Toolkit for VS Code, stellen Sie sicher, dass Ihr AWS Account und VS-Code erfüllen die Anforderungen in diesen Handbüchern:

- Für AWS Kontoanforderungen und AWS spezifische Benutzerberechtigungen für AWS IoT-Service finden Sie in der [Erste Schritte mit AWS IoT-Core](#) Entwicklerhandbuch.
- Spezifische Anforderungen für Toolkit for VS Code finden Sie in der [Toolkit for VS Code einrichten](#)-Benutzerhandbuch.

AWS IoT Elemente

AWS IoT verbinden von Geräten mit AWS Cloud-Dienste und Ressourcen. Sie können Ihre Geräte mit verbinden AWS IoT durch die Verwendung von Objekten namens Dinge. Ein Objekt ist eine Darstellung eines bestimmten Geräts oder einer logischen Entität. Es kann ein physisches Gerät oder ein Sensor sein (beispielsweise eine Glühbirne oder ein Wandschalter). Weitere Informationen über AWS IoT Dinge finden Sie im Entwicklerhandbuch [Geräteverwaltung mit AWS IoT](#).

Verwalten AWS IoT Dinge

Das Toolkit for VS Code hat mehrere Funktionen, die AWS IoT Ding-Management effizienter. Auf diese Weise können Sie das VS Code-Toolkit verwenden, um Ihre AWS IoT things:

- [Create a thing](#)
- [Attach a certificate to a thing](#)
- [Detach a certificate from a thing](#)
- [Delete a thing](#)

So erstellen Sie eine Objekt

1. aus AWS Explorer, erweitern Sie IoT Service-Überschrift und Kontextauswahl (rechte Maustaste) Elemente.
2. Klicken Sie auf Erstellen eines Objekts aus dem Kontext-Menü, um ein Dialogfenster zu öffnen.
3. Folgen Sie der Aufforderung, indem Sie einen Namen für Ihr IoT-Ding in das Feld Thing-Namefeld.
4. Wenn dies abgeschlossen ist, wird ein Thing-Icon gefolgt von dem von Ihnen angegebenen Namen wird in der Objektabschnitts erstellt.

So fügen Sie ein Zertifikat an ein Thing an

1. aus AWS Explorer, erweitern Sie IoT Service-Bereich.
2. Zeigen Sie unter dem Verzeichnis die folgenden Dateien an. Elemente-Unterabschnitt finden Sie den Sachewo Sie das Zertifikat anhängen.

3. Wählen Sie im Kontextmenü aus (rechte Maustaste) ausSacheund wähleAnfügen eines Zertifikateaus dem Kontextmenü, um eine Eingabeauswahl mit einer Liste Ihrer Zertifikate zu öffnen.
4. Wählen Sie aus der Liste ausZertifikat-IDdas entspricht dem Zertifikat, das Sie Ihrem Ding beifügen möchten.
5. Wenn dies abgeschlossen ist, können Sie auf Ihr Zertifikat in derAWSExplorer, als ein Element der Sache, an die Sie es angehängt haben.

So trennen Sie ein Zertifikat von einem Objekt

1. ausAWSExplorer, erweitern SieIoTService-Bereich
2. In derElemente-Unterabschnitt finden Sie denSachevon dem Sie ein Zertifikat trennen möchten.
3. Wählen Sie im Kontextmenü aus (rechte Maustaste) ausSacheund wähleTrennen eines Zertifikatsaus dem Kontext-Menü.
4. Wenn dies abgeschlossen ist, wird das gelöste Zertifikat nicht mehr unter dem Ding in derAWSExplorer, aber es wird weiterhin über denZertifikateUnterabschnitt.

So löschen Sie ein Objekt löschen

1. ausAWSExplorer, erweitern SieIoTService-Bereich.
2. In derElemente-Unterabschnitt finden Sie denSachelöschen möchten.
3. Wählen Sie das Objekt aus (rechte Maustaste) und wählen SieObjekt löschenaus dem Kontext-Menü, um es zu löschen.
4. Wenn dies abgeschlossen ist, ist das gelöschte Ding nicht mehr imElementeUnterabschnitt.

Note

Hinweis: Sie können nur Dinge löschen, an die kein Zertifikat angehängt ist.

AWS IoT-Zertifikate

Zertifikate sind eine übliche Methode, um eine sichere Verbindung zwischen IhremAWS IoT-Dienste und Geräte. X.509-Zertifikate sind digitale Zertifikate, die den X.509-Standard für

Public-Key-Infrastrukturen verwenden, um einen öffentlichen Schlüssel mit einer Identität in einem Zertifikat zu verknüpfen. Weitere Informationen über AWS IoT-Zertifikate finden Sie im Entwicklerhandbuch [Authentifizierung \(IoT\)](#).

Verwalten von Zertifikaten

Das VS Code Toolkit bietet Ihnen eine Vielzahl von Möglichkeiten zur Verwaltung Ihrer AWS IoT-Zertifikate, direkt von der AWS-Explorer.

- [Create a certificate](#)
- [Change a certificate status](#)
- [Attach a policy to a certificate](#)
- [Delete a certificate](#)

So erstellen Sie ein AWS IoT-Zertifikat

Ein X.509-Zertifikat kann verwendet werden, um eine Verbindung mit Ihrer Instanz von AWS IoT.

1. aus AWS Explorer, erweitern Sie den IoT-Service-Bereich und Kontextauswahl (rechte Maustaste) Zertifikate.
2. Klicken Sie auf Erstellen eines Zertifikats aus dem Kontext-Menü, um ein Dialogfenster zu öffnen.
3. Wählen Sie ein Verzeichnis in Ihrem lokalen Dateisystem aus, um Ihr RSA-Schlüsselpaar und das X.509-Zertifikat zu speichern.

Note

- Die Standarddateinamen enthalten die Zertifikat-ID als Präfix.
- Nur das X.509-Zertifikat wird mit Ihrem AWS-Konto, über das AWS IoT-Service-Service.
- Ihr RSA-Schlüsselpaar kann nur einmal ausgestellt werden. Speichern Sie es an einem sicheren Ort in Ihrem Dateisystem, wenn Sie dazu aufgefordert werden.
- Wenn entweder das Zertifikat oder das key pair zu diesem Zeitpunkt nicht in Ihrem Dateisystem gespeichert werden können, wird das AWS Toolkit das Zertifikat aus Ihrem AWS-Konto löschen.

So ändern Sie den Status eines Zertifikats

Der Status eines einzelnen Zertifikats wird neben seiner ID in der AWSExplorer und kann auf aktiv, inaktiv oder gesperrt gesetzt werden.

Note

- Ihr Zertifikat benötigt einen aktiv-Status, bevor Sie es verwenden können, um das -Gerät mit AWS IoT Service-Service.
- Importieren in &S3;inaktiv Zertifikat kann aktiviert werden, unabhängig davon, ob es zuvor deaktiviert wurde oder standardmäßig inaktiv ist.
- Ein Zertifikat, das widerrufen kann nicht reaktiviert werden.

1. aus AWSExplorer, erweitern Sie IoT Service-Bereich.
 2. In der Zertifikate-Unterabschnitt das Zertifikat, das geändert werden soll.
 3. Wählen Sie das Zertifikat aus (klicken Sie mit der rechten Maustaste) aus, um ein Kontextmenü zu öffnen, in dem die für dieses Zertifikat verfügbaren Statusänderungsoptionen angezeigt werden.
- Wenn ein Zertifikat den Status hat inaktiv, wählen aktivieren aus aktiv.
 - Wenn ein Zertifikat den Status hat aktiv, wählen deaktivieren aus inaktiv.
 - Wenn ein Zertifikat entweder einen aktiv oder inaktiv Status, wählen widerrufen aus widerrufen.

Note

Jede dieser Statusänderungsaktionen ist auch verfügbar, wenn Sie ein Zertifikat auswählen, das an ein Ding angehängt ist, während es in der Elemente-Unterabschnitt.

So hängen Sie eine IoT-Richtlinie an ein Zertifikat an

1. aus AWSExplorer, erweitern Sie IoT Service-Bereich.
2. In der Zertifikate-Unterabschnitt das Zertifikat, das geändert werden soll.

3. Wählen Sie das Zertifikat aus (rechte Maustaste) und `ausRichtlinie` anfügen vonaus dem Kontextmenü, um eine Eingabeauswahl mit einer Liste Ihrer verfügbaren Richtlinien zu öffnen.
4. Wählen Sie die Richtlinie aus, die Sie an das Zertifikat anfügen möchten.
5. Wenn dies abgeschlossen ist, wird die von Ihnen ausgewählte Richtlinie dem Zertifikat als Untermenüpunkt hinzugefügt.

So trennen Sie eine IoT-Richtlinie von einem Zertifikat

1. `ausAWSExplorer`, erweitern Sie `IoTService`-Bereich.
2. In der `Zertifikate`-Unterabschnitt das Zertifikat, das geändert werden soll.
3. Erweitern Sie das Zertifikat und suchen Sie die Richtlinie, die Sie trennen möchten.
4. Wählen Sie die Richtlinie aus (rechte Maustaste) und `ausAufhebung der Verknüpfung` aus
5. Wenn dies abgeschlossen ist, ist die Richtlinie kein Element mehr, auf das über Ihr Zertifikat zugegriffen werden kann, sondern über die `Richtlinie`-Unterabschnitt.

So löschen Sie ein Zertifikat

1. `ausAWSExplorer`, erweitern Sie `IoTService`-personalisieren.
2. In der `Zertifikate`-Unterabschnitt das Zertifikat, das Sie löschen möchten.
3. Wählen Sie das Zertifikat aus (rechte Maustaste) und `ausZertifikat löschen` aus

Note

Sie können ein Zertifikat nicht löschen, wenn es an ein Thing angefügt ist oder einen aktiven Status hat. Sie können ein Zertifikat löschen, das Richtlinien angehängt hat.

AWS IoT-Richtlinien

AWS IoT Kernrichtlinien werden in JSON-Dokumenten definiert, die jeweils eine oder mehrere Richtlinienanweisungen enthalten. Richtlinien definieren, wie AWS IoT, AWS und das -Gerät kann miteinander interagieren. Weitere Informationen zum Erstellen eines Richtlinien-Dokuments finden Sie im Entwicklerhandbuch [IoT-Richtlinien](#).

Note

Bei benannten Richtlinien wird Versionierung verwendet, sodass Sie sie zurücksetzen können. In der AWS Explorer, Ihre IoT-Richtlinien sind unter der Richtlinien-Unterabschnitt, im IoT-Dienst. Sie können Richtlinienversionen anzeigen, indem Sie eine Richtlinie erweitern. Die Standardversion ist mit einem Sternchen gekennzeichnet.

Verwalten von -Richtlinien

Das Toolkit for VS Code bietet Ihnen mehrere Möglichkeiten zur Verwaltung der AWS IoT Service-Richtlinien. Auf diese Weise können Sie Ihre Richtlinien direkt im AWS Explorer im VS-Code:

- [Create a policy](#)
- [Upload a new policy version](#)
- [Edit a policy version](#)
- [Change the policy version default](#)
- [Change the policy version default](#)

So erstellen Sie eine AWS IoT-Richtlinie

Note

Sie können eine neue Richtlinie über das AWS Explorer, aber das JSON-Dokument, das die Richtlinie definiert, muss bereits in Ihrem Dateisystem vorhanden sein.

1. aus AWS Explorer, erweitern Sie IoT Service-Bereich.
2. Wählen Sie im Kontextmenü aus (rechte Maustaste) aus Richtlinien Unterabschnitt und wählen Richtlinie aus Dokument erstellen aus Richtliniennamen-Eingabefeld.
3. Geben Sie einen Namen ein und folgen Sie den Anweisungen, um einen Dialog zu öffnen, in dem Sie aufgefordert werden, ein JSON-Dokument aus Ihrem Dateisystem auszuwählen.
4. Wählen Sie die JSON-Datei aus, die Ihre Richtliniendefinitionen enthält. Die Richtlinie ist im AWS Explorer, wenn dies abgeschlossen ist.

So laden Sie ein neues AWS IoT Richtlinienversion

Eine neue Version einer Richtlinie kann erstellt werden, indem ein JSON-Dokument in die Richtlinie hochgeladen wird.

Note

Das neue JSON-Dokument muss in Ihrem Dateisystem vorhanden sein, um eine neue Version mit dem AWS-Explorer.

1. aus dem AWS Explorer, erweitern Sie den IoT Service-Bereich.
2. Erweitern Sie Richtlinien aus dem AWS IoT Politik
3. Wählen Sie die Richtlinie aus, die Sie aktualisieren möchten, und wählen Sie Neue Version aus Dokument erstellen.
4. Wenn das Dialogfeld geöffnet wird, wählen Sie die JSON-Datei aus, die die Aktualisierungen Ihrer Richtliniendefinitionen enthält.
5. Auf die neue Version kann über Ihre Richtlinie in dem AWS-Explorer.

Bearbeiten einer AWS IoT Richtlinienversion

Ein Richtlinien dokument kann mit VS-Code geöffnet und bearbeitet werden. Wenn Sie mit der Bearbeitung des -Dokuments fertig sind, können Sie es in Ihrem Dateisystem speichern. Anschließend können Sie es in Ihrem AWS IoT Service von dem AWS-Explorer.

1. aus dem AWS Explorer, erweitern Sie den IoT Service-Bereich.
2. Erweitern Sie Richtlinien und suchen Sie die Richtlinie, die Sie aktualisieren möchten. Richtlinie aus Dokument erstellen aus Richtlinienname-Eingabefeld.
3. Erweitern Sie die Richtlinie, die Sie aktualisieren möchten, und wählen Sie dann die Version der Richtlinie, die Sie bearbeiten möchten, mit der rechten Maustaste aus.
4. Klicken Sie auf Anzeigen aus dem Kontextmenü, um die Version der Richtlinie in VS Code zu öffnen
5. Nehmen Sie beim Öffnen des Richtlinien dokument die gewünschten Änderungen vor und speichern Sie sie.

Note

Zu diesem Zeitpunkt werden die Änderungen, die Sie an der Richtlinie vorgenommen haben, nur in Ihrem lokalen Dateisystem gespeichert. Um die Version zu aktualisieren und mit dem AWSExplorer aus. Wiederholen Sie die im [Upload a new policy version-](#)Verfahren.

So wählen Sie eine neue Standardversion der Richtlinie aus

1. aus AWSExplorer, erweitern Sie den IoTService-Bereich.
2. Erweitern Sie die Richtlinien und suchen Sie die Richtlinie, die Sie aktualisieren möchten.
3. Erweitern Sie die Richtlinie, die Sie aktualisieren möchten, und wählen Sie dann die Version der Richtlinie, die Sie festlegen möchten, im Kontext (mit der rechten Maustaste) aus, und wählen Sie als Standard festlegen.
4. Wenn dies abgeschlossen ist, befindet sich neben der neuen Standardversion, die Sie ausgewählt haben, ein Stern.

So löschen Sie -Richtlinien

Note

Bevor Sie eine Richtlinie oder eine Richtlinienversion löschen können, müssen Bedingungen erfüllt sein.

- Sie können eine Richtlinie nicht löschen, wenn sie an ein Zertifikat angefügt ist.
- Sie können eine Richtlinie nicht löschen, wenn sie keine Standardversionen hat.
- Sie können die Standardversion einer Richtlinie nur löschen, wenn eine neue Standardversion ausgewählt wurde oder die gesamte Richtlinie gelöscht wird.
- Bevor Sie eine gesamte Richtlinie löschen können, muss zuerst die gesamte Nicht-Standardversion dieser Richtlinie gelöscht werden.

1. aus AWSExplorer, erweitern Sie den IoTService-Bereich.

2. Erweitern Sie die Richtlinie und suchen Sie die Richtlinie, die Sie aktualisieren möchten.
3. Erweitern Sie die Richtlinie, die Sie aktualisieren möchten, und wählen Sie dann (Rechtsklick) die Version der Richtlinie aus, die Sie löschen möchten, und wählen Sie Löschen.
4. Wenn eine Version gelöscht wird, ist sie im Explorer nicht mehr sichtbar.
5. Wenn die einzige Version, die für eine Richtlinie übrig bleibt, die Standardversion ist, können Sie die übergeordnete Richtlinie mit einem Kontext auswählen (mit der rechten Maustaste klicken) und Löschen, um es zu löschen.

Arbeiten mit AWS Lambda-Funktionen

Das AWS Toolkit for Visual Studio Code bietet Unterstützung für [AWS Lambda](#)-Funktionen. Mit dem Toolkit for VS Code können Sie Code für Lambda-Funktionen erstellen, die Bestandteil von [serverlose Anwendungen](#) aus. Darüber hinaus können Sie Lambda-Funktionen entweder lokal oder auf AWS aufrufen.

Lambda ist ein vollständig verwalteter Computing-Service, der Ihren Code beim Eintreten bestimmter Ereignisse ausführt, die von einem benutzerdefinierten Code oder von verschiedenen AWS-Services wie Amazon Simple Storage Service (Amazon S3, Amazon DynamoDB, Amazon Kinesis, Amazon Simple Notification Service (Amazon SNS) Amazon Cognito erzeugt wurden.

Themen

- [Interaktion mit Remote-Lambda-Funktionen](#)

Interaktion mit Remote-Lambda-Funktionen

Mit dem Toolkit for VS Code können Sie mit interagieren [AWS Lambda](#)-Funktionen auf verschiedene Arten, wie weiter unten in diesem Thema beschrieben.

Weitere Informationen zu Lambda finden Sie im [AWS Lambda Entwicklerhandbuch](#).

Note

Wenn Sie bereits Lambda-Funktionen mit der AWS Management Console oder auf andere Weise erstellt haben, können Sie sie aus dem Toolkit aufrufen. So erstellen Sie eine neue - Funktion (mit VS Code), die Sie bereitstellen können AWS Lambda, musst du zuerst [Erstellen einer serverless Anwendung](#) aus.

Themen

- [Voraussetzungen](#)
- [Aufruf einer Lambda-Funktion](#)
- [Löschen einer Lambda-Funktion](#)
- [Importieren einer Lambda-Funktion](#)
- [Laden Sie eine Lambda-Funktion hoch](#)

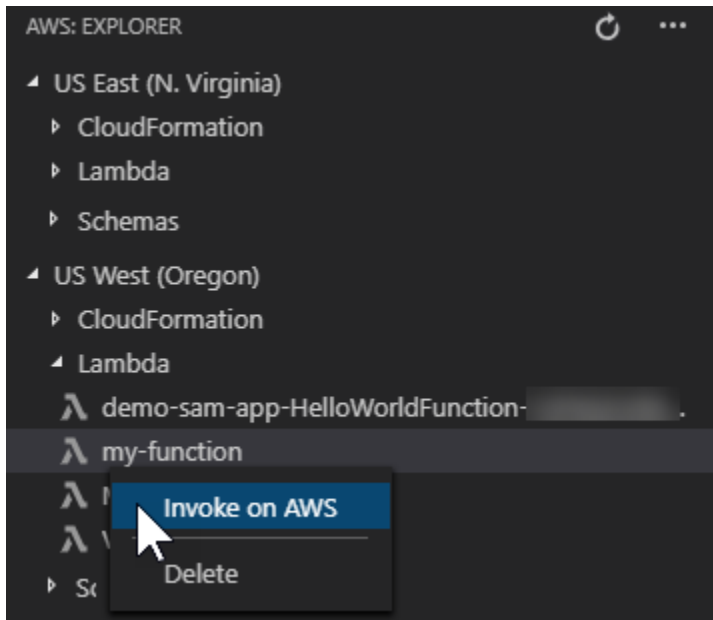
Voraussetzungen

- Stellen Sie sicher, dass Ihr System die Voraussetzungen erfüllt, die unter [Installieren des Toolkit for VS Code](#) aus.
- Stellen Sie sicher, dass die Anmeldeinformationen, die Sie in [Authentifizierung und Zugriff](#) konfiguriert haben, den entsprechenden Lese-/Schreibzugriff auf den AWS Lambda-Service enthalten. Wenn im AWS-WS Explorer unter Lambda als Meldung etwa „Fehler beim Laden von Lambda-Ressourcen“ angezeigt wird, überprüfen Sie die Berechtigungen, die diesen Anmeldeinformationen zugeordnet sind. Änderungen, die Sie an Berechtigungen vornehmen, dauern einige Minuten, bis sie sich auf AWS Explorer im VS-Code.

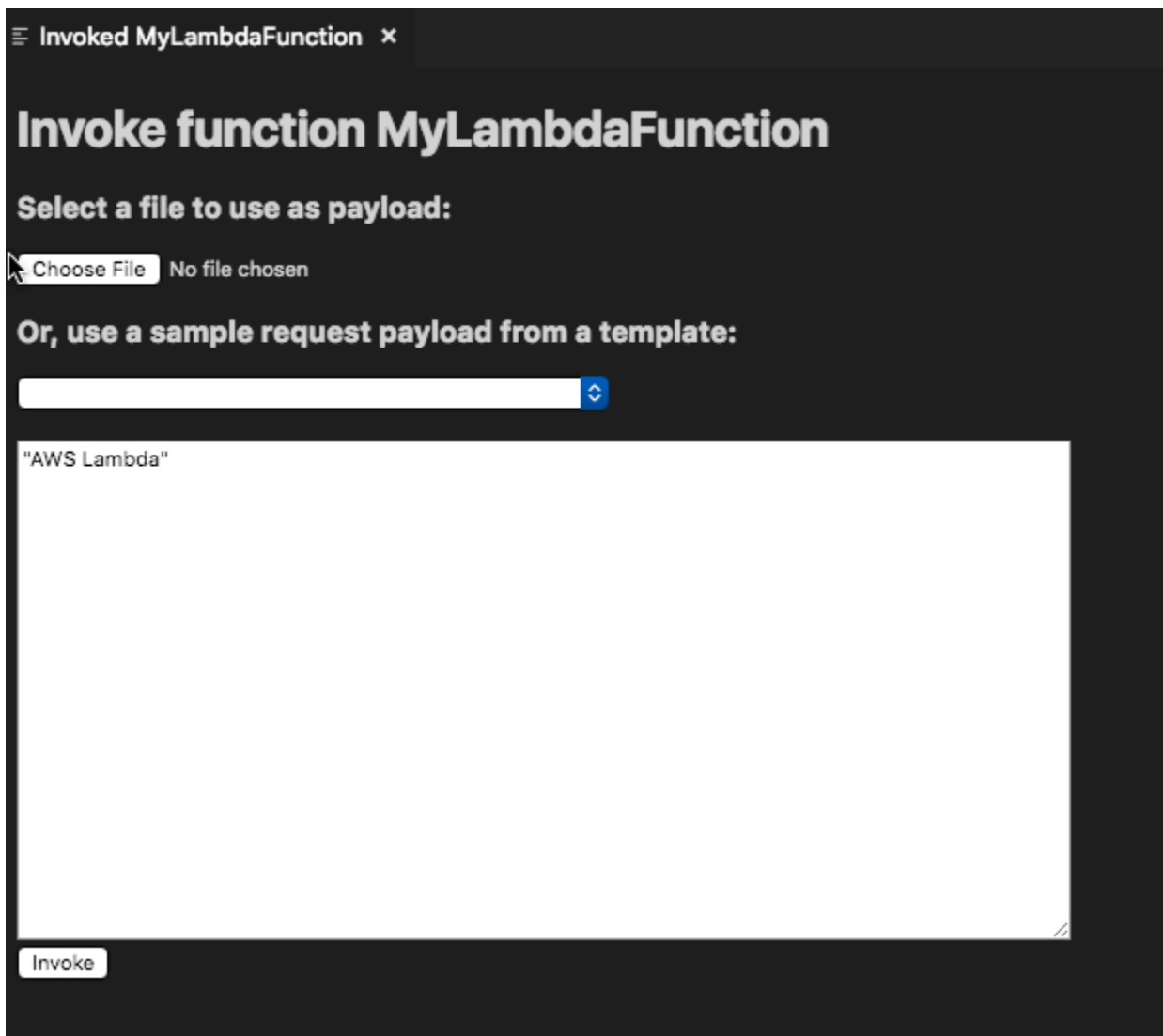
Aufruf einer Lambda-Funktion

Sie können eine Lambda-Funktion aufrufen auf AWS aus dem Toolkit for VS Code.

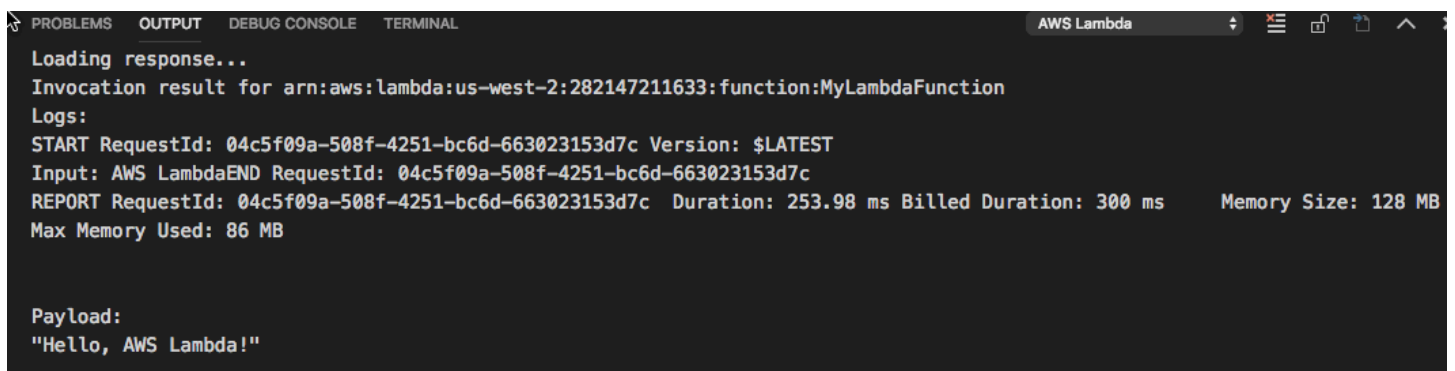
1. Wählen Sie im AWS Explorer den Namen der Lambda-Funktion aus, die Sie aufrufen möchten, und öffnen Sie das Kontextmenü.



2. Wählen Sie Invoke on AWS (In AWS aufrufen).
3. Geben Sie im aufrufenden Fenster, das sich öffnet, die Eingabe ein, die Ihre Lambda-Funktion benötigt wird. Die Lambda-Funktion kann beispielsweise eine Zeichenfolge als Eingabe erfordern, wie im Textfeld gezeigt.



Sie können die Ausgabe der Lambda-Funktion genauso wie bei jedem anderen Projekt mithilfe von VS Code anzeigen.



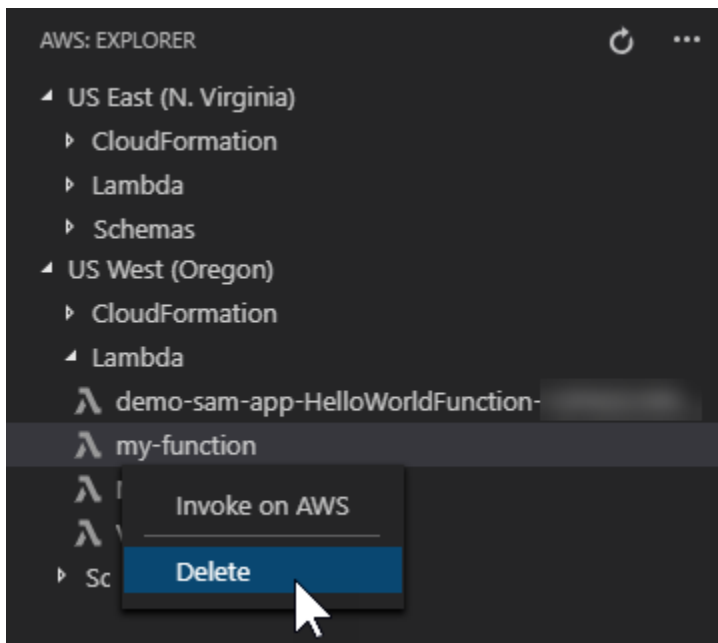
Löschen einer Lambda-Funktion

Sie können eine Lambda-Funktion auch über dasselbe Kontextmenü löschen.

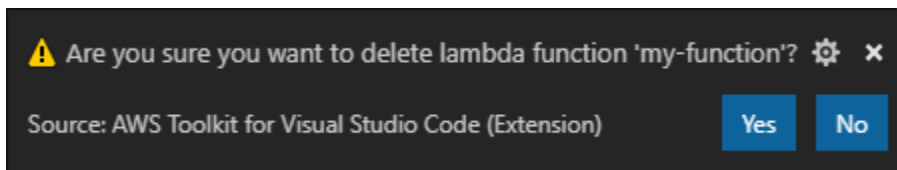
⚠ Warning

Verwenden Sie dieses Verfahren nicht, um -Funktionen zu löschen, die [AWS CloudFormation](#) zugeordnet sind (z. B. die Lambda-Funktion, die beim [Erstellen einer serverless Anwendung](#) weiter oben in diesem Handbuch erstellt wurde). Diese Funktionen müssen über den AWS CloudFormation-Stack gelöscht werden.

1. Wählen Sie im AWS Explorer den Namen der Lambda-Funktion aus, die Sie löschen möchten, und öffnen Sie das Kontextmenü.



2. Wählen Sie Delete (Löschen).
3. Wählen Sie in der angezeigten Meldung Yes (Ja) aus, um das Löschen zu bestätigen.



Nachdem die Funktion gelöscht wurde, wird sie nicht mehr im AWS Explorer aufgeführt.

Importieren einer Lambda-Funktion

Sie können Code aus einer Remote-Lambda-Funktion zum Bearbeiten und Debuggen in Ihren VS-Code-Workspace importieren.

Note

Das Toolkit unterstützt nur das Importieren von Lambda-Funktionen mithilfe der unterstützten Node.js- und Python-Laufzeiten.

1. In der **AWS Explorer** Wählen Sie den Namen der Lambda-Funktion aus, die Sie importieren möchten, und öffnen Sie das Kontextmenü.
2. Klicken Sie auf **Importieren Sie...**
3. Wählen Sie einen Ordner aus, in den der Lambda-Code importiert werden soll. Ihrem aktuellen Arbeitsbereich werden Ordner außerhalb des aktuellen Arbeitsbereichs hinzugefügt.
4. Nach dem Herunterladen fügt das Toolkit den Code zu Ihrem Workspace hinzu und öffnet die Datei mit dem Lambda-Handler-Code. Das Toolkit erstellt auch eine **Startkonfiguration**, das im VS-Code-Run Panel angezeigt wird, sodass Sie die Lambda-Funktion lokal ausführen und debuggen können **AWS Serverless Application Model** aus. Weitere Informationen zur Verwendung von **AWS SAM** finden Sie unter [the section called “Eine serverlose Anwendung über eine Vorlage ausführen und debuggen \(lokal\)”](#).

Laden Sie eine Lambda-Funktion hoch

Sie können vorhandene Lambda Funktionen mit lokalem Code aktualisieren. Das Aktualisieren von Code auf diese Weise verwendet nicht die **AWS SAM CLI** für die Bereitstellung und erstellt keine **AWS CloudFormation Stacks**. Diese Funktionalität kann eine Lambda Funktion mit jeder Laufzeit hochladen, die von Lambda unterstützt wird.

Warning

Das Toolkit kann nicht überprüfen, ob Ihr Code funktioniert. Stellen Sie sicher, dass der Code funktioniert, bevor Sie die Lambda-Funktionen der Produktion aktualisieren.

1. In der **AWS Explorer** Wählen Sie den Namen der Lambda-Funktion aus, die Sie importieren möchten, und öffnen Sie das Kontextmenü.
2. Klicken Sie auf **Hochladen von Lambda...**
3. Wählen Sie aus den drei Optionen zum Hochladen Ihrer Lambda Funktion. Zu den Optionen gehören:

Hochladen eines vorgefertigten ZIP-Archivs

- Klicken Sie auf **Zip-Archiv** aus dem Schnellauswahl-Menü.
- Wählen Sie eine ZIP-Datei aus Ihrem Dateisystem und bestätigen Sie den Upload mit dem modalen Dialog. Dadurch wird die ZIP-Datei wie sie ist hochgeladen und das Lambda nach der Bereitstellung sofort aktualisiert.

Upload eines Verzeichnisses

- Klicken Sie auf **Directory** aus dem Schnellauswahl-Menü.
- Wählen Sie ein Verzeichnis aus Ihrem Dateisystem aus.
- Klicken Sie auf **Nein** Wenn Sie aufgefordert werden, das Verzeichnis zu erstellen, bestätigen Sie den Upload mit dem modalen Dialog. Dadurch wird das Verzeichnis wie es ist hochgeladen und das Lambda nach der Bereitstellung sofort aktualisiert.

Erstellen und Hochladen eines Verzeichnisses

Note

Dies erfordert **AWS SAM CLI**.

- Klicken Sie auf **Directory** aus dem Schnellauswahl-Menü.
- Wählen Sie ein Verzeichnis aus Ihrem Dateisystem aus.
- Klicken Sie auf **Ja** Wenn Sie aufgefordert werden, das Verzeichnis zu erstellen, bestätigen Sie den Upload mit dem modalen Dialog. Dies erstellt den Code im Verzeichnis mit dem **AWS SAM CLI** `aws sam build` und aktualisiert sofort das Lambda nach der Bereitstellung.

Note

Das Toolkit warnt Sie, wenn es vor dem Hochladen keinen übereinstimmenden Handler erkennen kann. Wenn Sie den Handler ändern möchten, der an die Lambda-Funktion gebunden ist, können Sie dies über [AWS Management Console](#) oder das [AWS CLI](#) ausführen.

Amazon Redshift im Toolkit for VS Code

Amazon Redshift ist ein vollständig verwalteter Data-Warehouse-Service in Petabytegröße in der Cloud. Ausführliche Informationen zum Amazon Redshift Redshift-Service finden Sie im Inhaltsverzeichnis der [Amazon Redshift Redshift-Benutzerhandbücher](#).

In den folgenden Themen wird beschrieben, wie Sie mit Amazon Redshift arbeiten. AWS Toolkit for Visual Studio Code

Themen

- [Arbeiten mit Amazon Redshift aus dem Toolkit for VS Code](#)

Arbeiten mit Amazon Redshift aus dem Toolkit for VS Code

In den folgenden Abschnitten wird beschrieben, wie Sie mit der Arbeit mit Amazon Redshift beginnen können. AWS Toolkit for Visual Studio Code

Ausführliche Informationen zum Amazon Redshift Redshift-Service finden Sie in den Themen des [Amazon Redshift Redshift-Benutzerhandbuchs](#).

Erste Schritte

Bevor Sie von der aus mit der Arbeit mit Amazon Redshift beginnen [AWS Toolkit for Visual Studio Code](#), müssen die folgenden Anforderungen erfüllt sein.

1. Sie sind über das Toolkit mit Ihren AWS Konten verbunden. Weitere Informationen zum Herstellen einer Verbindung mit Ihrem AWS Konto über das Toolkit finden Sie im AWS Thema [Herstellen einer Verbindung zu](#) in diesem Benutzerhandbuch.
2. Sie haben ein bereitgestelltes oder serverloses Data Warehouse erstellt.

Wenn Sie noch keinen Amazon Redshift Serverless oder einen von Amazon Redshift bereitgestellten Cluster erstellt haben, beschreiben die folgenden Verfahren, wie Sie ein Data Warehouse mit einem Beispieldatensatz von der Konsole aus erstellen. AWS

Ein bereitgestelltes Data Warehouse erstellen

Weitere Informationen zur Erstellung eines von Amazon Redshift bereitgestellten Cluster-Data Warehouse finden Sie im Thema Amazon Redshift [Redshift-Beispielcluster erstellen im Amazon Redshift Redshift-Benutzerhandbuch](#) für die ersten Schritte.

1. Melden Sie sich über Ihren bevorzugten Internetbrowser bei der AWS Management Console an und öffnen Sie die Amazon Redshift Redshift-Konsole unter <https://console.aws.amazon.com/redshift/>.
2. Wählen Sie in der Amazon Redshift Redshift-Konsole das Dashboard Provisioned Clusters aus.
3. Wählen Sie im Dashboard für bereitgestellte Cluster die Schaltfläche Cluster erstellen, um den Bereich Cluster erstellen zu öffnen.
4. Füllen Sie die erforderlichen Felder im Abschnitt Clusterkonfiguration aus.
5. Wählen Sie im Abschnitt Beispieldaten das Feld Beispieldaten laden aus, um den Beispieldatensatz **Dev** mit dem **public** Schema **Tickit** in die Standarddatenbank zu laden.
6. Geben Sie im Abschnitt Datenbankkonfigurationen Werte für die Felder Admin-Benutzername und Admin-Benutzerkennwort ein.
7. Wählen Sie Create Cluster aus, um Ihr bereitgestelltes Data Warehouse zu erstellen.

Ein serverloses Data Warehouse erstellen

Weitere Informationen zur Erstellung eines Amazon Redshift Serverless Data Warehouse finden Sie im Abschnitt [Creating a Data Warehouse with Amazon Redshift Serverless](#) im Amazon Redshift Getting Started User Guide.

1. Melden Sie sich über Ihren bevorzugten Internetbrowser bei der AWS Management Console an und öffnen Sie die Amazon Redshift Redshift-Konsole unter <https://console.aws.amazon.com/redshift/>.
2. Wählen Sie in der Amazon Redshift-Konsole die Schaltfläche Amazon Redshift Serverless testen, um den Bereich Erste Schritte mit Amazon Redshift Serverless zu öffnen.
3. Wählen Sie im Abschnitt Konfigurationen die Option Standardeinstellungen verwenden aus.

4. Wählen Sie unten im Bereich Erste Schritte mit Amazon Redshift Serverless die Option Konfiguration speichern aus, um ein serverloses Data Warehouse mit Standardeinstellungen für Arbeitsgruppe, Namespace, Anmeldeinformationen und Verschlüsselung zu erstellen.

Über das Toolkit eine Verbindung zu einem Data Warehouse herstellen

Es gibt 3 Methoden, um über das Toolkit eine Verbindung zu einer Datenbank herzustellen:

- Datenbank-Benutzername und Passwort
- AWS-Secrets Manager
- Temporäre Anmeldeinformationen

Gehen Sie wie folgt vor, um über das Toolkit eine Verbindung zu einer Datenbank herzustellen, die sich in einem vorhandenen bereitgestellten Cluster oder serverlosen Data Warehouse befindet.

Important

Wenn Sie die Schritte im Abschnitt Voraussetzungen dieses Benutzerhandbuchthemas abgeschlossen haben und Ihr Data Warehouse im Toolkit-Explorer nicht sichtbar ist, stellen Sie sicher, dass Sie von der richtigen AWS Region im Explorer aus arbeiten.

Stellen Sie mithilfe der Methode Datenbankbenutzername und Passwort eine Verbindung zu Ihrem Data Warehouse her

1. Erweitern Sie im Toolkit-Explorer den Ort, AWS-Region an dem sich Ihr Data Warehouse befindet.
2. Erweitern Sie Redshift und wählen Sie Ihr Data Warehouse aus, um das Dialogfeld „Verbindungstyp auswählen“ in VS Code zu öffnen.
3. Wählen Sie im Dialogfeld „Verbindungstyp auswählen“ den Datenbankbenutzernamen und das Kennwort aus und geben Sie die Informationen ein, die für jede der Eingabeaufforderungen erforderlich sind.
4. Ihre verfügbaren Datenbanken, Tabellen und Schemas sind im Toolkit-Explorer sichtbar, wenn das Toolkit eine Verbindung zu Ihrem Data Warehouse herstellt und der Vorgang abgeschlossen ist.

Stellen Sie eine Verbindung zu Ihrem Data Warehouse her mit AWS Secrets Manager

Note

Für diesen Vorgang ist ein AWS Secrets Manager-Datenbankgeheimnis erforderlich. Anweisungen zum Einrichten eines geheimen Datenbankschlüssels finden Sie im AWS Secrets Manager-Benutzerhandbuch unter [Create an AWS Secrets Manager database secret](#).

1. Erweitern Sie im Toolkit-Explorer den Ort, AWS-Region an dem sich Ihr Data Warehouse befindet.
2. Erweitern Sie Redshift und wählen Sie Ihr Data Warehouse aus, um das Dialogfeld „Verbindungstyp auswählen“ in VS Code zu öffnen.
3. Wählen Sie im Dialogfeld „Verbindungstyp auswählen“ die Option Secrets Manager und geben Sie die Informationen ein, die für jede der Eingabeaufforderungen erforderlich sind.
4. Ihre verfügbaren Datenbanken, Tabellen und Schemas sind im Toolkit-Explorer sichtbar, wenn das Toolkit eine Verbindung zu Ihrem Data Warehouse herstellt und der Vorgang abgeschlossen ist.

Stellen Sie mit temporären Anmeldeinformationen eine Verbindung zu Ihrem Data Warehouse her

1. Erweitern Sie im Toolkit-Explorer die AWS Region, in der Ihr Data Warehouse existiert.
2. Erweitern Sie Redshift und wählen Sie Ihr Data Warehouse aus, um das Dialogfeld „Verbindungstyp auswählen“ in VS Code zu öffnen.
3. Wählen Sie im Dialogfeld „Verbindungstyp auswählen“ die Option Temporäre Anmeldeinformationen aus und geben Sie die Informationen ein, die für jede der Eingabeaufforderungen erforderlich sind.
4. Ihre verfügbaren Datenbanken, Tabellen und Schemas sind im Toolkit-Explorer sichtbar, wenn das Toolkit eine Verbindung zu Ihrem Data Warehouse herstellt und der Vorgang abgeschlossen ist.

Die Verbindung zu Ihrem Data Warehouse bearbeiten

Sie können die Verbindung zu Ihrem Data Warehouse bearbeiten, um zu ändern, mit welcher Datenbank eine Verbindung hergestellt werden soll.

1. Erweitern Sie im Toolkit-Explorer den Ort, AWS-Region an dem sich Ihr Data Warehouse befindet.
2. Erweitern Sie Redshift, klicken Sie mit der rechten Maustaste auf das Data Warehouse, mit dem Sie verbunden sind, wählen Sie Verbindung bearbeiten und geben Sie den Namen der Datenbank ein, zu der Sie eine Verbindung herstellen möchten.
3. Ihre verfügbaren Datenbanken, Tabellen und Schemas sind im Toolkit-Explorer sichtbar, wenn das Toolkit eine Verbindung zu Ihrem Data Warehouse herstellt und der Vorgang abgeschlossen ist.

Die Verbindung zu Ihrem Data Warehouse wird gelöscht

1. Erweitern Sie im Toolkit-Explorer den Ort, AWS-Region an dem sich Ihr Data Warehouse befindet.
2. Erweitern Sie Redshift, klicken Sie mit der rechten Maustaste auf das Data Warehouse mit der Verbindung, die Sie löschen möchten, und wählen Sie Verbindung löschen. Dadurch werden die verfügbaren Datenbanken, Tabellen und Schemas aus dem Toolkit-Explorer entfernt.
3. Um die Verbindung zu Ihrem Data Warehouse wiederherzustellen, wählen Sie Klicken, um eine Verbindung herzustellen, und geben Sie die Informationen ein, die für die einzelnen Eingabeaufforderungen erforderlich sind. Standardmäßig wird beim erneuten Herstellen der Verbindung die vorherige Authentifizierungsmethode verwendet, um eine Verbindung mit dem Data Warehouse herzustellen. Um eine andere Methode zu verwenden, klicken Sie im Dialogfeld auf den Zurück-Pfeil, bis Sie zur Authentifizierungsaufforderung gelangen.

SQL-Anweisungen ausführen

Die folgenden Verfahren beschreiben, wie Sie SQL-Anweisungen in Ihrer Datenbank von erstellen und ausführenAWS Toolkit for Visual Studio Code.

Note

Um die Schritte in den folgenden Verfahren durchzuführen, müssen Sie zunächst den Abschnitt Verbindung zu einem Data Warehouse mithilfe des Toolkits herstellen, der sich in diesem Thema im Benutzerhandbuch befindet, abschließen.

1. Erweitern Sie im Toolkit-Explorer Redshift und dann das Data Warehouse, das die Datenbank enthält, die Sie abfragen möchten.
2. Wählen Sie Create-Notebook, um einen Dateinamen und einen Speicherort für das lokale Speichern Ihres Notizbuchs anzugeben, und wählen Sie dann OK, um das Notizbuch in Ihrem VS Code-Editor zu öffnen.
3. Geben Sie im VS Code-Editor die SQL-Anweisungen ein, die Sie in diesem Notizbuch speichern möchten.
4. Wählen Sie die Schaltfläche Alle ausführen, um die von Ihnen eingegebenen SQL-Anweisungen auszuführen.
5. Die Ausgabe Ihrer SQL-Anweisungen wird unter den von Ihnen eingegebenen Anweisungen angezeigt.

Markdown zu einem Notizbuch hinzufügen

1. Wählen Sie in Ihrem Notizbuch im VS Code Editor die Markdown-Schaltfläche, um Ihrem Notizbuch eine Markdown-Zelle hinzuzufügen.
2. Geben Sie Ihren Markdown in die dafür vorgesehene Zelle ein.
3. Die Markdown-Zelle kann mit den Editor-Tools in der oberen rechten Ecke der Markdown-Zelle bearbeitet werden.

Code zu einem Notizbuch hinzufügen

1. Wählen Sie in Ihrem Notizbuch im VS Code-Editor die Code-Schaltfläche, um Ihrem Notizbuch eine Code-Zelle hinzuzufügen.
2. Geben Sie Ihren Code in die dafür vorgesehene Zelle ein.
3. Sie können wählen, ob Sie Ihren Code über oder unter der Codezelle ausführen möchten, indem Sie in den Tools des Zelleneditors in der oberen rechten Ecke der Codezelle die entsprechende Schaltfläche auswählen.

Arbeiten mit Amazon S3

Amazon Simple Storage Service (Amazon S3) ist ein skalierbarer Datenspeicherservice. AWS Toolkit for Visual Studio Code Damit können Sie Ihre Amazon S3 S3-Objekte und -Ressourcen direkt von VS Code aus verwalten.

Ausführliche Informationen zum Amazon S3 S3-Service finden Sie im [Amazon S3 S3-Benutzerhandbuch](#).

In den folgenden Themen wird beschrieben, wie Sie mit Amazon S3 S3-Objekten und -Ressourcen aus dem arbeitenAWS Toolkit for Visual Studio Code.

Themen

- [Arbeiten mit Amazon S3 S3-Ressourcen](#)
- [Arbeiten mit Amazon S3-Objekten](#)

Arbeiten mit Amazon S3 S3-Ressourcen

Sie können Amazon S3 von verwenden, AWS Toolkit for Visual Studio Code um Ihre Amazon S3 S3-Buckets und andere Ressourcen anzusehen, zu verwalten und zu bearbeiten.

In den folgenden Abschnitten wird beschrieben, wie Sie mit Amazon S3 S3-Ressourcen aus dem arbeiten. AWS Toolkit for Visual Studio Code Informationen zum Arbeiten mit Amazon S3 S3-Objekten, wie Ordnern und DateienAWS Toolkit for Visual Studio Code, finden Sie im Thema [Arbeiten mit S3-Objekten](#) in diesem Benutzerhandbuch.

Erstellung eines Amazon S3-Buckets

1. Öffnen Sie im Toolkit-Explorer das Kontextmenü (Rechtsklick) für den S3-Service und wählen Sie Create Bucket... . Wählen Sie alternativ das Symbol „Bucket erstellen“, um das Dialogfeld „Bucket erstellen“ zu öffnen.
2. Geben Sie im Feld Name the bucket (Benennen Sie den Bucket) einen Namen für den Bucket ein.

Drücken Sie die Eingabetaste, um den Bucket zu erstellen und das Dialogfeld zu schließen. Ihr neuer Bucket wird dann unter dem S3-Service im Toolkit angezeigt.

Note

Da Amazon S3 die Verwendung Ihres Buckets als URL ermöglicht, auf die öffentlich zugegriffen werden kann, muss der von Ihnen gewählte Bucket-Name weltweit eindeutig sein. Wenn ein anderes Konto bereits einen Bucket mit dem Namen erstellt hat, den Sie verwenden möchten, müssen Sie einen anderen Namen verwenden.

Wenn Sie keinen neuen Bucket erstellen können, überprüfen Sie die AWSToolkit-Protokolle auf der Registerkarte Ausgabe. Wenn Sie versuchen, einen ungültigen Bucket-Namen zu verwenden, tritt ein `BucketAlreadyExists` Fehler auf. Weitere Informationen finden Sie unter [Bucket-Einschränkungen und -Limits](#) im Benutzerhandbuch zu Amazon Simple Storage Service.

Hinzufügen eines Ordners zu einem Amazon S3 Bucket

Sie können den Inhalt eines S3-Buckets organisieren, indem Sie Ihre Objekte in Ordnern gruppieren. Sie können auch Ordner innerhalb von Ordnern erstellen.

1. Erweitern Sie im Toolkit-Explorer den S3-Dienst, um eine Liste Ihrer S3-Ressourcen anzuzeigen.
2. Wählen Sie das Symbol „Ordner erstellen“, um das Dialogfeld „Ordner erstellen“ zu öffnen. Oder öffnen Sie das Kontextmenü (Rechtsklick) für einen Bucket oder Ordner und wählen Sie dann Ordner erstellen.
3. Geben Sie einen Wert in das Feld Ordnername ein und drücken Sie die Eingabetaste, um den Ordner zu erstellen und das Dialogfeld zu schließen. Ihr neuer Ordner wird unter der entsprechenden S3-Ressource im Toolkit-Menü angezeigt.

Verwenden eines Amazon-S3-Buckets

Wenn Sie einen S3-Bucket löschen, löschen Sie auch die darin enthaltenen Ordner und Objekte. Wenn Sie also versuchen, einen Bucket zu löschen, werden Sie aufgefordert, zu bestätigen, dass Sie ihn löschen möchten.

1. Erweitern Sie im Toolkit-Hauptmenü den Amazon S3 S3-Service, um eine Liste Ihrer S3-Ressourcen anzuzeigen.
2. Öffnen Sie das Kontextmenü (Rechtsklick) für einen Bucket oder Ordner und wählen Sie dann S3-Bucket löschen.
3. Wenn Sie dazu aufgefordert werden, geben Sie den Namen des Buckets in das Textfeld ein und drücken Sie dann die Eingabetaste, um den Bucket zu löschen und die Bestätigungsaufforderung zu schließen.

Note

Wenn Ihr Bucket Objekte enthält, wird er geleert, bevor er gelöscht wird. Wenn Sie versuchen, eine große Anzahl von Ressourcen oder Objekten gleichzeitig zu löschen, kann es einige Zeit dauern, bis sie gelöscht sind. Nachdem sie gelöscht wurden, erhalten Sie eine Benachrichtigung, dass sie erfolgreich gelöscht wurden.

Arbeiten mit Amazon S3-Objekten

Ihre Dateien, Ordner und alle anderen Daten, die in einem S3-Ressourcen-Bucket gespeichert sind, werden als S3-Objekte bezeichnet.

In den folgenden Abschnitten wird beschrieben, wie Sie mit Amazon S3 S3-Objekten aus dem arbeitenAWS Toolkit for Visual Studio Code. Einzelheiten zur Arbeit mit Amazon S3 S3-Ressourcen, wie z. B. S3-BucketsAWS Toolkit for Visual Studio Code, finden Sie im Thema [Arbeiten mit S3-Ressourcen](#) in diesem Benutzerhandbuch.

Paginierung von Objekten

Wenn Sie mit einer großen Anzahl von Amazon S3 S3-Objekten und -Ordnern arbeiten, können Sie mit der Paginierung die Anzahl der Elemente angeben, die Sie auf einer Seite anzeigen möchten.

1. Navigieren Sie zur VS Code-Aktivitätsleiste und wählen Sie Erweiterungen.
2. Wählen Sie in der AWS Toolkit-Erweiterung das Einstellungssymbol und dann Erweiterungseinstellungen aus.
3. Scrollen Sie auf der Einstellungsseite nach unten zur Einstellung AWS> S3: Max. Elemente pro Seite.
4. Ändern Sie den Standardwert auf die Anzahl der S3-Elemente, die angezeigt werden sollen, bevor „Weitere laden“ angezeigt wird.

Note

Gültige Werte umfassen eine beliebige Zahl zwischen 3 und 1000. Diese Einstellung gilt nur für die Anzahl der gleichzeitig angezeigten Objekte oder Ordner. Alle Buckets, die Sie erstellt haben, werden gleichzeitig angezeigt. In der Standardeinstellung können Sie bis zu 100 Buckets in jedem Ihrer AWS-Konten erstellen.

5. Schließen Sie die Seite mit den Einstellungen, um Ihre Änderungen zu bestätigen.

Sie können die Einstellungen in einer Datei im JSON-Format auch aktualisieren, indem Sie oben rechts auf der Einstellungsseite auf das Symbol „Einstellungen öffnen“ (JSON) klicken.

Amazon S3 S3-Objekte hoch- und herunterladen

Sie können lokal gespeicherte Dateien in Ihre Amazon S3 S3-Buckets hochladen oder Amazon S3 S3-Remote-Objekte auf Ihr lokales System herunterladen, und zwar von. AWS Toolkit for Visual Studio Code

Laden Sie eine Datei mit dem Toolkit hoch

1. Erweitern Sie im Toolkit-Explorer den Amazon S3 S3-Service, um eine Liste Ihrer S3-Ressourcen anzuzeigen.
2. Wählen Sie das Symbol „Datei hochladen“ neben einem Bucket oder Ordner, um das Dialogfeld „Datei hochladen“ zu öffnen. Sie können auch das Kontextmenü (Rechtsklick) öffnen und Datei hochladen wählen.

Note

Um eine Datei in den übergeordneten Ordner oder die übergeordnete Ressource des Objekts hochzuladen, öffnen Sie das Kontextmenü (Rechtsklick) für ein beliebiges S3-Objekt und wählen Sie In übergeordnetes Objekt hochladen.

3. Verwenden Sie den Dateimanager Ihres Systems, um eine Datei auszuwählen, und wählen Sie dann Datei hochladen, um den Dialog zu schließen und die Datei hochzuladen.

Laden Sie eine Datei mithilfe der Befehlspalette hoch

Sie können die Toolkit-Oberfläche oder die Befehlspalette verwenden, um eine Datei in einen Bucket hochzuladen.

1. Um eine Datei für den Upload auszuwählen, wählen Sie die Registerkarte dieser Datei in VS Code.
2. Drücken Sie Strg+Shift+P, um die Befehlspalette anzuzeigen.
3. Geben Sie in der Befehlspalette den Ausdruck ein, `upload file` um eine Liste der empfohlenen Befehle anzuzeigen.

4. Wählen Sie den Befehl AWS: Datei hochladen, um den Dialog AWS: Datei hochladen zu öffnen.
5. Wenn Sie dazu aufgefordert werden, wählen Sie die Datei aus, die Sie hochladen möchten, und wählen Sie dann den Bucket aus, in den Sie die Datei hochladen möchten.
6. Bestätigen Sie Ihren Upload, um den Dialog zu schließen und den Upload-Vorgang zu starten. Wenn der Upload abgeschlossen ist, wird das Objekt im Toolkit-Menü mit Metadaten angezeigt, die die Objektgröße, das Datum der letzten Änderung und den Pfad enthalten.

Herunterladen eines Amazon S3 Objekts

1. Erweitern Sie im Toolkit-Explorer den S3-Dienst.
2. Öffnen Sie in einem Bucket oder Ordner das Kontextmenü (Rechtsklick) für ein Objekt, das Sie herunterladen möchten. Wählen Sie dann „Herunterladen als“, um das Dialogfeld „Herunterladen als“ zu öffnen. Oder wählen Sie alternativ das Symbol „Herunterladen als“ neben dem Objekt.
3. Wählen Sie mit dem Dateimanager Ihres Systems einen Zielordner aus, geben Sie einen Dateinamen ein und wählen Sie dann Herunterladen, um das Dialogfeld zu schließen und den Download zu starten.

Entfernte Objekte bearbeiten

Sie können den verwenden AWS Toolkit for Visual Studio Code, um Ihre Amazon S3 S3-Objekte zu bearbeiten, die in Ihren Amazon S3 S3-Remote-Ressourcen gespeichert sind.

1. Erweitern Sie im Toolkit-Explorer den S3-Service.
2. Erweitern Sie die S3-Ressource, die die Datei enthält, die Sie bearbeiten möchten.
3. Um die Datei zu bearbeiten, wählen Sie das Stiftsymbol (Datei bearbeiten).
4. Um eine Datei zu bearbeiten, die im schreibgeschützten Modus geöffnet ist, zeigen Sie die Datei im VS Code-Editor an und wählen Sie dann das Stiftsymbol in der oberen rechten Ecke der Benutzeroberfläche aus.

Note

- Wenn Sie VS Code neu starten oder beenden, trennt Ihre IDE die Verbindung zu Ihren S3-Ressourcen. Wenn beim Trennen der Verbindung Remote-S3-Dateien bearbeitet werden, wird die Bearbeitung beendet. Sie müssen VS Code neu starten und die Registerkarte Bearbeiten erneut öffnen, um die Bearbeitung fortzusetzen.

- Die Schaltfläche „Datei bearbeiten“ befindet sich in der oberen rechten Ecke der Benutzeroberfläche. Sie ist nur sichtbar, wenn Sie aktiv eine schreibgeschützte Datei im VS Code-Editor anzeigen.
- Nicht-Textdateien können nicht im schreibgeschützten Modus geöffnet werden. Sie werden immer im Bearbeitungsmodus geöffnet.
- Sie können nicht vom Nur-Edit-Modus in den Nur-Lese-Modus zurückkehren, sondern nur umgekehrt.

Den Pfad eines Amazon S3 S3-Objekts kopieren

Das folgende Verfahren beschreibt, wie Sie den Pfad eines Amazon S3 S3-Objekts aus dem kopierenAWS Toolkit for Visual Studio Code.

1. Erweitern Sie im Toolkit-Explorer den S3-Service.
2. Erweitern Sie den Ressourcen-Bucket, der das Objekt enthält, für das Sie den Pfad kopieren möchten.
3. Öffnen Sie das Kontextmenü (Rechtsklick) für das Objekt, für das Sie den Pfad kopieren möchten, und wählen Sie dann Pfad kopieren, um den Objektpfad in Ihre lokale Zwischenablage zu kopieren.

Generieren Sie eine vorsignierte URL für ein Amazon-S3-Objekt

Sie können private Amazon S3 S3-Objekte mit anderen teilen, indem Sie über die vorsignierte URL-Funktion zeitlich begrenzte Berechtigungen für Downloads gewähren. Weitere Informationen finden Sie unter [Objekte mit einer vorsignierten URL teilen](#).

1. Erweitern Sie im Toolkit-Explorer den S3-Dienst.
2. Öffnen Sie in einem Bucket oder Ordner das Kontextmenü (Rechtsklick) für ein Objekt, das Sie teilen möchten. Wählen Sie dann „Vorsignierte URL generieren“, um die Befehlspalette zu öffnen.
3. Geben Sie in der Befehlspalette die Anzahl der Minuten ein, für die die URL für den Zugriff auf Ihr Objekt verwendet werden kann. Wählen Sie dann die Eingabetaste, um den Dialog zu bestätigen und zu schließen.
4. Nachdem die vorsignierte URL generiert wurde, zeigt die VS Code-Statusleiste die vorsignierte URL für das Objekt an, das in Ihre lokale Zwischenablage kopiert wurde.

Löschen von Amazon-S3-Objekten

Wenn sich ein Objekt in einem Bucket ohne Versionsnummer befindet, können Sie es dauerhaft löschen. Bei Buckets, für die Versionsverwaltung aktiviert ist, wird das Objekt durch eine Löschanforderung nicht dauerhaft gelöscht. Stattdessen fügt Amazon S3 eine Löschmarkierung im Bucket ein. Weitere Informationen finden Sie unter [Objektversionen löschen](#).

1. Erweitern Sie im Toolkit-Explorer den S3-Dienst, um eine Liste Ihrer S3-Ressourcen anzuzeigen.
2. Öffnen Sie das Kontextmenü (Rechtsklick) für ein Objekt, das Sie löschen möchten, und wählen Sie dann Löschen, um das Bestätigungsdialogfeld zu öffnen.
3. Wählen Sie Löschen, um zu bestätigen, dass Sie das S3-Objekt löschen möchten. Schließen Sie dann den Dialog.

Arbeiten mit serverlosen Anwendungen

Das AWS Toolkit for Visual Studio Code bietet Unterstützung für [AWS-Serverless-Anwendung](#). In den folgenden Themen wird beschrieben, wie Sie mit dem Erstellen und Arbeiten mit AWS Serverless Application Model (AWS SAM) -Anwendungen beginnen, und zwar von der AWS Toolkit for Visual Studio Code.

Themen

- [Erste Schritte mit serverlosen Anwendungen](#)
- [Lambda-Funktionen direkt aus dem Code ausführen und debuggen](#)
- [Ausführen und Debuggen lokaler Amazon API Gateway Ressourcen](#)
- [Konfigurationsoptionen für das Debuggen von serverless Anwendungen](#)
- [Fehlerbehebung bei serverlosen Anwendungen](#)

Erste Schritte mit serverlosen Anwendungen

In den folgenden Abschnitten wird beschrieben, wie Sie mit der Erstellung von Stacks AWS-Serverless-Anwendung aus den AWS Toolkit for Visual Studio Code, using AWS Serverless Application Model (AWS SAM) und AWS CloudFormation Stacks beginnen.

Voraussetzungen

Bevor Sie eine erstellen oder damit arbeiten können AWS-Serverless-Anwendung, müssen die folgenden Voraussetzungen erfüllt sein.

Note

Bei den folgenden Vorgängen müssen Sie VS Code möglicherweise beenden oder neu starten, bevor die Änderungen abgeschlossen sind.

- Installieren Sie die AWS SAM Befehlszeilenschnittstelle (CLI). Weitere Informationen und Anweisungen zur Installation der AWS SAM CLI finden Sie im Thema [Installation der AWS SAM CLI](#) in diesem AWS Serverless Application Model Benutzerhandbuch.
- Identifizieren Sie in Ihrer AWS Konfigurationsdatei Ihre AWS Standardregion. Weitere Informationen zu Ihrer Konfigurationsdatei finden Sie unter dem Thema [Einstellungen für die Konfiguration und die Anmeldeinformationsdatei](#) im AWS Command Line Interface Benutzerhandbuch.
- Installieren Sie Ihr Sprach-SDK und konfigurieren Sie Ihre Toolchain. Weitere Informationen zur Konfiguration Ihrer Toolchain AWS Toolkit for Visual Studio Code finden Sie im Thema [Konfiguration Ihrer Toolchain](#) in diesem Benutzerhandbuch.
- Installieren Sie die [YAML-Sprachunterstützungserweiterung](#) vom VS Code Marketplace. Dies ist erforderlich, damit auf CodeLens AWS SAM Vorlagendateien zugegriffen werden kann. Weitere Informationen zu CodeLens finden Sie im [CodeLens](#) Abschnitt der VS Code-Dokumentation

IAM-Berechtigungen für serverlose Anwendungen

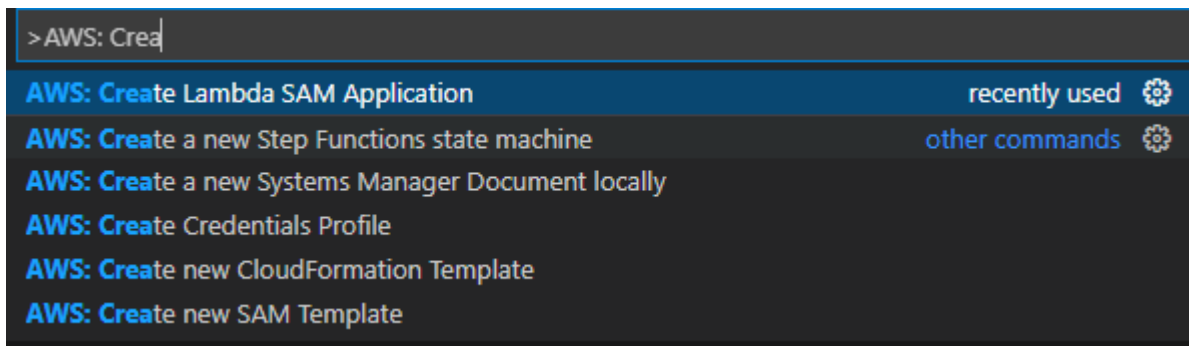
Im Toolkit for VS Code benötigen Sie ein Anmeldeinformationsprofil, das die AWS Identity and Access Management (IAM-) Berechtigungen enthält, die für die Bereitstellung und Ausführung serverloser Anwendungen erforderlich sind. Sie benötigen entsprechenden Lese-/Schreibzugriff auf die folgenden Dienste: AWS CloudFormation, IAM, Lambda, Amazon API Gateway, Amazon Simple Storage Service (Amazon S3) und Amazon Elastic Container Registry (Amazon ECR).

Weitere Informationen zur Einrichtung der Authentifizierung, die für die Bereitstellung und Ausführung serverloser Anwendungen erforderlich ist, finden Sie im Developer Guide unter [Managing Resource Access and Permissions](#). AWS Serverless Application Model Informationen zum Einrichten Ihrer Anmeldeinformationen finden Sie [AWS IAM-Anmeldeinformationen](#) in diesem Benutzerhandbuch.

Eine neue serverlose Anwendung (lokal) erstellen

Dieses Verfahren zeigt, wie Sie eine serverlose Anwendung mit dem Toolkit for VS Code erstellen, indem Sie AWS SAM. Die Ausgabe dieses Verfahrens ist ein lokales Verzeichnis auf Ihrem Entwicklungshost, das eine serverlose Beispielanwendung enthält, die Sie erstellen, lokal testen, ändern und in der Cloud bereitstellen können. AWS

1. Um die Befehlspalette zu öffnen, wählen Sie Ansicht, Befehlspalette und geben Sie dann die Eingabetaste ein AWS.
2. Wählen Sie AWSToolkit Create Lambda SAM Application aus.



Note

Wenn die AWS SAM CLI nicht installiert ist, wird in der unteren rechten Ecke des VS Code-Editors eine Fehlermeldung angezeigt. Stellen Sie in diesem Fall sicher, dass Sie alle [Annahmen und Voraussetzungen](#) erfüllt haben.

3. Wählen Sie die Laufzeit für Ihre AWS SAM Anwendung.

Note

Wenn Sie eine der Laufzeiten mit „(Image)“ auswählen, ist Ihre Anwendung Pakettyp Image. Wenn Sie eine der Laufzeiten ohne „(Image)“ auswählen, ist Ihre Anwendung der Typ Zip. Für weitere Informationen zu den Unterschieden zwischen Image und Zip Pakettypen finden Sie unter [Lambda-Bereitstellungspakete](#) im AWS Lambda Entwicklerhandbuch.

4. Abhängig von der ausgewählten Laufzeit werden Sie möglicherweise aufgefordert, einen Abhängigkeitsmanager und eine Laufzeitarchitektur für Ihre SAM-Anwendung auszuwählen.

Dependency Manager

Wählen Sie zwischen Gradle oder Maven.

Note

Diese Auswahl an Tools zur Build-Automatisierung ist nur für Java-Laufzeiten verfügbar.

Architecture

Wählen Sie zwischen x86_64 oder arm64.

Die Option, Ihre serverlose Anwendung in einer ARM64-basierten emulierten Umgebung anstelle der standardmäßigen x86_64-basierten Umgebung auszuführen, ist für die folgenden Laufzeiten verfügbar:

- nodejs12.x (ZIP und Bild)
- nodejs14.x (ZIP und Bild)
- python3.8 (ZIP und Bild)
- python3.9 (ZIP und Bild)
- python3.10 (ZIP und Bild)
- python3.11 (ZIP und Bild)
- python3.12 (ZIP und Bild)
- java8.al2 mit Gradle (ZIP und Bild)
- java8.al2 mit Maven (nur ZIP)
- java11 mit Gradle (ZIP und Bild)
- java11 mit Maven (nur ZIP)

⚠ Important

Sie müssen AWS CLI Version 1.33.0 oder höher installieren, damit Anwendungen in ARM64-basierten Umgebungen ausgeführt werden können. Weitere Informationen finden Sie unter [Voraussetzungen](#).

5. Wählen Sie einen Speicherort für Ihr neues Projekt aus. Sie können einen vorhandenen Arbeitsbereichordner verwenden, wenn einer geöffnet ist, mit Select a different folder (Einen anderen Ordner wählen) einen bereits vorhandenen Ordner auswählen oder einen neuen Ordner erstellen und auswählen. Wählen Sie in diesem Beispiel die Option There are no workspace folders open (Es sind keine Arbeitsbereichsordner geöffnet), um einen Ordner mit dem Namen MY-SAM-APP zu erstellen.
6. Geben Sie einen Namen für Ihr neues Projekt ein. Wählen Sie für dieses Beispiel my-sam-app-node.js. Nachdem Sie die Eingabetaste gedrückt haben, benötigt das Toolkit for VS Code einen Moment, um das Projekt zu erstellen.

Wenn das Projekt erstellt wird, wird Ihre Anwendung Ihrem aktuellen Arbeitsbereich hinzugefügt. Sie sollten es im Explorer-Fenster sehen.

Öffnen einer serverlosen Anwendung (lokal)

Um eine serverlose Anwendung auf Ihrem lokalen Entwicklungshost zu öffnen, öffnen Sie den Ordner, der die Vorlagendatei der Anwendung enthält.

1. Wählen Sie in der Datei die Option Ordner öffnen... .
2. Navigieren Sie im Dialogfeld „Ordner öffnen“ zu dem serverlosen Anwendungsordner, den Sie öffnen möchten.
3. Wählen Sie die Schaltfläche „Ordner auswählen“.

Wenn Sie den Ordner einer Anwendung öffnen, wird er dem Explorer-Fenster hinzugefügt.

Eine serverlose Anwendung über eine Vorlage ausführen und debuggen (lokal)

Sie können das Toolkit for VS Code verwenden, um zu konfigurieren, wie serverlose Anwendungen debuggt und lokal in Ihrer Entwicklungsumgebung ausgeführt werden.

Sie beginnen mit der Konfiguration des Debug-Verhaltens, indem Sie die VS [CodeLens](#) Code-Funktion verwenden, um eine geeignete Lambda-Funktion zu identifizieren. CodeLens ermöglicht inhaltsorientierte Interaktionen mit Ihrem Quellcode. Informationen dazu, wie Sie sicherstellen können, dass Sie auf die CodeLens Funktion zugreifen können, finden Sie im [Voraussetzungen](#) Abschnitt weiter oben in diesem Thema.

Note

In diesem Beispiel debuggen Sie eine Anwendung, die verwendet JavaScript. Sie können jedoch die Debugging-Funktionen von Toolkit for VS Code mit den folgenden Sprachen und Laufzeiten verwenden:

- C# — .NET Core 2.1, 3.1; .NET 5.0
- JavaScript/TypeScript — Node.js 12. x, 14. x
- Python — 3.6, 3.7, 3.8, 3.9, 3.10, 3.11, 3.12
- Java — 8, 8.al2, 11
- Gehe — 1.x

Ihre Sprachauswahl wirkt sich auch darauf aus, wie geeignete Lambda-Handler CodeLens erkannt werden. Weitere Informationen finden Sie unter [Lambda-Funktionen direkt aus dem Code ausführen und debuggen](#).

In diesem Verfahren verwenden Sie die Beispielanwendung, die im [Eine neue serverlose Anwendung \(lokal\) erstellen](#) Abschnitt weiter oben in diesem Thema erstellt wurde.

1. Um Ihre Anwendungsdateien im Datei-Explorer von VS Code anzuzeigen, wählen Sie Ansicht, Explorer.
2. Öffnen Sie die `template.yaml` Datei im Anwendungsordner (z. B. `my-sample-app`).

Note

Wenn Sie eine Vorlage mit einem anderen Namen als `template.yaml`, ist der CodeLens Indikator nicht automatisch in der YAML-Datei verfügbar. Das bedeutet, dass Sie manuell eine Debug-Konfiguration hinzufügen müssen.

3. Gehen Sie im Editor für `template.yaml` zu dem Resources Abschnitt der Vorlage, der serverlose Ressourcen definiert. In diesem Fall ist dies die `HelloWorldFunction` Ressource vom Typ `AWS::Serverless::Function`.

Wählen Sie im CodeLens Indikator für diese Ressource die Option Debug-Konfiguration hinzufügen aus.

4. Wählen Sie in der Befehlspalette die Laufzeit aus, in der Ihre AWS SAM Anwendung ausgeführt werden soll.
5. Klicken Sie im Editor für die Datei `launch.json`, bearbeiten oder bestätigen Sie Werte für die folgenden Konfigurationseigenschaften:
 - `"name"`— Geben Sie einen leserfreundlichen Namen ein, der in der Konfiguration im Dropdownfeld Ausführen angezeigt wird.
 - `"target"`— Stellen Sie sicher, dass der Wert `"template"` so ist, dass die AWS SAM Vorlage der Einstiegspunkt für die Debug-Sitzung ist.
 - `"templatePath"`— Geben Sie einen relativen oder absoluten Pfad für die Datei `template.yaml` an.
 - `"logicalId"`— Stellen Sie sicher, dass der Name mit dem Namen übereinstimmt, der im Abschnitt Ressourcen der AWS SAM Vorlage angegeben ist. In diesem Fall handelt es sich um den `HelloWorldFunction` vom Typ `AWS::Serverless::Function`.

Weitere Informationen zu diesen und anderen Einträgen in `launch.json` finden Sie in [Konfigurationsoptionen für das Debuggen von serverless Anwendungen](#).

6. Wenn Sie mit Ihrer Debug-Konfiguration zufrieden sind, speichern Sie `launch.json`. Um dann mit dem Debuggen zu beginnen, wählen Sie in der RUN-Ansicht die grüne Schaltfläche „Abspielen“.

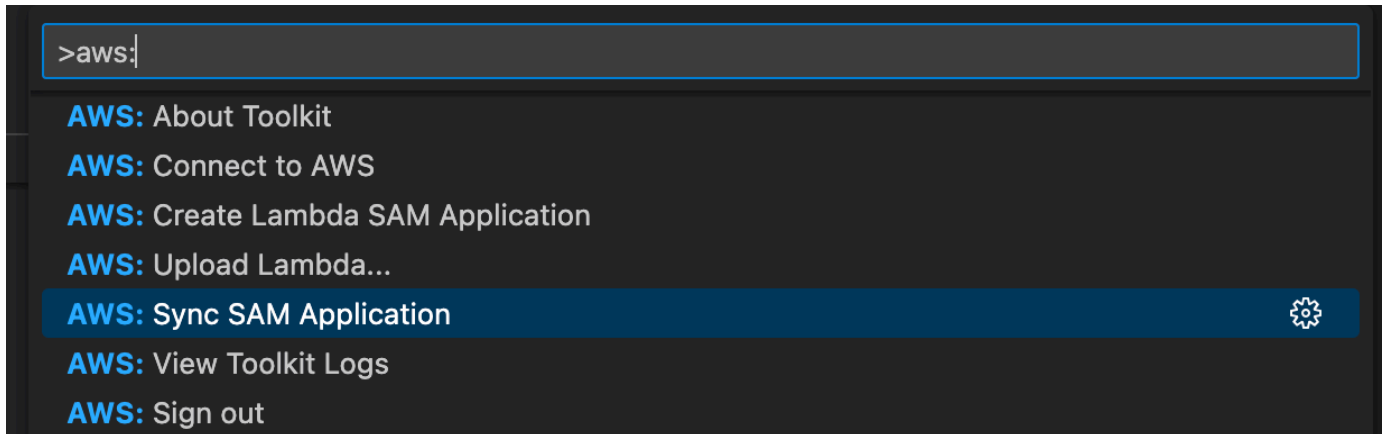
Wenn die Debugging-Sitzungen gestartet werden, wird die DEBUGKONSOLE zeigt die Debugging-Ausgabe an und zeigt alle von der Lambda Funktion zurückgegebenen Werte an. (Beim Debuggen von AWS SAM Anwendungen wird das AWSToolkit im Bedienfeld „Ausgabe“ als Ausgabekanal ausgewählt.)

Anwendungen werden synchronisiert AWS SAM

Das AWS Toolkit for Visual Studio Code führt den AWS SAM CLI-Befehl `aussam sync`, um Ihre serverlosen Anwendungen auf dem AWS Cloud bereitzustellen. Weitere Informationen zur AWS SAM Synchronisierung finden Sie im [Referenzthema AWS SAM CLI Command](#) im AWS Serverless Application Model Developer Guide

Das folgende Verfahren beschreibt, wie Sie Ihre serverlosen Anwendungen AWS Cloud mithilfe `sam sync` des Toolkit for VS Code bereitstellen.

1. Öffnen Sie im Hauptmenü von VS Code die Befehlspalette, indem Sie Ansicht erweitern und Befehlspalette auswählen.
2. Suchen Sie in der Befehlspalette nach SAM-Anwendung synchronisieren AWS und wählen Sie diese aus, um mit der Einrichtung Ihrer Synchronisierung zu beginnen.




3. Wählen Sie die AWS Region aus, mit der Ihre serverlose Anwendung synchronisiert werden soll.
4. Wählen Sie die `template.yaml`-Datei aus, die für die Bereitstellung verwendet werden soll.
5. Wählen Sie einen vorhandenen Amazon S3 S3-Bucket aus oder geben Sie einen neuen Amazon S3 S3-Bucket-Namen ein, in dem Sie Ihre Anwendung bereitstellen möchten.

Important

Ihr Amazon S3 S3-Bucket muss die folgenden Anforderungen erfüllen:

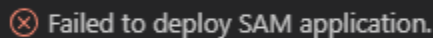
- Der Bucket muss sich in der Region befinden, mit der Sie synchronisieren.
- Der Amazon-S3-Bucket-Name muss unter den in Amazon S3 vorhandenen Bucket-Namen global eindeutig sein.


6. Wenn Ihre serverless Anwendung eine Funktion mit dem Pakettyp Image. Geben Sie den Namen eines Amazon ECR-Repositorys ein, das diese Bereitstellung verwenden kann. Der Bucket muss sich in der Region befinden, in der Sie bereitstellen.
7. Wählen Sie einen Deployment-Stack aus der Liste Ihrer vorherigen Deployments aus oder erstellen Sie einen neuen Deployment-Stack, indem Sie einen neuen Stack-Namen eingeben. Fahren Sie dann fort, um den Synchronisierungsvorgang zu starten.


 Note

Stacks, die in früheren Bereitstellungen verwendet wurden, werden pro Workspace und Region zurückgerufen.

8. Während des Synchronisierungsvorgangs wird der Status Ihrer Bereitstellung auf der Registerkarte Terminal von VS Code erfasst. Stellen Sie auf der Registerkarte Terminal sicher, dass Ihre Synchronisierung erfolgreich war. Wenn ein Fehler auftritt, erhalten Sie eine Benachrichtigung.



 Failed to deploy SAM application.

 Note

Weitere Informationen zu Ihrer Synchronisierung finden Sie in den AWS Toolkit for Visual Studio Code Protokollen über die Befehlspalette.

Um über die Befehlspalette auf Ihre AWS Toolkit for Visual Studio Code Protokolle zuzugreifen, erweitern Sie Ansicht, wählen Sie Befehlspalette **AWS: View AWS Toolkits Logs**, suchen Sie danach und wählen Sie sie aus, wenn sie in der Liste angezeigt wird.

Wenn die Bereitstellung abgeschlossen ist, wird Ihre Anwendung im AWSExplorer aufgeführt. Weitere Informationen zum Aufrufen der Lambda-Funktion, die als Teil der Anwendung erstellt wurde, finden Sie im [Interaktion mit Remote-Lambda-Funktionen](#) Thema in diesem Benutzerhandbuch.

Löschen einer serverlosen Anwendung aus dem AWS Cloud

Beim Löschen einer serverless Anwendung wird der AWS CloudFormation-Stack gelöscht, den Sie zuvor für die AWS Cloud bereitgestellt haben. Beachten Sie, dass dieses Verfahren Ihr Anwendungsverzeichnis nicht von Ihrem lokalen Host löscht.

1. Öffnen Sie die [AWS Entdecker](#).
2. Erweitern Sie im AWSToolkit-Explorer-Fenster die Region, die die bereitgestellte Anwendung enthält, die Sie löschen möchten, und erweitern Sie dann AWS CloudFormation
3. Öffnen Sie das Kontextmenü (Rechtsklick) für den Namen des AWS CloudFormation Stacks, der der serverlosen Anwendung entspricht, die Sie löschen möchten, und wählen Sie dann Stack löschen AWS CloudFormation aus.
4. Um zu bestätigen, dass Sie den ausgewählten Stack löschen möchten, wählen Sie Ja.

Wenn das Löschen des Stacks erfolgreich ist, entfernt das Toolkit for VS Code den Stacknamen aus der AWS CloudFormation Liste im AWS Explorer.

Lambda-Funktionen direkt aus dem Code ausführen und debuggen

Beim Testen der AWS SAM Anwendung können Sie wählen, ob Sie nur die Lambda-Funktion ausführen und debuggen und andere Ressourcen ausschließen möchten, die in der AWS SAM Vorlage definiert sind. Bei diesem Ansatz wird die [CodeLens](#)Funktion verwendet, um Lambda-Funktionshandler im Quellcode zu identifizieren, die Sie direkt aufrufen können.

Die Lambda-Handler, die von erkannt werden, CodeLens hängen von der Sprache und Laufzeit ab, die Sie für Ihre Anwendung verwenden.

Sprache/Laufzeit	Kriterien für die Identifizierung von Lambda-Funktionen anhand von Indikatoren CodeLens
C# (dotnetcore2.1, 3.1; .NET 5.0)	<p>Die Suchfunktion bietet zudem die folgenden Funktionen:</p> <ul style="list-style-type: none"> • Es ist eine öffentliche Funktion einer öffentlichen Klasse. • Sie hat einen oder zwei Parameter. Bei zwei Parametern muss der zweite Parameter die

Sprache/Laufzeit	<p>Kriterien für die Identifizierung von Lambda-Funktionen anhand von Indikatoren CodeLens</p> <p>ILambdaContext Schnittstelle implementieren.</p> <ul style="list-style-type: none">• Es hat eine *.csproj Datei in seinem übergeordneten Ordner innerhalb des VS Code-Workspace-Ordners. <p>Die Erweiterung ms-dotnettools.csharp (oder jede Erweiterung, die Sprachsymbole für C# bereitstellt) ist installiert und aktiviert.</p>
JavaScript/(Node.js 12.x, 14.x) TypeScript	<p>Die Suchfunktion bietet zudem die folgenden Funktionen:</p> <ul style="list-style-type: none">• Es ist eine exportierte Funktion mit bis zu drei Parametern.• Es hat eine package.json Datei in seinem übergeordneten Ordner innerhalb des VS Code-Workspace-Ordners.
Python (3,7, 3,8, 3,9, 3,10, 3,11, 3,12)	<p>Die Suchfunktion bietet zudem die folgenden Funktionen:</p> <ul style="list-style-type: none">• Es ist eine Top-Level-Funktion.• Es hat eine requirements.txt Datei in seinem übergeordneten Ordner innerhalb des VS Code-Workspace-Ordners. <p>Die Erweiterung ms-python.python (oder jede Erweiterung, die Sprachsymbole für Python bereitstellt) ist installiert und aktiviert.</p>

Sprache/Laufzeit	Kriterien für die Identifizierung von Lambda-Funktionen anhand von Indikatoren CodeLens
Java (8, 8.al2, 11)	<p>Die Suchfunktion bietet zudem die folgenden Funktionen:</p> <ul style="list-style-type: none">• Es ist eine öffentliche Funktion einer öffentlichen, nicht abstrakten Klasse.• Sie hat einen, zwei oder drei Parameter:<ul style="list-style-type: none">• Ein Parameter: Der Parameter kann alles sein.• Zwei Parameter: Parameter müssen ein <code>java.io.InputStream</code> und ein <code>java.io.OutputStream</code> ODER, der letzte Parameter muss ein <code>com.amazonaws.services.lambda.runtime.Context</code> sein.• Drei Parameter: Parameter müssen ein <code>java.io.InputStream</code> ein <code>java.io.OutputStream</code> und der letzte Parameter muss ein <code>com.amazonaws.services.lambda.runtime.Context</code> sein.• Es hat eine <code>build.gradle</code> (Gradle) oder <code>pom.xml</code> (Maven) -Datei in seinem übergeordneten Ordner innerhalb des VS Code-Workspace-Ordners. <p>Die Erweiterung redhat.java (oder jede Erweiterung, die Sprachsymbole für Java bereitstellt) ist installiert und aktiviert. Diese Erweiterung benötigt Java 11, unabhängig davon, welche Java-Runtime Sie verwenden.</p> <p>Das vscjava.vscodex-ide-debugDie Erweiterung (oder eine beliebige Erweiterung, die einen</p>

Sprache/Laufzeit	Kriterien für die Identifizierung von Lambda-Funktionen anhand von Indikatoren CodeLens
	Java-Debugger bereitstellt) ist installiert und aktiviert.
Gehe zu (1.x)	<p>Die Suchfunktion bietet zudem die folgenden Funktionen:</p> <ul style="list-style-type: none"> • Es ist eine Top-Level-Funktion. • Es benötigt zwischen 0 und 2 Argumente. Bei zwei Argumenten muss das erste Argument <code>implementiere context.Context</code>. • Es gibt zwischen 0 und 2 Argumente zurück. Wenn es mehr als 0 Argumente gibt, muss das letzte Argument implementiert <code>werdenerror</code>. • Es hat eine <code>go.mod</code> Datei im VS Code Workspace-Ordner. <p>Die Erweiterung <code>golang.go</code> ist installiert, konfiguriert und aktiviert.</p>

So führen und debuggen Sie eine serverless Anwendung direkt aus dem Anwendungscode

1. Um Ihre Anwendungsdateien im VS Code File Explorer anzuzeigen, wählen Sie Ansicht, Explorer.
2. Erweitern Sie im Anwendungsordner (z. B. `my-sample-app`) den Funktionsordner (in diesem Fall `hello-world`) und öffnen Sie die `app.js` Datei.
3. Wählen Sie in dem CodeLens Indikator, der einen geeigneten Lambda-Funktionshandler identifiziert, `add Debug Configuration`.
4. Wählen Sie in der Befehlspalette die Laufzeit aus, in der Ihre AWS SAM Anwendung ausgeführt werden soll.
5. Klicken Sie im Editor für die Datei `launch.json`, bearbeiten oder bestätigen Sie Werte für die folgenden Konfigurationseigenschaften:

- "name" – Geben Sie einen leserfreundlichen Namen ein, der in der im Dropdown-Feld Configuration (Konfiguration) der Ansicht Run (Ausführen) angezeigt wird.
- "target"— Stellen Sie sicher, dass der Wert "code" ist, sodass ein Lambda Funktionshandler direkt aufgerufen wird.
- "lambdaHandler" – Geben Sie den Namen der Methode innerhalb Ihres Codes ein, die die Lambda-Funktion zum Ausführen Ihrer Funktion aufruft. Für Anwendungen in JavaScript ist die Standardeinstellung beispielsweise `app.lambdaHandler`.
- "projectRoot"— Geben Sie den Pfad zur Anwendungsdatei ein, die die Lambda -Funktion enthält.
- "runtime" – Geben Sie eine gültige Laufzeit für die Lambda Ausführungsumgebung ein oder bestätigen Sie diese, z. B. "nodejs.12x".
- "payload"— Wählen Sie eine der folgenden Optionen zum Definieren der Ereignis-Nutzlast aus, die Sie Ihrer Lambda -Funktion als Eingabe bereitstellen möchten:
 - "json": JSON-formatierte Schlüssel-Wert-Paare, welche die Ereignisnutzlast definieren.
 - "path": Ein Pfad zu der Datei, die als Ereignisnutzlast verwendet wird.

Im folgenden Beispiel definiert die "json" Option die Nutzlast.

Weitere Informationen zu diesen und anderen Einträgen in `launch.json` finden Sie in [Konfigurationsoptionen für das Debuggen von serverless Anwendungen](#).

6.

Wenn Sie mit der Debug-Konfiguration zufrieden sind, klicken Sie auf den grünen Play-Pfeil neben RUN, um mit dem Debuggen zu beginnen.

Wenn die Debugging-Sitzungen gestartet werden, zeigt das Bedienfeld DEBUG CONSOLE die Debugging-Ausgabe und alle Werte an, die die Lambda-Funktion zurückgibt. (Beim Debuggen von AWS SAM Anwendungen wird AWSToolkit im Bedienfeld „Ausgabe“ als Ausgabekanal ausgewählt.)

Ausführen und Debuggen lokaler Amazon API Gateway Ressourcen

Sie können ausführen oder debuggen AWS SAM Lokale API-Gateway-Ressourcen, angegeben in `template.yaml`, indem Sie eine VS-Code-Startkonfiguration von `type=aws-sam` mit `debugTarget.target=api`.

Note

API Gateway unterstützt zwei Arten von APIs, REST und HTTP. Die API Gateway -Funktion mit dem AWS Toolkit for Visual Studio Code unterstützt nur REST-APIs. Manchmal werden HTTP-APIs „API Gateway V2-APIs“ genannt.

So führen und debuggen Sie lokale API Gateway Ressourcen

1. Wählen Sie einen der folgenden Ansätze zum Erstellen einer Startkonfiguration für eine AWS SAM API Gateway Ressource:
 - Option 1: Besuchen Sie den Handler-Quellcode (.js, .cs oder .py Datei) in Ihrem AWS SAM-Projekt, bewegen Sie den Mauszeiger über den Lambda-Handler und wählen Sie **Hinzufügen einer Debugkonfiguration CodeLens**. Wählen Sie dann im Menü den Eintrag **Ereignis**.
 - Option 2: Bearbeiten `launch.json` und erstellen Sie eine neue Startkonfiguration mithilfe der folgenden Syntax.

```
{
  "type": "aws-sam",
  "request": "direct-invoke",
  "name": "myConfig",
  "invokeTarget": {
    "target": "api",
    "templatePath": "n12/template.yaml",
    "logicalId": "HelloWorldFunction"
  },
  "api": {
    "path": "/hello",
    "httpMethod": "post",
    "payload": {
      "json": {}
    }
  },
  "sam": {},
  "aws": {}
}
```

2. Im VS-CodeFühren Sie Folgendes aus:wählen Sie die Startkonfiguration (benanntmyConfigim obigen Beispiel).
3. (Optional) Fügen Sie Ihrem Lambda Projektcode Haltepunkte hinzu.
4. TypF5oder wähleWiedergabeimFühren Sie Folgendes aus:-Bedienfeld.
5. Zeigen Sie im Ausgabefenster die Ergebnisse an.

Konfiguration

Wenn Sie den `invokeTarget.targetEigenschaftswertapi` benutzen, ändert das Toolkit die Validierung und das Verhalten der Startkonfiguration, um ein `api` -Feld zu unterstützen.

```
{
  "type": "aws-sam",
  "request": "direct-invoke",
  "name": "myConfig",
  "invokeTarget": {
    "target": "api",
    "templatePath": "n12/template.yaml",
    "logicalId": "HelloWorldFunction"
  },
  "api": {
    "path": "/hello",
    "httpMethod": "post",
    "payload": {
      "json": {}
    },
    "queryString": "abc=def&qrs=tuv",
    "headers": {
      "cookie": "name=value; name2=value2; name3=value3"
    }
  },
  "sam": {},
  "aws": {}
}
```

Ersetzen Sie die Werte im Beispiel wie folgt:

InvokeTarget.LogicalID

Eine API-Ressource.

path (Pfad)

Der API-Pfad, den die Startkonfiguration anfordert, z. B. "path": "/hello".

Muss ein gültiger API-Pfad sein, der aus der `template.yaml` angegeben von `invokeTarget.templatePath` ist.

httpMethod

Es kann eines der folgenden Verben sein: GET, POST, PUT, DELETE, PATCH, HEAD, OPTIONS.

Nutzlast

Die JSON-Payload (HTTP-Body), die in der Anforderung gesendet werden soll, mit der gleichen Struktur und den gleichen Regeln wie das Feld [lambda.payload](#).

`payload.path` verweist auf eine Datei, die die JSON-Payload enthält.

`payload.json` gibt eine JSON-Nutzlast inline an.

Header

Optionale Zuordnung von Name-Wert-Paaren, mit denen Sie HTTP-Header angeben, die in die Anforderung einbezogen werden sollen, wie im folgenden Beispiel dargestellt.

```
"headers": {
  "accept-encoding": "deflate, gzip;q=1.0, *;q=0.5",
  "accept-language": "fr-CH, fr;q=0.9, en;q=0.8, de;q=0.7, *;q=0.5",
  "cookie": "name=value; name2=value2; name3=value3",
  "user-agent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_14_6)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/86.0.4240.198 Safari/537.36",
}
```

querystring

Optionale Zeichenfolge, die die `querystring` Anforderung einstellt z. B. "`querystring`": "`abc=def&ghi=jkl`".

AWS

So stellen Sie die AWS Verbindungsinformationen zur Verfügung. Weitere Informationen finden Sie im `.AWSVerbindungseigenschaften` („aws“)Tabelle in der [Konfigurationsoptionen für das Debuggen von serverless Anwendungen](#)Abschnitts erstellt.

sam

So entwickelt AWS SAM CLI die Anwendung. Weitere Informationen finden Sie im `.AWSSAM-CLI-Eigenschaften` („sam“)Tabelle in der [Konfigurationsoptionen für das Debuggen von serverless Anwendungen](#)Abschnitts erstellt.

Konfigurationsoptionen für das Debuggen von serverless Anwendungen

Beim Öffnen der `launch.json`-Datei, um Debug-Konfigurationen zu bearbeiten, können Sie den VS-Code verwenden [IntelliSense](#) Funktion zum Anzeigen und automatischen Vervollständigen gültiger Eigenschaften. Auslöservolumen IntelliSense drücken Sie im Editor `Ctrl+Leertaste`.

```
"lambda": {  
  "runtime": "nodejs12.x",  
  "event": {  
    "json": {}  
  }  
}
```

IntelliSense ermöglicht es Ihnen, Eigenschaften zum Aufrufen von Lambda Funktionen direkt oder mit der AWS SAM-Vorlage. Sie können auch Eigenschaften für "lambda" definieren (wie die Funktion ausgeführt wird), "sam" (wie das AWS SAM CLI die Anwendung entwickelt) und "aws" (wie AWS die Verbindungsinformationen zur Verfügung stellt).


AWS SAM: Direkter Lambda-Handler aufrufen/vorlagenbasierter Lambda-Aufruf

Property (Eigenschaft)	Description (Beschreibung)
type	Gibt an, welche Erweiterung die Startkonfiguration verwaltet . Immer auf <code>aws-sam</code> setzen, um die AWS SAM CLI lokal zu erstellen und zu debuggen.
name	Gibt einen leserfreundlichen Namen an, der in der Liste Konfiguration des Debuggen Startstarts angezeigt wird.

Property (Eigenschaft)	Description (Beschreibung)
<code>request</code>	Gibt die Art der Konfiguration an von der benannten Erweiterung ausgeführt werden (<code>aws-sam</code>). Immer auf <code>direct-invoke</code> gesetzt, um die Lambda-Funktion zu starten.

Property (Eigenschaft)	Description (Beschreibung)
<code>invokeTarget</code>	<p>Gibt den Eintrittspunkt für den Aufruf der Ressource an.</p> <p>Um die Lambda Funktion direkt aufzurufen, legen Sie Werte für die folgenden <code>invokeTarget</code> -Felder:</p> <ul style="list-style-type: none">• <code>target</code>— Stellen Sie auf <code>code</code>.• <code>lambdaHandler</code> — Der Name des aufzurufenden Lambda - Funktionshandlers.• <code>projectRoot</code> — Der Pfad für die Anwendungsdatei, die Lambda-Funktions-Handler.• <code>architecture</code> — Prozessorarchitektur der emulierten Umgebung, in der Ihre lokale SAM-Lambda-Anwendung ausgeführt wird. Für bestimmte Laufzeiten können Sie wählen <code>arm64</code> anstelle der Standardeinstellung <code>x86_64</code> Architektur. Weitere Informationen finden Sie unter Eine neue serverlose Anwendung (lokal) erstellen. <p>Für den Aufruf der Lambda-Ressourcen mit dem AWS SAM-Vorlage, Werte für <code>invokeTarget</code> -Felder:</p> <ul style="list-style-type: none">• <code>target</code>— Stellen Sie auf <code>template</code>.• <code>templatePath</code> — Der Pfad zum AWS SAM-Vorlagendatei.• <code>logicalId</code> — Der Ressourcename des <code>AWS::Lambda::Function</code> oder <code>AWS::Serverless::Function</code> , der aufgerufen werden soll. Sie finden den Ressourcennamen im YAML-Format AWS SAM-Vorlage. Beachten Sie, AWS Toolkit erkennt implizit Funktionen, die mit <code>PackageType: Image</code> im AWS SAM-Vorlage als Image-basiert Lambda-Funktionen. Weitere Informationen finden Sie unter Lambda-Bereitstellungspakete im AWS Lambda Entwicklerhandbuch.

Lambda ("**lambda**") Eigenschaften

Property (Eigenschaft)	Description (Beschreibung)
<code>environmentVariables</code>	<p>Verwenden Sie, um Betriebsparameter an Ihre Lambda-Funktion weiterzugeben. Wenn Sie z. B. Daten in einen Amazon-S3-Bucket schreiben, anstatt den Bucket-Namen, Bucket-Namen als Umgebungsvariable zu codieren, hartzucodieren, hartzucodieren, hartzucodieren, hartzucodieren, hartzucodieren,</p> <div data-bbox="623 604 656 638"></div> <p>Note</p> <p>Wenn Sie Umgebungsvariablen für eine serverlose Anwendung angeben, müssen Sie Konfigurationen zu beiden AWS SAM-Vorlage (<code>template.yaml</code>) und das <code>launch.json</code> file.</p> <p>Beispiel für die Formatierung einer Umgebungsvariablen in der AWS SAM-Vorlage:</p> <pre>Resources: HelloWorldFunction: Type: AWS::Serverless::Function Properties: CodeUri: hello-world/ Handler: app.lambdaHandlerN10 Runtime: nodejs10.x Environment: Variables: SAMPLE1: Default Sample 1 Value</pre> <p>Beispiel für die Formatierung einer Umgebungsvariablen in der <code>launch.json</code> file:</p> <pre>"environmentVariables": { "SAMPLE1": "My sample 1 value" }</pre>

Property (Eigenschaft)	Description (Beschreibung)
	<ul style="list-style-type: none">• <code>"json"</code>: JSON-formatierte Schlüssel-Wert-Paare, welche die Ereignisnutzlast definieren.• <code>"path"</code>: Ein Pfad zu der Datei, die als Ereignisnutzlast verwendet wird.
<code>memoryMB</code>	Gibt Megabyte (MB) Speicher an, der für die Ausführung einer aufgerufenen Lambda-Funktion bereitgestellt wird.
<code>runtime</code>	Gibt die Laufzeitumgebung an, die Lambda-Funktion verwendet. Weitere Informationen finden Sie unter AWS Lambda-Laufzeiten .
<code>timeoutSec</code>	Legt die zulässige Zeit in Sekunden fest, bevor die Debug-Sitzung das Timeout überschreitet.

Property (Eigenschaft)	Description (Beschreibung)
pathMappings	<p>Gibt an, wo sich der lokale Code in Bezug auf die Ausführung im Container befindet.</p> <p>Standardmäßig setzt das Toolkit for VS Code <code>localRoot</code> zum Code-Root der Lambda-Funktion im lokalen Workspace und <code>remoteRoot</code> zu <code>/var/task</code>, das Standard-Arbeitsverzeichnis für Code, der in Lambda ausgeführt wird. Wenn das Arbeitsverzeichnis in der Dockerfile geändert wird oder mit dem <code>WorkingDirectory</code> -Parameter im AWS CloudFormation-Vorlagendatei, mindestens ein <code>pathMapping</code> -Eintrag muss angegeben werden, damit der Debugger erfolgreich lokal gesetzte Breakpoints dem im Lambda-Container ausgeführten Code zuordnen kann.</p> <p>Beispiel für Formatierung für <code>pathMappings</code> im <code>launch.json</code> file:</p> <pre data-bbox="597 982 1507 1304">"pathMappings": [{ "localRoot": " \${workspaceFolder}/sam-app/ HelloWorldFunction ", "remoteRoot": " /var/task " }]</pre> <p>Einschränkungen:</p> <ul style="list-style-type: none">• Für .NET-Image-basierte Lambda-Funktionen ist der <code>remoteRoot</code> -Eintrag muss das Build-Verzeichnis sein.• Für Node.js-basierte Lambda-Funktionen können Sie nur einen einzelnen Pfadzuordnungseintrag angeben.

Das Toolkit for VS Code verwendet den AWS SAM CLI zum Erstellen und Debuggen von serverless Anwendungen lokal. Sie können das Verhalten von AWS SAM CLI-Befehle, die Eigenschaften des "sam"-Konfiguration in der Datei `launch.json` konfigurieren.

AWS SAM CLI ("**sam**") Eigenschaften

Property (Eigenschaft)	Description (Beschreibung)	Standardwert
<code>buildArguments</code>	Konfiguriert, wie die <code>sam build</code> Ihren Lambda-Quellcode erstellt. Informationen zu Buildoptionen finden Sie unter sam build im AWS Serverless Application Model Entwicklerhandbuch.	Leere Zeichenfolge
<code>containerBuild</code>	Gibt an, ob Ihre Funktion innerhalb eines Lambda-ähnlichen Docker-Containers entwickelt wird.	<code>false</code>
<code>dockerNetwork</code>	Der Name oder die ID eines vorhandenen Docker-Netzwerks, mit dem Lambda Docker-Container eine Verbindung herstellen sollen, mit dem Standard-Bridge-Netzwerk. Bei fehlender Angabe stellen die Lambda-Container nur eine Verbindung mit dem Standard-Bridge-Docker-Netzwerk her.	Leere Zeichenfolge
<code>localArguments</code>	Gibt zusätzliche lokale Aufruf-Argumente an.	Leere Zeichenfolge
<code>skipNewImageCheck</code>	Gibt an, ob der Befehl das Herunterladen des neuesten Docker-Images für die Lambda -Laufzeitumgebung überspringen soll.	<code>false</code>

Property (Eigenschaft)	Description (Beschreibung)	Standardwert
template	Passt AWS SAM-Vorlage, die Parameter zur Eingabe von Kundenwerten verwendet. Weitere Informationen finden Sie unter Parameter im AWS CloudFormation-Benutzerhandbuch.	"parameters": {}

AWS-Verbindungseigenschaften ("aws")

Property (Eigenschaft)	Description (Beschreibung)	Standardwert
credentials	Wählt ein bestimmtes Profil aus (z. B. <code>profile: default</code>) aus Ihrer Anmeldeinformationsdatei, um AWS-Anmeldeinformationen zu erhalten.	Die AWS-Anmeldeinformationen, die Ihre vorhandene geteilte AWS Konfigurationsdatei oder freigegebene AWS Tools ebenfalls unterstützt stellen Sie im Toolkit for VS Code bereit.
region	Legt die AWS-Region des Services fest (z. B. <code>us-east-1</code>).	Die AWS-Region, die dem Profil der aktiven Anmeldeinformationen zugeordnet ist.

Beispiel: Konfiguration des Vorlagenstarts

Im Folgenden finden Sie ein Beispiel für Startkonfigurationsdatei für AWS SAM-Vorlagenziel:

```
{
  "configurations": [
    {
      "type": "aws-sam",
      "request": "direct-invoke",
      "name": "my-example:HelloWorldFunction",
      "invokeTarget": {
        "target": "template",
```



```
        "templatePath": "template.yaml",
        "logicalId": "HelloWorldFunction"
    },
    "lambda": {
        "payload": {},
        "environmentVariables": {}
    }
}
]
```

Beispiel: Startkonfiguration

Hier ist ein Beispiel für eine Startkonfigurationsdatei für ein Lambda-Funktionsziel:

```
{
  "configurations": [
    {
      "type": "aws-sam",
      "request": "direct-invoke",
      "name": "my-example:app.lambda_handler (python3.7)",
      "invokeTarget": {
        "target": "code",
        "projectRoot": "hello_world",
        "lambdaHandler": "app.lambda_handler"
      },
      "lambda": {
        "runtime": "python3.7",
        "payload": {},
        "environmentVariables": {}
      }
    }
  ]
}
```

Fehlerbehebung bei serverlosen Anwendungen

In diesem Thema werden häufig Fehler beschrieben, die beim Erstellen serverloser Anwendungen mit dem Toolkit for VS Code auftreten können und wie diese behoben werden können.

Themen

- [Wie kann ich eine samconfig.toml mit einer SAM-Launch-Konfiguration verwenden?](#)

- [Fehler: „RuntimeError: Container ist nicht vorhanden“](#)
- [Fehler: „Docker.Errors.ApiError: 500 Server-Fehler... Sie haben Ihr Pullrate-Limit erreicht.“](#)
- [Fehler: „500 Serverfehler: Einhängen von C:\Users\...“](#)
- [Verwenden von WSL, Webviews \(z. B. das „Aufrufen amAWS“-Formular\) sind kaputt](#)
- [Debuggen einer TypeScript-Anwendung, aber Haltepunkte funktionieren nicht](#)

Wie kann ich eine `samconfig.toml` mit einer SAM-Launch-Konfiguration verwenden?

Geben Sie den Speicherort Ihrer SAM CLI [samconfig.toml](#) durch Konfigurieren des `--config-file` Argumentes in der `localArguments` Eigenschaft Ihrer Startkonfiguration. Wenn sich die `samconfig.toml`-Datei beispielsweise auf der obersten Ebene Ihres Workspace befindet:

```
"sam": {  
  "localArguments": ["--config-file", "${workspaceFolder}/samconfig.toml"],  
}
```

Fehler: „RuntimeError: Container ist nicht vorhanden“

Dieses `build` kann diesen Fehler anzeigen, wenn Ihr System nicht genügend Speicherplatz für den Docker-Container hat. Wenn Ihr Systemspeicher nur 1-2 GB Speicherplatz zur Verfügung hat, kann `sam build` während der Verarbeitung fehlschlagen, auch wenn der Systemspeicher vor Beginn des Builds nicht vollständig voll ist. Weitere Informationen finden Sie unter [dieses GitHub-Problem](#) aus.

Fehler: „Docker.Errors.ApiError: 500 Server-Fehler... Sie haben Ihr Pullrate-Limit erreicht.“

Docker Hub begrenzt Anfragen, die anonyme Benutzer stellen können. Wenn Ihr System das Limit erreicht, schlägt Docker fehl und dieser Fehler erscheint in der OUTPUT-Ansicht von VS-Code:

```
docker.errors.APIError: 500 Server Error: Internal Server Error ("toomanyrequests: You  
have  
reached your pull rate limit. You may increase the limit by authenticating and  
upgrading:  
https://www.docker.com/increase-rate-limit")
```

Stellen Sie sicher, dass Ihr System Docker Service hat sich mit Ihren Docker Hub-Anmeldeinformationen authentifiziert.

Fehler: „500 Serverfehler: Einhängen von C:\Users\...“

Windows-Benutzer sehen möglicherweise diesen Docker-Einhängefehler beim Debuggen AWS SAM-Anwendungen:

```
Fetching lambci/lambda:nodejs10.x Docker container image.....
2019-07-12 13:36:58 Mounting C:\Users\\AppData\Local\Temp\ ... as /var/
task:ro,delegated inside runtime container
Traceback (most recent call last):
...
requests.exceptions.HTTPError: 500 Server Error: Internal Server Error ...
```

Aktualisieren Sie die Anmeldeinformationen für Ihre freigegebenen Laufwerke (in den Docker-Einstellungen).

Verwenden von WSL, Webviews (z. B. das „Aufrufen am AWS“-Formular) sind kaputt

Dies ist ein bekanntes VS-Code-Problem für Benutzer von Cisco VPN. Weitere Informationen finden Sie unter [dieses GitHub-Problem](#) aus.

Eine Problemumgehung wird in vorgeschlagen [dieses WSL-Tracking-Problem](#) aus.

Debuggen einer TypeScript-Anwendung, aber Haltepunkte funktionieren nicht

Dies geschieht, wenn es keine Quellzuordnung gibt, um die kompilierte JavaScript-Datei mit der Quell-TypeScript-Datei zu verknüpfen. Öffnen Sie Ihre `tsconfig.json`-Datei und stellen Sie sicher, dass die folgende Option und der folgende Wert festgelegt sind: `"inlineSourceMap": true` aus.

Arbeiten mit Systems-Manager-Automation-Dokumenten

AWS Systems Manager bietet Ihnen Transparenz und Kontrolle über Ihre Infrastruktur auf AWS. Systems Manager verfügt über eine einheitliche Benutzeroberfläche zum Anzeigen von operativen Daten aus mehreren AWS-Services, damit Sie operative Aufgaben in Ihren AWS-Ressourcen automatisieren können.

Ein [Systems-Manager-Dokument](#) definiert die Aktionen, die Systems Manager auf Ihren verwalteten Instances durchführt. Ein Automatisierungsdokument ist eine Art Systems-Manager-Dokument, das Sie verwenden, um allgemeine Wartungs- und Bereitstellungsaufgaben wie das Erstellen oder Aktualisieren eines Amazon Machine Image (AMI) durchzuführen. In diesem Thema wird beschrieben, wie Dokumente zur Automatisierung mit AWS Toolkit for Visual Studio Code.

Themen

- [Annahmen und Voraussetzungen](#)
- [IAM-Berechtigungen für Dokumente zur Automatisierung von Systems Manager](#)
- [Erstellen eines neuen Systems-Manager-Automation-dokuments](#)
- [Öffnen eines bestehenden Systems-Manager-Automation-dokuments](#)
- [Bearbeiten eines Systems-Manager-Automation-Dokuments](#)
- [Veröffentlichen eines Dokuments zur Automatisierung von Systems Manager](#)
- [Löschen eines Systems-Manager-Automation-Dokuments](#)
- [Ausführen eines Systems-Manager-Automation-Dokuments](#)
- [Problembehandlung bei Systems Manager Automation-Dokumenten in Toolkit for VS Code](#)

Annahmen und Voraussetzungen

Bevor Sie beginnen, stellen Sie Folgendes sicher:

- Sie haben Visual Studio Code und die neueste Version des AWS Toolkit for Visual Studio Code. Weitere Informationen finden Sie unter [Installieren des AWS Toolkit for Visual Studio Code](#).
- Sie sind mit Systems Manager vertraut. Weitere Informationen finden Sie im [AWS Systems Manager-Benutzerhandbuch](#).
- Sie sind mit den Anwendungsfällen der Automatisierung von Systems Manager Automation vertraut. Weitere Informationen finden Sie unter [AWS Systems-Manager-Automatisierung](#) im AWS Systems Manager-Benutzerhandbuch.

IAM-Berechtigungen für Dokumente zur Automatisierung von Systems Manager

Im Toolkit for VS Code müssen Sie über ein Anmeldeprofil verfügen, das AWS Identity and Access Management (IAM) -Berechtigungen, die zum Erstellen, Veröffentlichen und Löschen von Dokumenten zur Automatisierung von Systems Manager Automation-Dokumenten erforderlich sind. Das folgende Richtliniendokument definiert die erforderlichen IAM-Berechtigungen, die in einer Hauptrichtlinie verwendet werden können:

```
{
```

```
"Version": "2012-10-17",
"Statement": [
  {
    "Effect": "Allow",
    "Action": [
      "ssm:ListDocuments",
      "ssm:ListDocumentVersions",
      "ssm:DescribeDocument",
      "ssm:GetDocument",
      "ssm:CreateDocument",
      "ssm:UpdateDocument",
      "ssm:UpdateDocumentDefaultVersion",
      "ssm>DeleteDocument"
    ],
    "Resource": "*"
  }
]
```

Informationen zum Aktualisieren einer IAM-Richtlinie finden Sie unter [Erstellen von IAM-Richtlinien](#) im IAM-Benutzerhandbuch. Informationen zum Einrichten Ihres Anmeldeprofils finden Sie unter [AWS IAM-Anmeldeinformationen](#).

Erstellen eines neuen Systems-Manager-Automation-dokuments

Sie können ein neues Automatisierungsdokument inJSONoderYAMLmit Visual Studio Code. Wenn Sie ein neues Automatisierungsdokument erstellen, wird es in einer unbenannten Datei dargestellt. Sie können Ihre Datei benennen und in VS-Code speichern, der Name der Datei ist jedoch nicht sichtbarAWS.

Erstellen Sie ein neues Automation-dokument

1. Öffnen Sie VS-Code.
2. Auf derAnzeigenMenü, wählen Sie-Befehlspaletteum die Befehlspalette zu öffnen.
3. Geben Sie in der BefehlspaletteAWSToolkit Erstellen Sie lokal ein neues Systems-Manager-Dokument.
4. Wählen Sie eine der Startervorlagen für ein Hello-World-Beispiel aus.
5. Wählen Sie JSON oder YAML.

Ein neues Automatisierungsdokument wird erstellt.

Note

Ihr neues Automatisierungsdokument in VS Code erscheint nicht automatisch in AWS. Sie müssen es auf AWS veröffentlichen, bevor Sie es ausführen können.

Öffnen eines bestehenden Systems-Manager-Automation-dokuments

Du benutzt AWS Explorer, um vorhandene Dokumente zur Automatisierung von Systems Manager Automation-Explorer. Wenn Sie ein vorhandenes Automation-dokument öffnen, wird es als unbenannte Datei in VS-Code angezeigt.

So öffnen Sie Ihr Automatisierungsdokument

1. Öffnen Sie VS-Code.
2. Wählen Sie in der linken Navigationsleiste AWS Sum AWS-Explorer.
3. In der AWS-Explorer, für Systems Manager, wählen Sie das Downloadsymbol auf dem Dokument, das Sie öffnen möchten, und wählen Sie dann die Dokumentversion aus. Die Datei wird im Format für diese Version geöffnet. Andernfalls wählen Sie entweder Als JSON herunterladen oder Als YAML herunterladen.

Note

Wenn Sie ein Automatisierungsdokument lokal als Datei in VS Code speichern, wird es nicht in AWS Explorer angezeigt. Es muss veröffentlicht werden auf AWS vor der Ausführung.

Bearbeiten eines Systems-Manager-Automation-Dokuments

Wenn Sie Automatisierungsdokumente besitzen, werden diese in der **Im Besitz von mir** Kategorie der Systems Manager Manager-Dokumente in der AWS Explorer. Sie können Automatisierungsdokumente besitzen, die bereits in AWS, und Sie können Eigentümer neuer oder aktualisierter Dokumente sein, in denen Sie zuvor veröffentlicht haben AWS von VS Code.

Wenn Sie ein Automationsdokument zur Bearbeitung in VS Code öffnen, können Sie mehr daraus machen als im AWS Management Console. Zum Beispiel:

- Es gibt eine Schemaüberprüfung für beide JSON und YAML-Formate.

- Im Dokumenteditor stehen Snippets zur Verfügung, mit denen Sie jeden der Automatisierungsschritttypen erstellen können.
- Es gibt Unterstützung für die automatische Vervollständigung verschiedener Optionen inJSONundYAML.

Arbeiten mit Versionen

Dokumente zur Automatisierung von Systems Manager Automation-Versionen für das Änderungsmanagement. Sie können die Standardversion für ein Automatisierungsdokument in VS Code auswählen.

Legen Sie eine Standardversion fest wie folgt

- Navigieren Sie im AWS Explorer zu dem Dokument, für das Sie die Standardversion festlegen möchten, öffnen Sie das Kontextmenü (Rechtsklick) für das Dokument und wählen Sie Set default version (Standardversion festlegen).

Note

Wenn das ausgewählte Dokument nur eine Version enthält, können Sie den Standardwert nicht ändern.

Veröffentlichen eines Dokuments zur Automatisierung von Systems Manager

Nachdem Sie Ihr Automatisierungsdokument in VS-Code bearbeitet haben, können Sie es aufAWS.

So veröffentlichen Sie Ihr Automatisierungsdokument

1. Öffnen Sie das Automationdokument, das Sie veröffentlichen möchten, mit dem in beschriebenen Verfahren [Öffnen eines bestehenden Systems-Manager-Automation-dokuments](#).
2. Nehmen Sie die Änderungen vor, die veröffentlicht werden sollen. Weitere Informationen finden Sie unter [Bearbeiten eines Systems-Manager-Automation-Dokuments](#).
3. Wählen Sie rechts unten in der geöffneten Datei das Uploadsymbol aus.
4. Wählen Sie im Dialogfeld des VeröffentlichungsarbeitsablaufsAWS-Region, in der Sie das Automatisierungsdokument veröffentlichen möchten

5. Wenn Sie ein neues Dokument veröffentlichen, wählen Sie **Quick Create**. Wählen Sie andernfalls **Quick Update** um ein vorhandenes Automatisierungsdokument darin zu aktualisieren **AWS Region** :
6. Geben Sie den Namen für dieses Automatisierungsdokument ein.

Wenn Sie ein Update für ein vorhandenes Automatisierungsdokument in veröffentlichen **AWS** wird dem Dokument eine neue Version hinzugefügt.

Löschen eines Systems-Manager-Automation-Dokuments

Sie können Automatisierungsdokumente in VS-Code löschen. Durch das Löschen eines Automatisierungsdokuments werden das Dokument und alle Versionen des Dokuments gelöscht.

Important

- Löschen ist eine zerstörerische Aktion, die nicht rückgängig gemacht werden kann.
- Durch das Löschen eines Automatisierungsdokuments, das bereits ausgeführt wurde, wird das **AWS Ressourcen**, die beim Start erstellt oder geändert wurden.

So löschen Sie Ihr Automatisierungsdokument

1. Öffnen Sie VS-Code.
2. Wählen Sie in der linken Navigationsleiste **AWS** **Sum AWS-Explorer**.
3. In der **AWS-Explorer**, für **Systems Manager**, öffnen Sie das Kontextmenü (rechte Maustaste) für das Dokument, das Sie löschen möchten, und wählen Sie **Dokument löschen**.

Ausführen eines Systems-Manager-Automation-Dokuments

Sobald Ihr Automatisierungsdokument in veröffentlicht wurde **AWS** können Sie es ausführen, um Aufgaben in Ihrem Namen in Ihrem **AWS Konto**. Um Ihr Automatisierungsdokument auszuführen, verwenden Sie die **AWS Management Console**, die **Systems-Manager-APIs**, die **AWS CLI** oder **AWS Tools for PowerShell**. Anweisungen zum Ausführen eines Automationdokuments finden Sie unter [Ausführen einer einfachen Automatisierung](#) im **AWS Systems Manager Benutzerhandbuch**.

Wenn Sie alternativ eines der **AWS-SDKs** mit den **Systems-Manager-APIs** verwenden möchten, um Ihr Automatisierungsdokument auszuführen, lesen Sie die [AWS-SDK-Referenzen](#).

Note

Durch das Ausführen eines Automatisierungsdokuments können neue Ressourcen in AWS und kann Abrechnungskosten verursachen. Wir empfehlen Ihnen dringend nachzuvollziehen, was Ihr Automatisierungsdokument in Ihrem Konto erstellt, bevor Sie es starten.

Problembehandlung bei Systems Manager Automation-Dokumenten in Toolkit for VS Code

Ich habe mein Automatisierungsdokument in VS-Code gespeichert, aber ich sehe es nicht in AWS Management Console.

Durch das Speichern eines Automatisierungsdokuments in VS-Code wird das Automatisierungsdokument nicht in AWS. Weitere Informationen zum Veröffentlichen Ihres Automatisierungsdokuments finden Sie unter [Veröffentlichen eines Dokuments zur Automatisierung von Systems Manager](#).

Die Veröffentlichung meines Automatisierungsdokuments ist wegen eines Berechtigungsfehlers fehlgeschlagen.

Stellen Sie sicher, dass Ihr AWS-Anmeldeprofil über die erforderlichen Berechtigungen zum Veröffentlichen von Dokumenten zur Automatisierung verfügt. Ein Beispiel für eine Berechtigungsrichtlinie finden Sie unter [IAM-Berechtigungen für Dokumente zur Automatisierung von Systems Manager](#).

Ich habe mein Automatisierungsdokument veröffentlicht in AWS, aber ich sehe es nicht in AWS Management Console.

Stellen Sie sicher, dass Sie das Dokument in derselben veröffentlicht haben AWS Region, in der Sie sich gerade befinden AWS Management Console.

Ich habe mein Automatisierungsdokument gelöscht, die erstellten Ressourcen werden mir jedoch weiterhin in Rechnung gestellt.

Das Löschen eines Automationdokuments löscht nicht die Ressourcen, die es erstellt oder geändert hat. Sie können die von Ihnen erstellten AWS-Ressourcen in der [AWS-Managementkonsole zur Fakturierung](#) identifizieren, Ihre Gebühren untersuchen und auswählen, welche Ressourcen von dort gelöscht werden sollen.

Arbeiten mit AWS Step Functions

Der AWS Toolkit for Visual Studio Code (VS Code) bietet Unterstützung für [AWS Step Functions](#). Mit dem Toolkit for VS Code können Sie Step Functions Functions-Zustandsmaschinen erstellen, aktualisieren und ausführen.

Themen

- [Arbeiten mit AWS Step Functions](#)

Arbeiten mit AWS Step Functions

Sie können den AWS Toolkit for Visual Studio Code (VS-Code) verwenden, um verschiedene Operationen mit [Zustandsmaschinen](#) auszuführen.

Themen

- [Voraussetzungen](#)
- [Arbeiten Sie mit Zustandsmaschinen in VS Code](#)
- [Vorlagen für Zustandsmaschinen](#)
- [Visualisierung von Zustandsmaschinen](#)
- [Codefragmente](#)
- [Vervollständigung und Validierung des Codes](#)

Voraussetzungen

- Stellen Sie sicher, dass Ihr System die unter [Installation des Toolkit for VS Code](#) angegebenen Voraussetzungen erfüllt, und installieren Sie dann das Toolkit.
- Stellen Sie sicher, dass Sie Ihre Anmeldeinformationen konfiguriert haben, bevor Sie den AWS Explorer öffnen.

Arbeiten Sie mit Zustandsmaschinen in VS Code

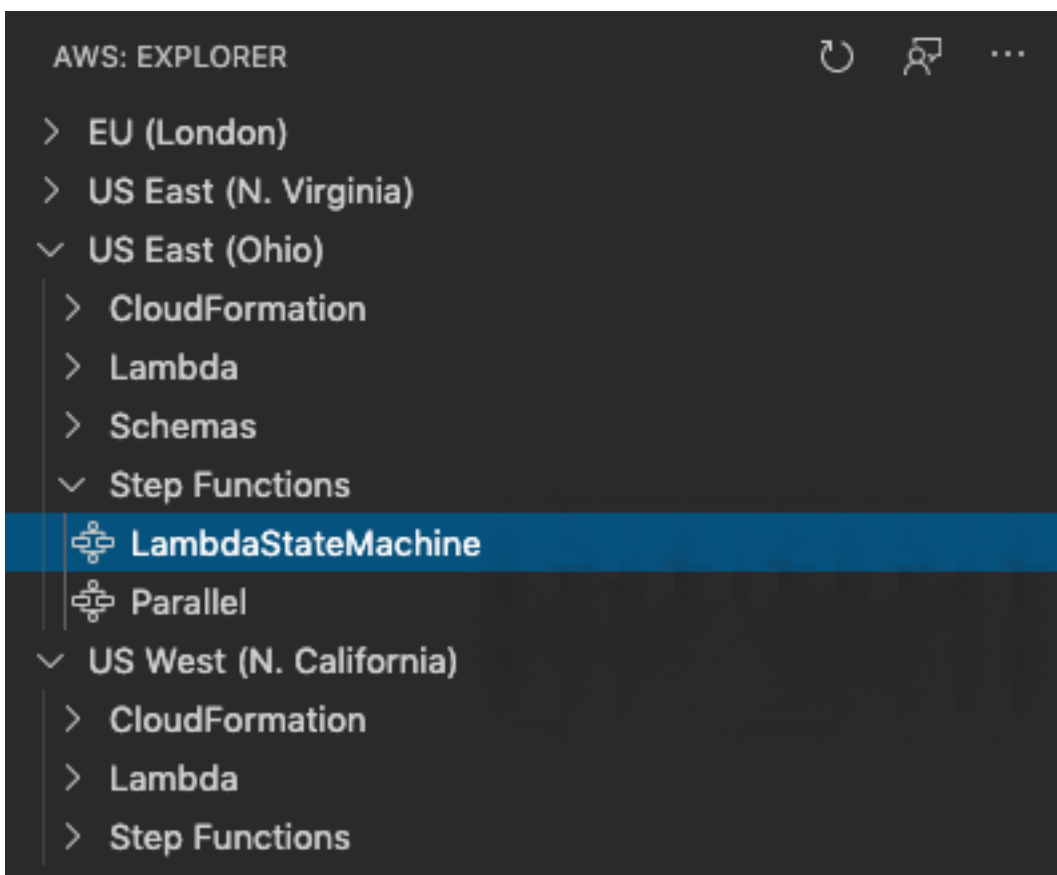
Sie können VS Code verwenden, um mit Remote-Zustandsmaschinen zu interagieren und Zustandsmaschinen lokal im JSON- oder YAML-Format zu entwickeln. Sie können Zustandsmaschinen erstellen oder aktualisieren, vorhandene Zustandsmaschinen auflisten,

sie ausführen und herunterladen. Mit VS Code können Sie auch neue Zustandsautomaten aus Vorlagen erstellen, eine Visualisierung Ihres Zustandsautomaten anzeigen und Codeausschnitte, Codevervollständigung und Codevalidierung bereitstellen.

Vorhandene Zustandsmaschinen auflisten

Wenn Sie bereits Zustandsautomaten erstellt haben, können Sie eine Liste dieser Automaten anzeigen:

1. Öffnen Sie den AWS Explorer.
2. Wählen Sie Step Functions
3. Prüfen Sie, ob alle Zustandsautomaten in Ihrem Konto aufgelistet sind.

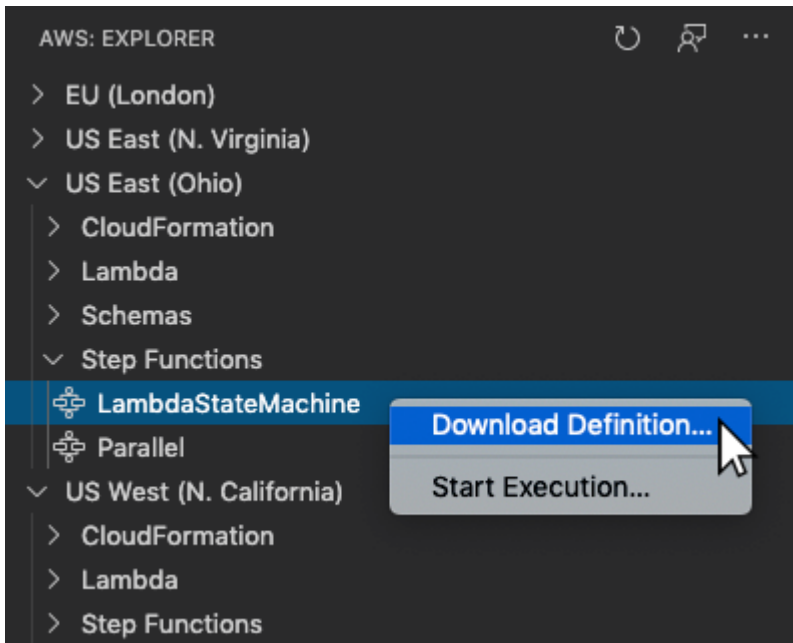


Laden Sie eine Zustandsmaschine herunter

So laden Sie einen Zustandsautomaten herunter:

1. Klicken Sie im AWS Explorer mit der rechten Maustaste auf den State Machine, den Sie herunterladen möchten.

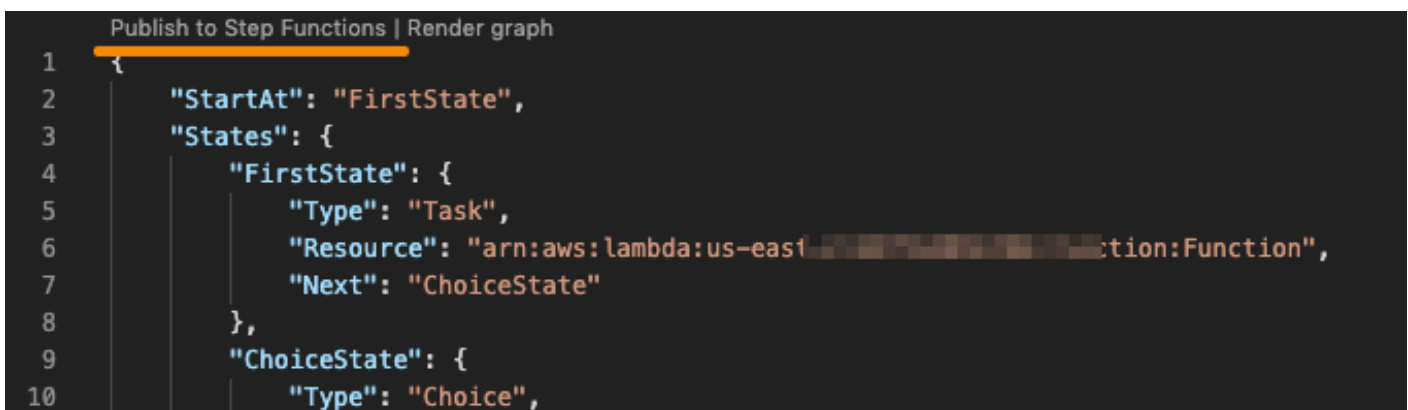
2. Wählen Sie „Download (Herunterladen)“ aus, und wählen Sie dann den Speicherort aus, an dem Sie den Zustandsautomaten herunterladen möchten.
3. Überprüfen Sie, ob er richtig heruntergeladen wurde.



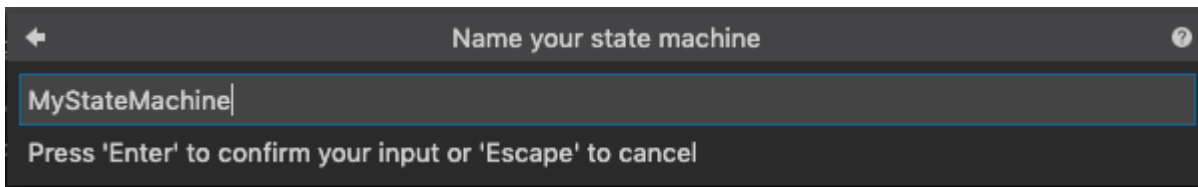
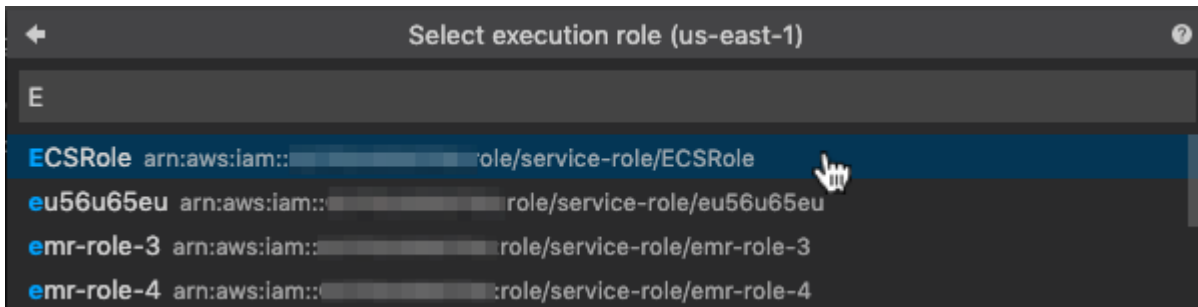
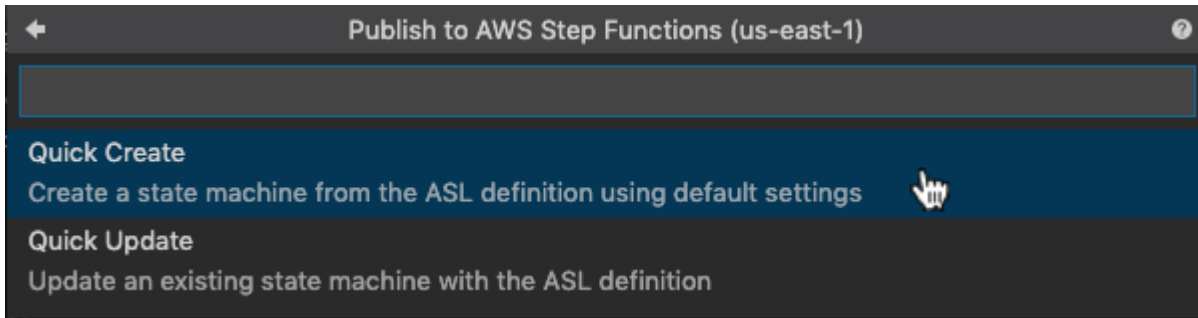
Erstellen eines Zustandsautomaten

Sie können selbst einen neuen Zustandsautomaten erstellen oder eine Vorlage verwenden. Weitere Informationen zum Erstellen eines Zustandsautomaten aus einer Vorlage finden Sie im Abschnitt [Vorlagen für Zustandsautomaten](#). So erstellen Sie einen neuen Zustandsautomaten:

1. Erstellen Sie eine neue [Amazon States Language](#) (ASL)-Datei mit Ihrer Zustandsautomatendefinition. Verwenden Sie das Menü unten rechts, um es als Amazon States Language festzulegen.
2. Wählen Sie [Publish to Step Functions](#) (Zu Schrittfunktionen veröffentlichen).



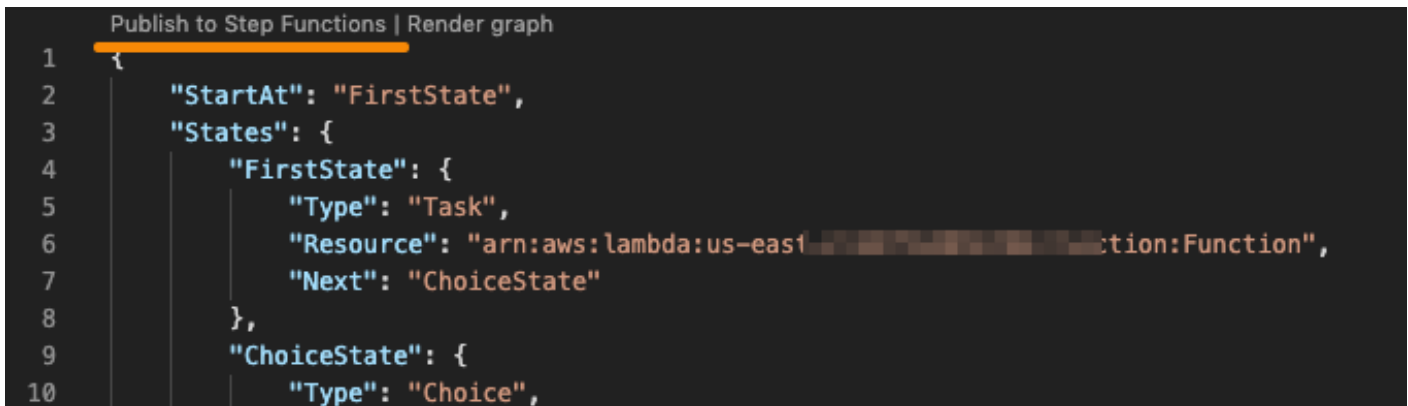
- Wählen Sie Quick Create (Schnellerstellung), wählen Sie eine Rolle aus, und benennen Sie Ihren Zustandsautomaten.



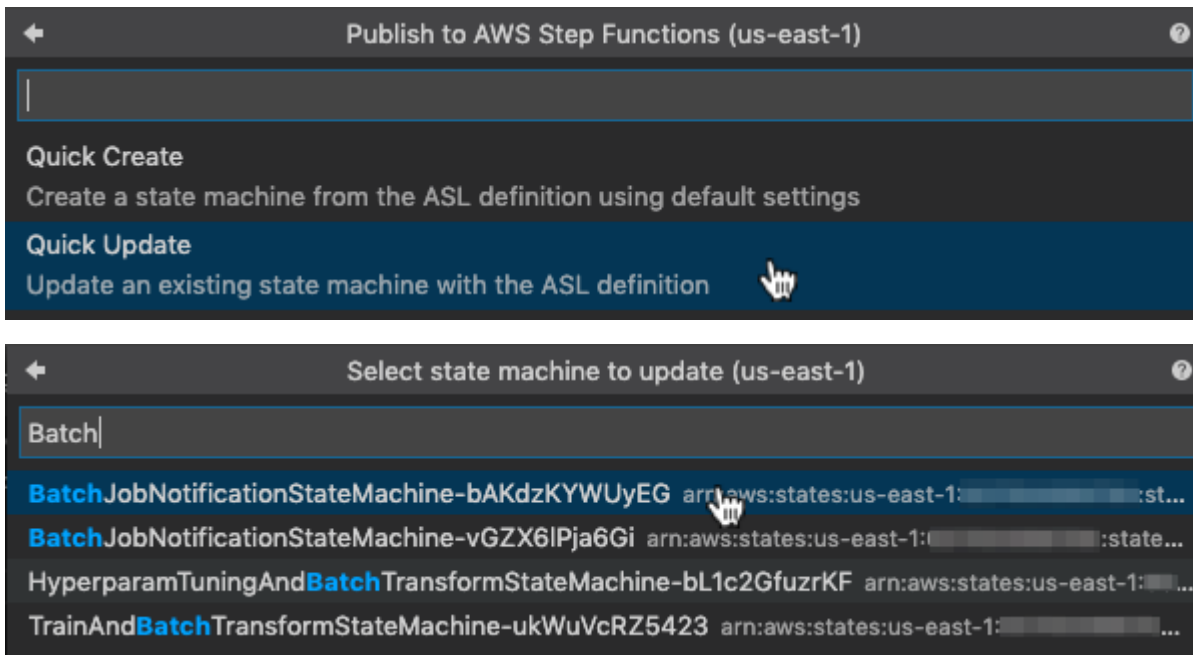
Aktualisieren eines Zustandsautomaten

So aktualisieren Sie einen Zustandsautomaten:

- Bearbeiten Sie die ASL-Datei mit Ihrer Zustandsautomatendefinition.
- Wählen Sie Publish to Step Functions (Zu Schrittfunktionen veröffentlichen).



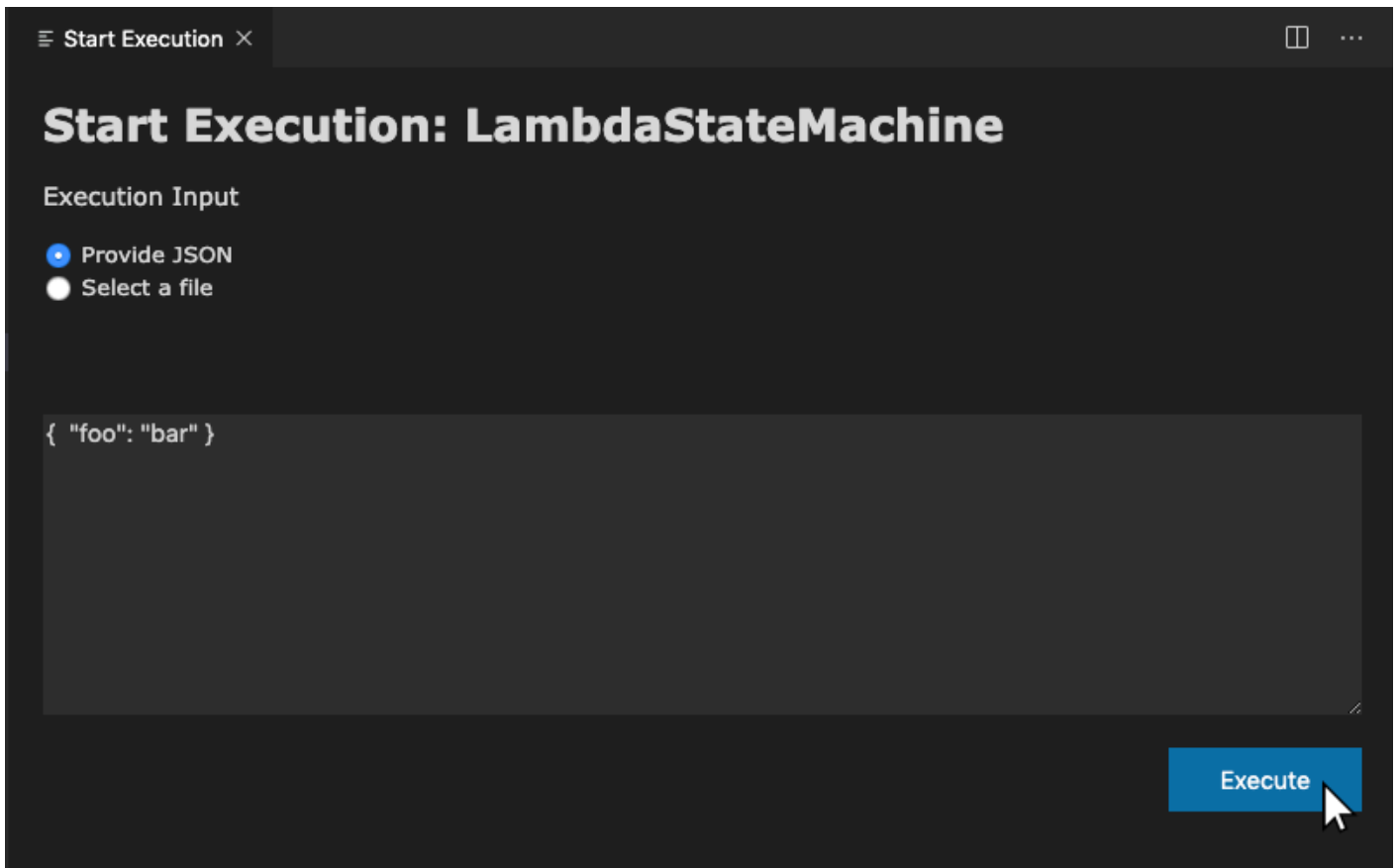
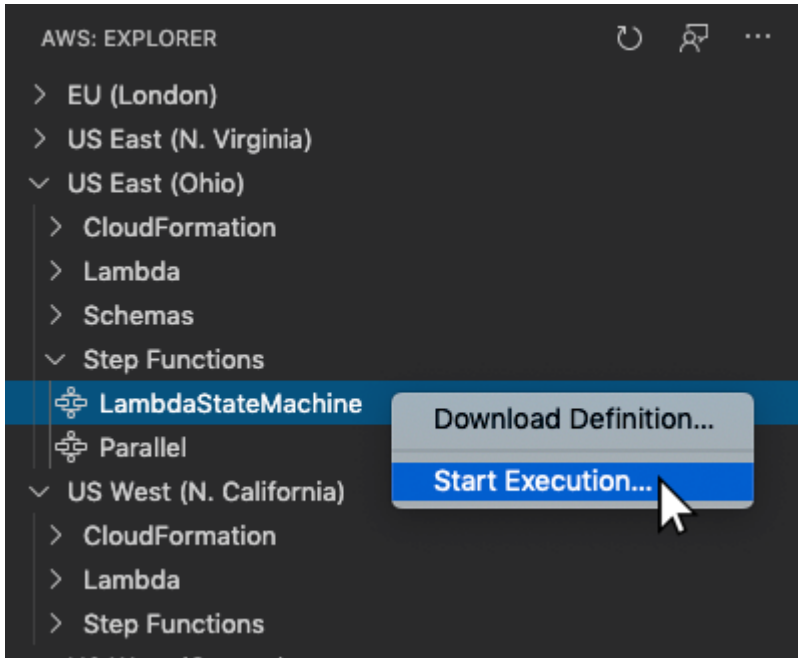
3. Wählen Sie Quick Update (Schnellaktualisierung) aus, und wählen Sie dann den Zustandsautomaten aus, den Sie aktualisieren möchten.



Führen Sie eine Zustandsmaschine aus

Um eine Zustandsmaschine auszuführen:

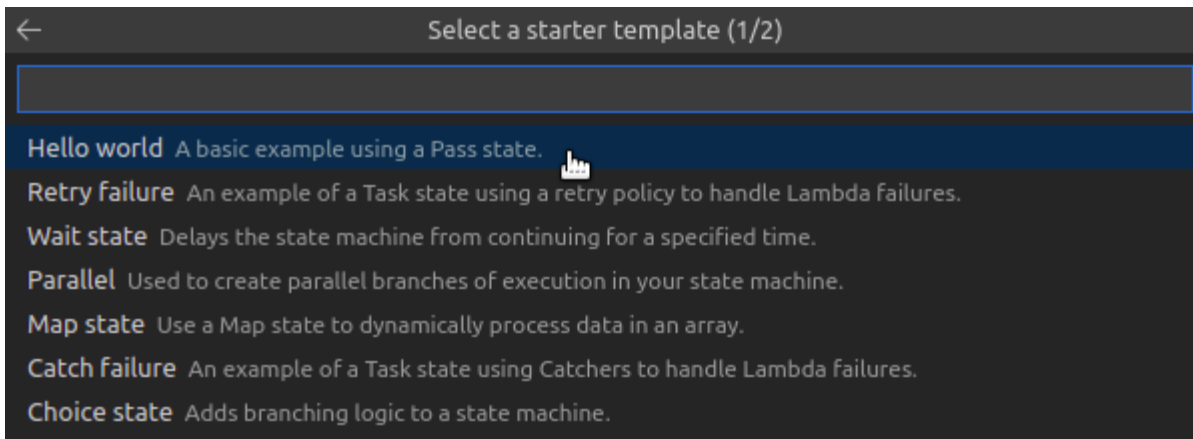
1. Klicken Sie im AWS Explorer mit der rechten Maustaste auf den Zustandsmaschine, den Sie ausführen möchten.
2. Geben Sie Eingaben für die Zustandsmaschine ein. Sie können sowohl die Eingabe aus einer Datei als auch die Eingabe in ein Textfeld versuchen.
3. Starten Sie den Zustandsmaschine und stellen Sie sicher, dass er erfolgreich ausgeführt wird.



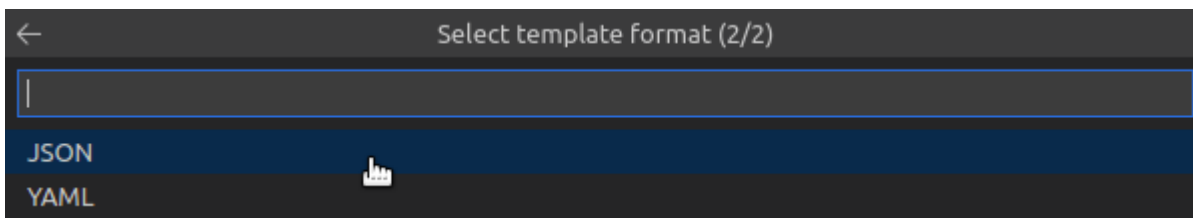
Vorlagen für Zustandsmaschinen

Wenn Sie einen Zustandsautomaten erstellen, haben Sie die Möglichkeit, ihn aus einer Vorlage zu erstellen. Die Vorlage enthält eine Beispielformatdefinition für einen Zustandsautomaten mit mehreren häufig verwendeten Zuständen und stellt Ihnen einen Startpunkt zur Verfügung. So verwenden Sie Vorlagen für Zustandsautomaten::

1. Öffnen Sie die Befehlspalette in VS Code.
2. Wählen Sie AWS Toolkit Create a new Step Functions State Machine aus.
3. Wählen Sie die gewünschte Vorlage aus.



4. Wählen Sie aus, ob Sie das JSON- oder das YAML-Vorlagenformat verwenden möchten.



Visualisierung von Zustandsmaschinen

Mit grafischen Visualisierungen können Sie sehen, wie Ihr Zustandsautomat im grafischen Format aussieht. Wenn Sie eine Graphvisualisierung erstellen, wird eine weitere Registerkarte geöffnet, auf der eine Visualisierung der Zustandsmaschine JSON oder YAML angezeigt wird. Sie können dann die Zustandsautomatendefinition, die Sie schreiben, gleichzeitig mit der Visualisierung vergleichen. Wenn Sie die Zustandsautomatendefinition ändern, wird die Visualisierung aktualisiert.

Note

Um eine Visualisierung einer Zustandsautomatendefinition zu erstellen, muss die Definition im aktiven Editor geöffnet sein. Wenn Sie die Definitionsdatei schließen oder umbenennen, wird die Visualisierung geschlossen.

So erstellen Sie eine Visualisierung eines Zustandsautomatendiagramms:

1. Definieren Sie Ihren Zustandsautomaten.
2. Öffnen Sie die Befehlspalette in VS Code.
3. Um eine Visualisierung zu erstellen, verwenden Sie die Visualisierungsschaltfläche in der oberen rechten Ecke oder wählen Sie Diagramm AWS rendern.

The screenshot displays the AWS Step Functions ASL editor in VS Code. The left pane shows the JSON definition for a state machine named 'Choice-example.asl.json'. The right pane shows the rendered state machine diagram.

```

1  {
2    "Comment": "An example of the Amazon States Language using a choice
3    state.",
4    "StartAt": "FirstState",
5    "States": {
6      "FirstState": {
7        "Type": "Task",
8        "Resource":
9        "arn:aws:lambda:us-east-2:637554956784:function:Function",
10       "Next": "ChoiceState"
11     },
12     "ChoiceState": {
13       "Type": "Choice",
14       "Choices": [
15         {
16           "Variable": "$.foo",
17           "NumericEquals": 1,
18           "Next": "FirstMatchState"
19         },
20         {
21           "Variable": "$.foo",
22           "NumericEquals": 2,
23           "Next": "SecondMatchState"
24         }
25       ],
26       "Default": "DefaultState"
27     },
28     "FirstMatchState": {
29       "Type": "Task",
30       "Resource":
31       "arn:aws:lambda:us-east-2:637554956784:function:Function",
32       "Next": "NextState"
33     },
34     "SecondMatchState": {
35       "Type": "Task",
36       "Resource":
37       "arn:aws:lambda:us-east-2:637554956784:function:Function",
38       "Next": "NextState"
39     },
40     "NextState": {
41       "Type": "Task",
42       "Resource":
43       "arn:aws:lambda:us-east-2:637554956784:function:Function",
44       "Next": "End"
45     },
46     "DefaultState": {
47       "Type": "Task",
48       "Resource":
49       "arn:aws:lambda:us-east-2:637554956784:function:Function",
50       "Next": "End"
51     },
52     "End": {}
53   }
54 }

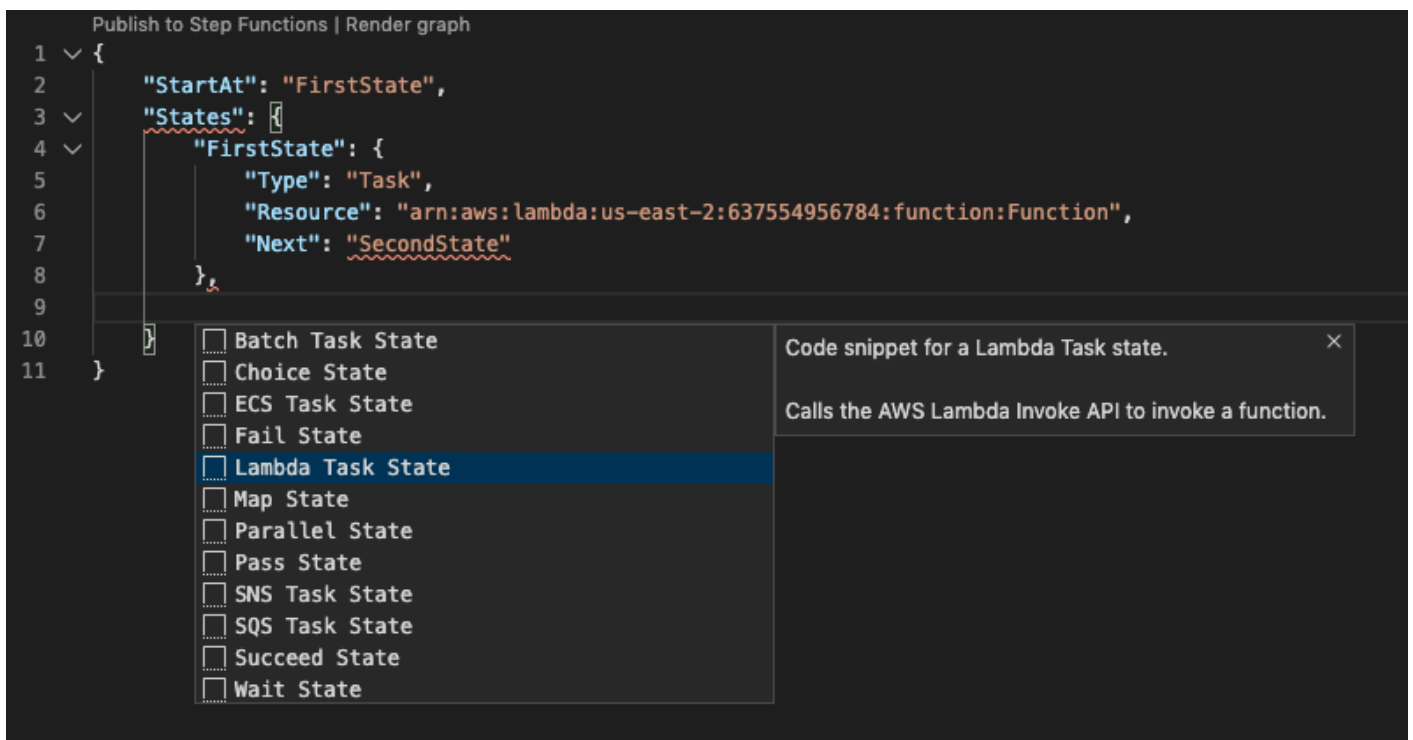
```

The diagram on the right illustrates the state machine flow. It starts at a 'Start' node, leading to 'FirstState'. From 'FirstState', the flow goes to 'ChoiceState'. 'ChoiceState' branches into 'FirstMatchState' and 'SecondMatchState'. Both 'FirstMatchState' and 'SecondMatchState' lead to 'NextState'. 'NextState' leads to 'End'. Additionally, 'ChoiceState' has a 'Default' path leading to 'DefaultState', which also leads to 'End'.

Codefragmente

Mit Codeausschnitten können Sie kurze Codeteile einfügen. So verwenden Sie Codeausschnitte:

1. Öffnen Sie eine Datei und speichern Sie sie mit der Erweiterung `.asl.json` für das JSON-Format oder `.asl.yaml` für das YAML-Format.
2. Erstellen Sie einen neuen Zustandsautomaten mit der Eigenschaft `States` (Zustände).
3. Platzieren Sie den Cursor innerhalb von `States` (Zustände).
4. Verwenden Sie die Tastenkombination `Control + Space`, und wählen Sie Ihren bevorzugten Codeausschnitt aus.
5. Verwenden Sie `Tab`, um die Variable und Parameter in dem Codeausschnitt zu durchlaufen.
6. Testen Sie die Ausschnitte `Retry` und `Catch`, indem Sie den Cursor in den jeweiligen Zustand setzen.



Vervollständigung und Validierung des Codes

So sehen Sie, wie die Code-Vervollständigung funktioniert:

1. Erstellen Sie mehrere Zustände.
2. Platzieren Sie den Cursor hinter einer `Next` `StartAt`- oder `Default`-Eigenschaft.
3. Verwenden Sie die Tastenkombination `Control + Space`, um verfügbare Vervollständigungen aufzulisten. Sie können erneut mit `Control + Space` auf zusätzliche Eigenschaften zugreifen, basierend auf der `Type` von `State`.

4. Während der Arbeit erfolgt die Code-Validierung für:

- Fehlende Eigenschaften
- Inkorrekte Werte
- Fehlenden Beendigungsstatus
- Nicht vorhandene Zustände, auf die verwiesen wird

```

"FirstMatchState": {
  "Type": "Task",
  "Resource": "arn:aws:lambda:us-east-2:637554956784:function:Function",
  "Next": ""
},
"SecondMatchState": {
  "Type": "ChoiceState",
  "Resource": "arn:aws:lambda:us-east-2:637554956784:function:Function",
  "Next": "SecondMatchState"
},
"DefaultState": {
  "Type": "Fail",
  "Error": "DefaultStateError",
  "Cause": "No Matches!"
},
"NextState": {
  "Type": "Task",
  "Resource": "arn:aws:lambda:us-east-2:637554956784:function:Function",
  "End": true
}

```

```

"FirstMatchState": {
  "Type": "Task",
  "Resource": "arn:aws:lambda:us-east-2:637554956784:function:Function",
  "Catch": [
    {
      "Error": "DefaultStateError",
      "Cause": "No Matches!"
    }
  ]
},

```

An array of objects, called Catchers, that define a fallback state. This state is executed if the state encounters runtime errors and its retry policy is exhausted or isn't defined.

Mit Threat Composer arbeiten

Sie können das verwenden AWS Toolkit for Visual Studio Code , um mit dem Threat Composer-Tool zu arbeiten. Threat Composer ist ein Tool zur Bedrohungsmodellierung, das Ihren Prozess zur Bedrohungsmodellierung vereinfachen kann.

Ausführliche Informationen zum Threat Composer-Tool finden Sie im [Threat GitHub Composer-Repository](#).

Die folgenden Themen beschreiben die Arbeit mit Threat Composer im AWS Toolkit for Visual Studio Code.

Themen

- [Arbeiten mit Threat Composer aus dem Toolkit](#)

Arbeiten mit Threat Composer aus dem Toolkit

Mit Threat Composer können Sie Threat Composer-Bedrohungsmodelle direkt in VS Code erstellen, anzeigen und bearbeiten. Detaillierte Informationen zum Threat Composer-Tool finden Sie im [Threat GitHub Composer-Repository](#).

In den folgenden Abschnitten wird beschrieben, wie Sie auf die Threat Composer-Tools in der zugreifen können AWS Toolkit for Visual Studio Code.

Über das Toolkit auf Threat Composer zugreifen

Es gibt drei Hauptmethoden, wie Sie über das Toolkit auf Threat Composer zugreifen können.

Zugriff auf Threat Composer über ein vorhandenes Bedrohungsmodell

Um Threat Composer zu öffnen, öffnen Sie eine bestehende Bedrohungsmodell-Datei (Erweiterung .tc.json) in VS Code. Threat Composer öffnet automatisch und rendert eine Visualisierung Ihrer Bedrohungsmodelldatei im VS Code-Editor-Fenster.

Ein neues Threat Composer-Bedrohungsmodell erstellen

1. Erweitern Sie im Hauptmenü von VS Code die Option Datei und wählen Sie dann Neue Datei aus.
2. Wählen Sie im Dialogfeld „Neue Datei“ die Option Threat Composer-Datei... .

3. Wenn Sie dazu aufgefordert werden, geben Sie `a` ein und drücken Sie dann die **enter** Eingabetastefile `name`, um Threat Composer zu öffnen und eine Visualisierung Ihrer leeren Bedrohungsmodelldatei in einem neuen VS Code-Editor-Fenster zu erstellen.

Ein neues Threat Composer-Bedrohungsmodell über die Befehlspalette erstellen

1. Öffnen Sie in VS Code die Befehlspalette, indem Sie auf **Cmd + Shift + P** oder **Ctrl + Shift + P** (Windows) drücken.
2. Geben Sie in das Suchfeld „Neue Threat Composer-Datei erstellen“ ein **Threat Composer** und wählen Sie sie aus, wenn sie in den Ergebnissen angezeigt wird.
3. Wenn Sie dazu aufgefordert werden, geben Sie `a` ein und drücken Sie dann die **enter** Eingabetastefile `name`, um Threat Composer zu öffnen und eine Visualisierung Ihrer leeren Bedrohungsmodelldatei in einem neuen VS Code-Editor-Fenster zu erstellen.

Arbeiten mit -Ressourcen

Neben dem Zugriff für AWS-Services, die standardmäßig im AWS Explorer aufgelistet sind, können Sie auch zu Ressourcen gehen und aus Hunderten von Ressourcen auswählen, die Sie zur Schnittstelle hinzufügen möchten. In AWS ist eine Ressource eine Entität, mit der Sie arbeiten können. Zu den Ressourcen, die hinzugefügt werden können, gehört Amazon AppFlow, Amazon Kinesis Data Streams, AWS IAM-Rollen, Amazon VPC und Amazon CloudFront -Verteilungen.

Nachdem Sie Ihre Auswahl getroffen haben, können Sie zu Ressourcen und erweitern Sie den Ressourcentyp, um die verfügbaren Ressourcen für diesen Typ aufzulisten. Wenn Sie zum Beispiel `AWS Toolkit::Lambda::Function`-Ressourcentyp wählen, können Sie auf die Ressourcen zugreifen, die verschiedene Funktionen, ihre Eigenschaften und ihre Attribute definieren.

Nach dem Hinzufügen eines Ressourcentyps zu Ressourcen können Sie auf folgende Weise mit ihm und seinen Ressourcen interagieren:

- Zeigen Sie eine Liste der vorhandenen Ressourcen an, die in der aktuellen AWS-Region für diesen Ressourcentyp.
- Zeigen Sie eine schreibgeschützte Version der JSON-Datei an, die eine Ressource beschreibt.
- Kopieren Sie die Ressourcenkennung für die Ressource.
- Zeigen Sie die AWS-Dokumentation an, die den Zweck des Ressourcentyps und des Schemas (in JSON- und YAML-Formaten) für die Modellierung einer Ressource erläutert.

- Erstellen Sie eine neue Ressource, indem Sie eine Vorlage im JSON-Format bearbeiten und speichern, die einem Schema entspricht.*
- Aktualisieren oder löschen Sie eine vorhandene Ressource.*

Important

*In der aktuellen Version des AWS Toolkit for Visual Studio Code Die Option zum Erstellen, Bearbeiten und Löschen von Ressourcen ist ein experimentelles Feature. Da experimentelle Funktionen weiterhin getestet und aktualisiert werden, können sie Probleme mit der Benutzerfreundlichkeit haben. Und experimentelle Funktionen können aus dem entfernt werden AWS Toolkit for Visual Studio Code ohne vorherige Ankündigung.

Um die Verwendung von experimentellen Funktionen für Ressourcen zu ermöglichen, öffnen Sie die Einstellungen in Ihrer VS Code IDE, und erweitern Sie Erweiterungen und wähle AWS Toolkit.

Under AWS Toolkit-Experimente wählen Sie jsonResourceModification um Ressourcen zu erstellen, zu aktualisieren und zu löschen.

Weitere Informationen finden Sie unter [Arbeiten mit experimentellen Funktionen](#).

IAM-Berechtigungen für den Zugriff auf Ressourcen

Sie benötigen spezifische AWS Identity and Access Management-Berechtigungen für den Zugriff auf die Ressourcen, die AWS-Services zugeordnet sind. Beispielsweise benötigt eine IAM-Entität, wie ein Benutzer oder eine Rolle, Lambda-Berechtigungen, um auf `AWS Toolkit:Lambda::Function`-Ressourcen zuzugreifen.

Zusätzlich zu den Berechtigungen für Service-Ressourcen benötigt eine IAM-Entität Berechtigungen, um die Toolkit for VS Code zum Aufrufen zu berechtigen AWS Cloud-Control-API-Operationen in seinem Namen. Cloud-Control-API-Vorgänge ermöglichen dem IAM-Benutzer oder der IAM-Rolle, auf die Remote-Ressourcen zuzugreifen und diese zu aktualisieren.

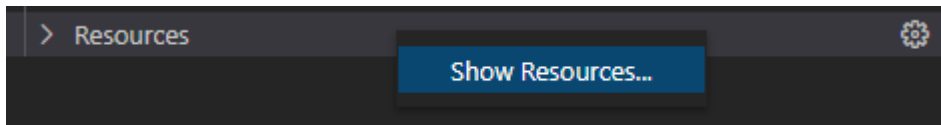
Die einfachste Methode zum Erteilen von Berechtigungen besteht darin, eine AWS-Verwaltete Richtlinie, `PowerUserAccess` an die IAM-Entität anruft, die diese API-Operationen über die Toolkit-Schnittstelle aufruft. Diese [verwaltete Richtlinie](#) erteilt eine Reihe von Berechtigungen zum Ausführen von Anwendungsentwicklungsaufgaben, einschließlich des Aufrufs von API-Operationen.

Bestimmte Berechtigungen, die zulässige API-Vorgänge für Remote-Ressourcen definieren, finden Sie im [API-Benutzerhandbuch zu AWS-Cloud-Control](#).

Hinzufügen und Interaktion mit bestehenden Ressourcen

1. In der AWSExplorer, rechtsklicken Ressourcen und wählen Ressourcen einblenden.

In einem Bereich wird eine Liste von Ressourcentypen angezeigt, die zur Auswahl stehen.



2. Wählen Sie im Auswahlbereich die Ressourcentypen aus, die zum hinzugefügt werden sollen AWSExplorer und drücken Sie Ergebnis oder wählen OKAY zur Bestätigung.

Die Ressourcentypen, die Sie ausgewählt haben, werden unter Ressourcen.

Note

Wenn Sie bereits einen Ressourcentyp zum AWSExplorer und deaktivieren Sie dann das Kontrollkästchen für diesen Typ, er ist nicht mehr unter Ressourcen nachdem du dich entschieden hast OKAY. Nur die Ressourcentypen, die derzeit ausgewählt sind, sind in der AWSExplorer.

3. Um die Ressourcen anzuzeigen, die bereits für einen Ressourcentyp vorhanden sind, erweitern Sie den Eintrag für diesen Typ.

Eine Liste der verfügbaren Ressourcen wird unter ihrem Ressourcentyp angezeigt.

4. Klicken Sie mit der rechten Maustaste auf den Namen und wählen Sie eine der folgenden Optionen aus, um mit einer bestimmten Ressource zu interagieren:
 - Kopieren eines Ressourcen-: Kopieren Sie den Bezeichner für die spezifische Ressource in die Zwischenablage. (Beispielsweise kann die `AWS Toolkit:DynamoDB::Table`-Ressource mit der `TableName`-Eigenschaft identifiziert werden.)
 - -Vorschau: Zeigen Sie eine schreibgeschützte Version der JSON-formatierten Vorlage an, die die Ressource beschreibt.

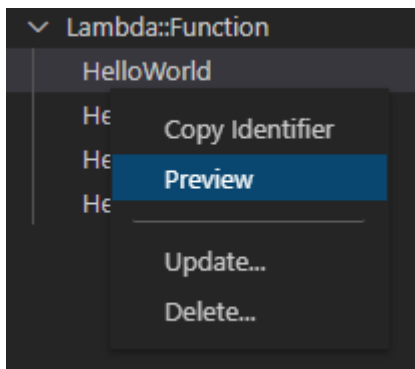
Nachdem die Ressourcenvorlage angezeigt wurde, können Sie sie ändern, indem Sie die Option Aktualisierung Symbol auf der rechten Seite der Editor-Registerkarte. Um eine Ressource zu aktualisieren, benötigen Sie die erforderlichen [???](#)aktiviert.

- Aktualisierung: Bearbeiten Sie die JSON-formatierte Vorlage für die Ressource in einem VS-Code-Editor. Weitere Informationen finden Sie unter [Erstellen und Bearbeiten von Ressourcen](#).

- Löschen: Löschen Sie die Ressource, indem Sie das Löschen in einem angezeigten Dialogfeld bestätigen. (Das Löschen von Ressourcen ist derzeit ein [Bugs](#) in dieser Version von AWS Toolkit for Visual Studio Code.)

Warning

Wenn Sie eine Ressource löschen, werden alle AWS CloudFormation Stapel, die diese Ressource verwendet, nicht aktualisiert werden. Um diesen Aktualisierungsfehler zu beheben, müssen Sie entweder die Ressource neu erstellen oder den Verweis darauf im Stapel entfernen AWS CloudFormation Vorlage. Weitere Informationen finden Sie im [Wissenscenter-Artikel](#).



Erstellen und Bearbeiten von Ressourcen

Important

Die Erstellung und Aktualisierung von Ressourcen ist derzeit ein [Bugs](#) in dieser Version des AWS Toolkit for Visual Studio Code.

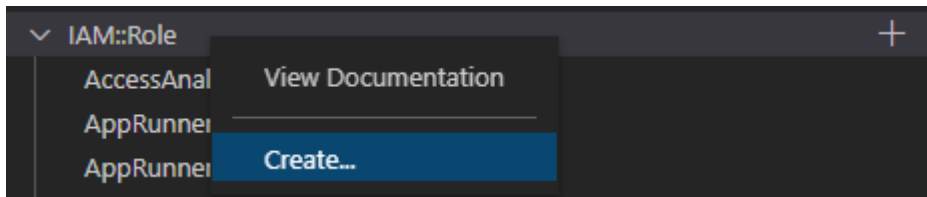
Das Erstellen einer neuen Ressource beinhaltet das Hinzufügen eines Ressourcentyps zur Ressourcenliste und dann eine JSON-formatierte Vorlage bearbeiten, die die Ressource, ihre Eigenschaften und Attribute definiert.

Eine Ressource, die zum `AWS Toolkit:SageMaker::UserProfile` Ressourcentyp wird mit einer Vorlage definiert, die ein Benutzerprofil für Amazon erstellt SageMaker Studio. Die Vorlage, die diese Benutzerprofilressource definiert, muss dem Ressourcentypschema für entsprechen AWS

`Toolkit:SageMaker::UserProfile`. Wenn die Vorlage beispielsweise aufgrund fehlender oder falscher Eigenschaften dem Schema nicht entspricht, kann die Ressource nicht erstellt oder aktualisiert werden.

1. Fügen Sie den Ressourcentyp für die Ressource hinzu, die Sie erstellen möchten, indem Sie `mitRessourcen` und wählen `Ressourcen` einblenden.
2. Nachdem der Ressourcentyp unter `hinzugefügt` wurde `Ressourcen`, wählen Sie das Plusymbol („+“), um die Vorlagendatei in einem neuen Editor zu öffnen.

Sie können auch mit der rechten Maustaste auf den Namen des Ressourcentyps klicken und Geben Sie einen Namen für den Benutzer ein und klicken Sie dann auf. Sie können auch auf Informationen zum Modellieren der Ressource zugreifen, indem Sie `View Documentation` (Dokumentation anzeigen).



3. Definieren Sie im Editor Eigenschaften, aus denen sich die Ressourcenvorlage zusammensetzt. Die Funktion zur automatischen Vervollständigung schlägt Eigenschaftsnamen vor, die dem Schema Ihrer Vorlage entsprechen. Wenn Sie den Mauszeiger über einen Eigenschaftstyp bewegen, wird in einem Bereich eine Beschreibung dessen Verwendung angezeigt. Ausführliche Informationen zum Schema finden Sie im `View Documentation` (Dokumentation anzeigen).

Jeder Text, der nicht dem Ressourcenschema entspricht, wird durch eine wellenförmige rote Unterstreichung gekennzeichnet.

```

1  {
2    "Role": "arn:aws:iam::488247187723:role/service-role/HelloWorld-role",
3    ""
4    "Code"
5    "CodeSigningConfigArn"
6    "DeadLetterConfig"
7    "Environment"
8    "ImageConfig"
9    "KmsKeyArn"
10   "Layers"
11   "ReservedConcurrentExecutions"
12   "Tags"
13   "VpcConfig"
14   "Handler": "index.handler",
15   "Arn": "arn:aws:lambda:us-west-2:488247187723:function:HelloWorld"
16 }

```

4. Nachdem Sie mit der Deklaration Ihrer Ressource fertig sind, wählen Sie den Save-Symbol, um Ihre Vorlage zu validieren und die Ressource auf der Fernbedienung zu speichern AWS Cloud.

Wenn Ihre Vorlage die Ressource gemäß ihrem Schema definiert, wird eine Meldung angezeigt, um zu bestätigen, dass die Ressource erstellt wurde. (Wenn die Ressource bereits existiert, bestätigt die Meldung, dass die Ressource aktualisiert wurde.)

Nachdem die Ressource erstellt wurde, wird sie der Liste unter der Überschrift Ressourcentyp hinzugefügt.

5. Wenn Ihre Datei Fehler enthält, wird eine Meldung angezeigt, die erklärt, dass die Ressource nicht erstellt oder aktualisiert werden konnte. Klicken Sie auf Anzeigen von Protokollnummern, um die Vorlagenelemente zu identifizieren, die Sie reparieren müssen.

Sicherheit für AWS Toolkit for Visual Studio Code

Themen

- [Datenschutz in AWS Toolkit for Visual Studio Code](#)

Datenschutz in AWS Toolkit for Visual Studio Code

Das AWS [Modell](#) der gilt für den Datenschutz im AWS Toolkit for Visual Studio Code. Wie in diesem Modell beschrieben, ist AWS für den Schutz der globalen Infrastruktur verantwortlich, 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 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 mit AWS Toolkit for Visual Studio Code oder anderen Programmen AWS-Services über die Konsole AWS CLI, API oder AWS SDKs 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.

Dokumentverlauf für das AWS Toolkit for Visual Studio Code-Benutzerhandbuch

Letzte wichtige Aktualisierung der Dokumentation: 14. Oktober 2021

In der folgenden Tabelle sind wichtige Änderungen in den einzelnen Versionen des AWS Toolkit for Visual Studio Code beschrieben. Für Benachrichtigungen über Aktualisierungen dieser Dokumentation können Sie einen [RSS-Feed](#) abonnieren.

Änderung	Beschreibung	Datum
Benutzeranleitung erstellt: Erstellen Sie eine Vorlage CloudFormation	Es wurde ein neues Benutzerhandbuch erstellt, in dem beschrieben wird, wie eine CloudFormation Vorlage mit dem Toolkit for VS Code erstellt wird	17. Dezember 2021
Kleineres UI-Update	Der bestehende Text für „Maschinenstatus in der Vorschau anzeigen“ wurde auf „Diagramm rendern“ aktualisiert, um ihn besser an die Benutzeroberfläche anzupassen.	14. Dezember 2021
Benutzerhandbuch für Amazon Elastic Container Service Exec wurde erstellt	Dies ist ein Überblick über Amazon ECS Exec.	13. Dezember 2021
Benutzerleitfaden für den AWS IoT Toolkit for VS Code Code-Dienst erstellt	Dieses Benutzerhandbuch soll Ihnen den Einstieg in die Nutzung des AWS IoT Dienstes für Toolkit for VS Code erleichtern.	22. November 2021

Unterstützung experimenteller Funktionen	Unterstützung der Aktivierung experimenteller Funktionen für AWS-Services hinzugefügt	14. Oktober 2021
Unterstützung für AWS-Ressourcen	Unterstützung des Zugriffs auf Ressourcentypen sowie Schnittstellenoptionen zum Erstellen, Bearbeiten und Löschen von Ressourcen hinzugefügt	14. Oktober 2021
Überblick über den Amazon ECR-Service für AWS Toolkit for Visual Studio Code	Es wurden eine Übersicht und eine exemplarische Vorgehensweise für die Merkmale und Funktionen des Amazon ECR-Service hinzugefügt, auf die in VS Code zugegriffen werden kann	14. Oktober 2021
Support für ARM64-Umgebungen	Sie können jetzt serverlose Anwendungen sowohl in ARM64-basierten emulierten Umgebungen als auch in x86_64-basierten Umgebungen ausführen.	1. Oktober 2021
AWS-Serverless-Anwendung	Unterstützung für die Ausführung von Anwendungen auf der ARM64-Plattform hinzugefügt AWS SAM	30. September 2021
Abschnitt „Formatieren Sie das Update Node.js“	Laut Kundenfeedback wurde die Formatierung für Node.js/TypeScript aktualisiert.	12. August 2021
Support für App Runner	Zusätzliche Unterstützung für AWS App Runner und AWS Toolkit for Visual Studio Code.	11. August 2021

Go-Funktionen debuggen	Unterstützung für das Debuggen lokaler Go-Funktionen hinzugefügt.	10. Mai 2021
Debuggen von Java-Funktionen	Unterstützung für das Debuggen lokaler Java-Funktionen hinzugefügt.	22. April 2021
YAML-Unterstützung für AWS Step Functions	YAML-Unterstützung für hinzugefügt. AWS Step Functions	4. März 2021
Debuggen von Amazon API Gateway Gateway-Resources	Unterstützung für das Debuggen lokaler Amazon API Gateway Gateway-Resources hinzugefügt.	1. Dezember 2020
Amazon API Gateway	Unterstützung für Amazon API Gateway hinzugefügt.	1. Dezember 2020
AWS-Serverless-Anwendung	Unterstützung für Lambda-Container-Images mit serverlosen Anwendungen hinzugefügt.	1. Dezember 2020
AWS Systems Manager--Support	Unterstützung für Systems Manager Automation-Dokumente hinzugefügt.	30. September 2020
CloudWatch Protokolle	Unterstützung für CloudWatch Logs hinzugefügt.	24. August 2020
Amazon S3	Unterstützung für Amazon S3 hinzugefügt.	30. Juli 2020
AWS Step Functions--Support	Unterstützung für hinzugefügt AWS Step Functions.	31. März 2020
Inhalt zum Thema Sicherheit	Sicherheitsinhalte hinzugefügt.	6. Februar 2020

Arbeiten mit EventBridge Amazon-Schemas	Unterstützung für Amazon EventBridge Schemas wurde hinzugefügt	1. Dezember 2019
AWS CDK	Vorschauversion des AWS CDK Dienstes.	25. November 2019
Verwenden eines externen Anmeldeprozesses	Es wurden Informationen zur Verwendung eines externen Anmeldeverfahrens zum Abrufen AWS von Anmeldeinformationen hinzugefügt.	25. September 2019
Wird IntelliSense für Aufgabendefinitionsdateien verwendet	IntelliSense Unterstützung für die Arbeit mit Amazon ECS-Aufgabendefinitionsdateien wurde hinzugefügt.	24. September 2019
Benutzerhandbuch für AWS Toolkit for Visual Studio Code	Veröffentlichung für die allgemeine verfügbare Version.	11. Juli 2019
Benutzerhandbuch für den AWS Toolkit for Visual Studio Code	Die Dokumentstruktur wurde der Übersichtlichkeit und Benutzerfreundlichkeit halber aktualisiert.	3. Juli 2019
Installieren des AWS Toolkit for Visual Studio Code	Es wurden Informationen zum Installieren von Sprach-SDKs zur Unterstützung verschiedener Toolketten hinzugefügt.	12. Juni 2019
Konfigurieren Sie Ihre Toolchain	Es wurden Informationen zur Konfiguration verschiedener Toolchains hinzugefügt.	12. Juni 2019

Erste Veröffentlichung

Erstveröffentlichung des
Benutzerhandbuchs für AWS
Toolkit for Visual Studio Code.

28. März 2019

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.