Benutzerhandbuch

AWS Toolkit for VS Code

Copyright © 2025 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

AWS Toolkit for VS Code: Benutzerhandbuch

Copyright © 2025 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 Handelsmarken, die nicht Eigentum von Amazon sind, gehören den jeweiligen Besitzern, die möglicherweise zu Amazon gehören oder nicht, 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
Das Anmeldefenster öffnen 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
Konfiguration Ihrer Toolchain 10
Konfigurieren Sie eine Toolchain für .NET Core 11
Konfigurieren Sie eine Toolchain für Node.js 11
Eine Toolchain für Python konfigurieren 11
Konfigurieren Sie eine Toolchain für Java 12
Konfigurieren Sie eine Toolchain für Go 12
Verwenden Sie Ihre Toolchain 13
Authentifizierung und Zugriff 14
IAM Identity Center 14
IAM-Anmeldeinformationen 15
Erstellen eines IAM-Benutzers 15
Erstellen einer Datei mit gemeinsamen Anmeldeinformationen aus dem AWS Toolkit for Visual Studio Code 16
Fügen Sie zusätzliche Anmeldeinformationsprofile hinzu 17

AWS Builder-ID ... 18
Verwendung eines externen Anmeldeprozesses ... 18
Aktualisierung von Firewalls und Gateways ... 19
AWS Toolkit for Visual Studio Code Endpunkte ... 19
Endpunkte des Amazon Q-Plug-ins ... 20
Amazon Q Developer-Endpunkte ... 20
Endpunkte für Amazon Q Code Transform ... 20
Endpunkte für die Authentifizierung ... 21
Identitätsendpunkte ... 21
Telemetrie ... 22
Referenzen ... 22
Arbeiten mit AWS ... 23
Experimentelle Funktionen ... 24
AWS Entdecker ... 24
AWS -Documents ... 26
Erste Schritte mit Dokumenten AWS ... 26
Dokumentation, Autovervollständigung und Validierung in VS Code anzeigen ... 26
Amazon CodeCatalyst ... 27
Was ist Amazon CodeCatalyst? ... 27
Erste Schritte mit CodeCatalyst ... 28
Arbeiten mit CodeCatalyst-Ressourcen ... 28
Mit Entwicklungsumgebungen arbeiten ... 32
Fehlerbehebung ... 35
Amazon API Gateway ... 36
AWS App Runner ... 37
Voraussetzungen ... 38
Preisgestaltung ... 41
Erstellen von App-Runner-Services ... 41
Verwalten von App-Runner-Services ... 45
AWS Anwendungsgenerator ... 48
Mit AWS Application Builder arbeiten ... 48
AWS Infrastruktur-Komponist ... 52
Arbeiten mit AWS Infrastructure Composer ... 53
AWS CDK ... 54
AWS CDK Anwendungen ... 55
AWS CloudFormation Stapel ... 57

Einen AWS CloudFormation Stapel löschen ... 57
Erstellen Sie eine CloudFormation Vorlage ... 58
CloudWatch Amazon-Protokolle ... 60
CloudWatch Protokollgruppen und Protokolldatenströme anzeigen ... 60
Arbeiten mit CloudWatch Protokollereignissen ... 62
Protokollgruppen durchsuchen ... 64
CloudWatch Protokolliert Live-Tail ... 66
Amazon DocumentDB ... 68
Arbeiten mit Amazon DocumentDB ... 69
Amazon EC2 ... 74
Mit Amazon arbeiten EC2 ... 75
Problembehebung bei Amazon EC2 ... 84
Amazon ECR ... 87
Arbeiten mit Amazon ECR ... 88
Einen App Runner-Dienst erstellen ... 98
Amazon ECS ... 100
Wird IntelliSense für Aufgabendefinitionsdateien verwendet ... 100
Amazon ECS Exec ... 102
Amazon EventBridge ... 104
Mit EventBridge Amazon-Schemas arbeiten ... 104
AWS IAM-Zugriffsanalysator ... 107
Arbeiten mit AWS IAM Access Analyzer ... 107
AWS IoT ... 112
AWS IoT Voraussetzungen ... 112
AWS IoT Dinge ... 112
AWS IoT Zertifikate ... 114
AWS IoT Richtlinien ... 118
AWS Lambda Funktionen ... 121
Mit AWS Lambda Funktionen arbeiten ... 122
Lambda-Debugging per Fernzugriff ... 129
AWS Lambda console zu IDE ... 139
Amazon Redshift ... 140
Arbeiten mit Amazon Redshift ... 141
Amazon S3 ... 146
Arbeiten mit S3-Ressourcen ... 146
Mit S3-Objekten arbeiten ... 148

AWS-Serverless-Anwendung 153
Erste Schritte 153
Arbeiten mit Serverless Land 162
Lambda-Funktionen direkt aus dem Code ausführen und debuggen 164
Ausführen und Debuggen lokaler Amazon API Gateway Ressourcen 168
Konfigurationsoptionen für das Debuggen von serverless Anwendungen 172
Fehlerbehebung 180
AWS Systems Manager 182
Annahmen und Voraussetzungen 183
IAM-Berechtigungen für Dokumente zur Automatisierung von Systems Manager 183
Ein neues Systems Manager Automation-Dokument erstellen 184
Ein vorhandenes Systems Manager Automation-Dokument öffnen 185
Bearbeiten eines Systems Manager Automation-Dokuments 185
Veröffentlichen eines Systems Manager Automation-Dokuments 186
Löschen eines Systems Manager Automation-Dokuments 187
Ausführen eines Systems Manager Automation-Dokuments 187
Fehlerbehebung 188
AWS Step Functions 189
Arbeiten mit Step Functions 189
Mit Workflow Studio arbeiten 193
Threat Composer 198
Arbeiten mit Threat Composer 198
Ressourcen 199
IAM-Berechtigungen für den Zugriff auf Ressourcen 201
Hinzufügen von Ressourcen und Interagieren mit bereits vorhandenen Ressourcen 201
Ressourcen erstellen und bearbeiten 203
Fehlerbehebung 206
Bewährte Methoden zur Fehlerbehebung 206
Profil... konnte in der Konfigurationsdatei nicht gefunden werden 207
SAM-JSON-Schema: Das Schema in der Datei template.yaml kann nicht geändert werden 208
Sicherheit 210
Datenschutz 210
Dokumentverlauf 212
..... ccxx

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 Siehttps://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-Sicherheitsscans. 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 JetBrains Links

- 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 JetBrainsBenutzerhandbuch.

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 Entwicklungsumgebung (IDE) von VS Code zur Verfügung.

Um Ihnen den Einstieg zu erleichtern, beschreiben die folgenden Themen, wie Sie den einrichten, installieren und konfigurieren AWS Toolkit for Visual Studio Code.

Themen

- Installation des AWS Toolkit for Visual Studio Code
- Verbindung herstellen zu AWS
- AWS Regionen im Wandel
- Konfiguration Ihrer Toolchain

Installation 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

Bevor Sie bestimmte Funktionen von verwenden können AWS Toolkit for Visual Studio Code, müssen Sie über Folgendes verfügen:

- 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, weiter unten beschrieben)

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

Weitere Informationen finden Sie unter Installation 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 im 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

Das Anmeldefenster öffnen

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 CodeEXPLORER.
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 NameAccess 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.

Konfiguration Ihrer Toolchain

Das AWS Toolkit for Visual Studio Code unterstützt mehrere Sprachen in allen Diensten. AWS In den folgenden Abschnitten wird beschrieben, wie Sie Ihre Toolchain für verschiedene Sprachen konfigurieren.

Konfigurieren Sie eine Toolchain für .NET Core

1. Stellen Sie sicher, dass Sie das AWS Toolkit for VS Code installiert haben.
2. Installieren Sie die C # -Erweiterung. Diese Erweiterung ermöglicht es VS Code, .NET Core-Anwendungen zu debuggen.
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 Sie eine Toolchain für Node.js

1. Stellen Sie sicher, dass Sie das AWS 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 debuggen (die Startkonfiguration hat`"target": "code"`), muss der TypeScript Compiler entweder global oder in Ihrem Projekt installiert werden. `package.json`

Eine Toolchain für Python konfigurieren

1. Stellen Sie sicher, dass Sie das AWS Toolkit for VS Code installiert haben.
2. Installieren Sie die Python-Erweiterung für Visual Studio-Code. Diese Erweiterung ermöglicht VS Code das Debuggen von Python-Anwendungen.
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`

Konfigurieren Sie eine 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 VS Code das Debuggen von Java-Anwendungen.
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 Go1.15+-Laufzeiten erforderlich.

4. Installieren Sie die Go-Tools mithilfe der Befehlspalette:
   a. Wählen Sie in der BefehlspaletteGo: Install/Update Tools.
   b. Wählen Sie aus den Kontrollkästchen `dlv` und aus`gopls`.
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 zum Ausführen oder Debuggen der Anwendung verwenden. AWS SAM

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
- Verwendung eines externen Anmeldeprozesses
- Aktualisierung von Firewalls und Gateways, um den Zugriff zu ermöglichen

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 AWS SDKs zu 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 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 und im Tools-Referenzhandbuch abschließen.AWS SDKs

> **Note**
>
> Schritt 3: Die Aktualisierung der Datei mit den gemeinsamen Anmeldeinformationen im AWS SDKs Tools-Referenzhandbuch ist optional.
> Wenn Sie Schritt 3 abgeschlossen haben, erkennt das AWS Toolkit for Visual Studio Code automatisch Ihre Anmeldeinformationen während der unten the section called “Erstellen einer Datei mit gemeinsamen Anmeldeinformationen aus dem AWS Toolkit for Visual Studio Code” aufgeführten 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`**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.
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 NameAccess 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

6. 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 es Syntaxfehler *[Profile_Name]* 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.

Verwendung eines externen Anmeldeprozesses

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

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

Aktualisierung von Firewalls und Gateways, um den Zugriff zu ermöglichen

Wenn Sie den Zugriff auf bestimmte AWS Domains oder URL-Endpunkte mithilfe einer Lösung zum Filtern von Webinhalten filtern, müssen die folgenden Endpunkte in der Liste „Zulässig“ aufgeführt sein, um auf alle Dienste und Funktionen zugreifen zu können, die über Amazon Q verfügbar sind. AWS Toolkit for Visual Studio Code

AWS Toolkit for Visual Studio Code Endpunkte

Im Folgenden finden Sie Listen mit AWS Toolkit for Visual Studio Code bestimmten Endpunkten und Referenzen, die zugelassen werden müssen.

Endpunkt

```
https://idetoolkits.amazonwebservices.com/endpoints.json
```

Gehostete Dateien

```
https://idetoolkits-hostedfiles.amazonaws.com/Notifications/VSCode/startup/1.x.json
https://idetoolkits-hostedfiles.amazonaws.com/Notifications/VSCode/emergency/1.x.json
```

Schema-Unterstützung

```
https://raw.githubusercontent.com/aws/serverless-application-model/main/samtranslator/
schema/schema.json
https://api.github.com/repos/devfile/api/releases/latest
https://raw.githubusercontent.com/devfile/api/${devfileSchemaVersion}/schemas/latest/
devfile.json
```

cSharpSamInstallationsskript debuggen

```
https://aka.ms/getvsdbgps1
https://aka.ms/getvsdbgsh
```

Endpunkte des Amazon Q-Plug-ins

Im Folgenden finden Sie eine Liste der Amazon Q-Plug-in-spezifischen Endpunkte und Referenzen, die zugelassen werden müssen.

```
https://idetoolkits-hostedfiles.amazonaws.com/*    (Plugin for configs)
https://idetoolkits.amazonwebservices.com/*   (Plugin for endpoints)
https://aws-toolkit-language-servers.amazonaws.com/*  (Language Server Process)
https://client-telemetry.us-east-1.amazonaws.com/ (Telemetry)
https://cognito-identity.us-east-1.amazonaws.com    (Telemetry)
https://aws-language-servers.us-east-1.amazonaws.com (Language Server Process)
```

Amazon Q Developer-Endpunkte

Im Folgenden finden Sie eine Liste der Amazon Q Developer-spezifischen Endpunkte und Referenzen, die zugelassen werden müssen.

```
https://codewhisperer.us-east-1.amazonaws.com (Inline,Chat, QSDA,...)
https://q.us-east-1.amazonaws.com (Inline,Chat, QSDA....)
https://desktop-release.codewhisperer.us-east-1.amazonaws.com/ (Download url for CLI.)
https://specs.q.us-east-1.amazonaws.com (Url for autocomplete specs used by CLI)
* aws-language-servers.us-east-1.amazonaws.com (Local Workspace context)
```

Endpunkte für Amazon Q Code Transform

Im Folgenden finden Sie eine Liste der spezifischen Endpunkte und Referenzen für Amazon Q Code Transform, die zugelassen werden müssen.

```
https://docs.aws.amazon.com/amazonq/latest/qdeveloper-ug/security_iam_manage-access-
with-policies.html
```

Endpunkte für die Authentifizierung

Im Folgenden finden Sie eine Liste der Authentifizierungsendpunkte und Referenzen, die zugelassen werden müssen.

```
[Directory ID or alias].awsapps.com
* oidc.[Region].amazonaws.com
*.sso.[Region].amazonaws.com
*.sso-portal.[Region].amazonaws.com
*.aws.dev
*.awsstatic.com
*.console.aws.a2z.com
*.sso.amazonaws.com
```

Identitätsendpunkte

Die folgenden Listen enthalten identitätsspezifische Endpunkte, wie z. B. die AWS IAM Identity Center AWS Builder-ID.

AWS IAM Identity Center

Einzelheiten zu den erforderlichen Endpunkten für IAM Identity Center finden Sie unter dem Thema „IAM Identity Center aktivieren“ im Benutzerhandbuch. AWS IAM Identity Center

Enterprise IAM Identity Center

```
https://[Center director id].awsapps.com/start (should be permitted to initiate auth)
https://us-east-1.signin.aws (for facilitating authentication, assuming IAM Identity
 Center is in IAD)
https://oidc.(us-east-1).amazonaws.com
https://log.sso-portal.eu-west-1.amazonaws.com.
https://portal.sso.eu-west-1.amazonaws.com
```

AWS Builder-ID

```
https://view.awsapps.com/start (must be blocked to disable individual tier)
```

```
https://codewhisperer.us-east-1.amazonaws.com and q.us-east-1.amazonaws.com (should be
 permitted)
```

Telemetrie

Im Folgenden sind telemetriespezifische Endpunkte aufgeführt, die zugelassen werden müssen.

```
https://telemetry.aws-language-servers.us-east-1.amazonaws.com/
https://client-telemetry.us-east-1.amazonaws.com
```

Referenzen

Im Folgenden finden Sie eine Liste von Endpunktreferenzen.

```
idetoolkits-hostedfiles.amazonaws.com.
cognito-identity.us-east-1.amazonaws.com.
amazonwebservices.gallery.vsassets.io.
eu-west-1.prod.pr.analytics.console.aws.a2z.com.
prod.pa.cdn.uis.awsstatic.com.
portal.sso.eu-west-1.amazonaws.com.
log.sso-portal.eu-west-1.amazonaws.com.
prod.assets.shortbread.aws.dev.
prod.tools.shortbread.aws.dev.
prod.log.shortbread.aws.dev.
a.b.cdn.console.awsstatic.com.
assets.sso-portal.eu-west-1.amazonaws.com.
oidc.eu-west-1.amazonaws.com.
aws-toolkit-language-servers.amazonaws.com.
aws-language-servers.us-east-1.amazonaws.com.
idetoolkits.amazonwebservices.com.
```

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
- AWS -Documents
- Amazon CodeCatalyst für VS Code
- Arbeiten mit Amazon API Gateway
- Verwenden AWS App Runner mit AWS Toolkit for Visual Studio Code
- AWS Anwendungsgenerator
- AWS Infrastruktur-Komponist
- AWS CDK für VS Code
- Mit AWS CloudFormation Stacks arbeiten
- Arbeiten mit CloudWatch Protokollen mithilfe der AWS Toolkit for Visual Studio Code
- Amazon DocumentDB
- Amazon Elastic Compute Cloud
- Arbeiten mit Amazon Elastic Container Registry
- Arbeiten mit Amazon Elastic Container Service
- Mit Amazon arbeiten EventBridge
- AWS IAM-Zugriffsanalysator
- Arbeiten mit AWS IoT in AWS Toolkit for Visual Studio Code
- AWS Lambda Funktionen
- Amazon Redshift im Toolkit for VS Code
- Arbeiten mit Amazon S3
- Arbeiten mit serverlosen Anwendungen
- Arbeiten mit Systems Manager Automation-Dokumenten

- AWS Step Functions
- Arbeiten mit Threat Composer
- Arbeiten mit -Ressourcen

Arbeiten mit experimentellen Funktionen

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

> **Warning**
>
> Da experimentelle Funktionen noch getestet und aktualisiert werden, kann es zu Problemen im Zusammenhang mit der Benutzerfreundlichkeit kommen. Und experimentelle Funktionen können AWS Toolkit for Visual Studio Code ohne vorherige Ankündigung aus dem entfernt werden.

Sie können experimentelle Funktionen für bestimmte AWS Dienste im AWS Toolkit-Bereich des Einstellungsbereichs in Ihrer VS Code-IDE aktivieren.

1. Um AWS Einstellungen in VS Code zu bearbeiten, wählen Sie Datei, Einstellungen, Einstellungen.
2. Erweitern Sie im Bereich Einstellungen die Erweiterung Erweiterungen und wählen Sie AWS Toolkit.
3. Wählen Sie unter AWS: Experimente die Kontrollkästchen für die experimentellen Funktionen aus, auf die Sie vor der Veröffentlichung zugreifen möchten. Wenn Sie eine experimentelle Funktion deaktivieren möchten, deaktivieren Sie das entsprechende Kontrollkästchen.

Mit AWS Diensten im AWS Explorer arbeiten

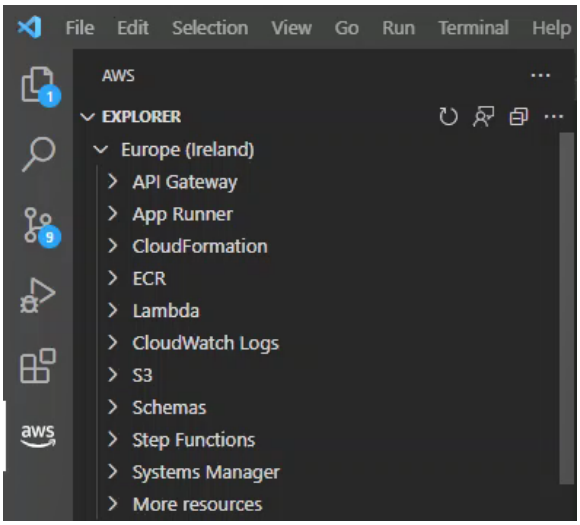
Der AWS Explorer bietet Ihnen eine Übersicht über einige AWS Dienste, mit denen Sie arbeiten können, wenn Sie den verwenden AWS Toolkit for Visual Studio Code.

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.AWS Um den AWS Explorer zu öffnen, wählen Sie AWSdas Symbol in der Aktivitätsleiste.

  Zum Beispiel:



- 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

AWS -Documents

Das AWS Toolkit for Visual Studio Code unterstützt die Option AWS Serverless Application Model `JSON Schema` für `AWS SAM templates` und verbessert die Erfahrung beim Erstellen von Vorlagen, indem Definitionen, automatische Vervollständigung und Validierung direkt in VS Code aktiviert werden. AWS Documents unterstützt alle Ressourcen AWS SAM . AWS CloudFormation Weitere Informationen finden Sie in den folgenden Ressourcen:

- Spezifische Informationen zum JSON-Schema finden Sie auf der Website JSON Schema JSON-Schema.org.
- Weitere Informationen zu AWS SAM Vorlagen finden Sie unter dem Thema Aufbau von AWS SAM Vorlagen im AWS Serverless Application ModelEntwicklerhandbuch.
- Weitere Informationen zu AWS Ressourcen und Eigenschaftstypen finden Sie im Referenzthema AWS Ressourcen- und Eigenschaftstypen im AWS CloudFormationBenutzerhandbuch.
- Ausführliche Informationen zu dem vom AWS Toolkit verwendeten AWS SAM Schema finden Sie im AWS Serverless Application ModelSchema im AWS GitHub Repository.

Erste Schritte mit Dokumenten AWS

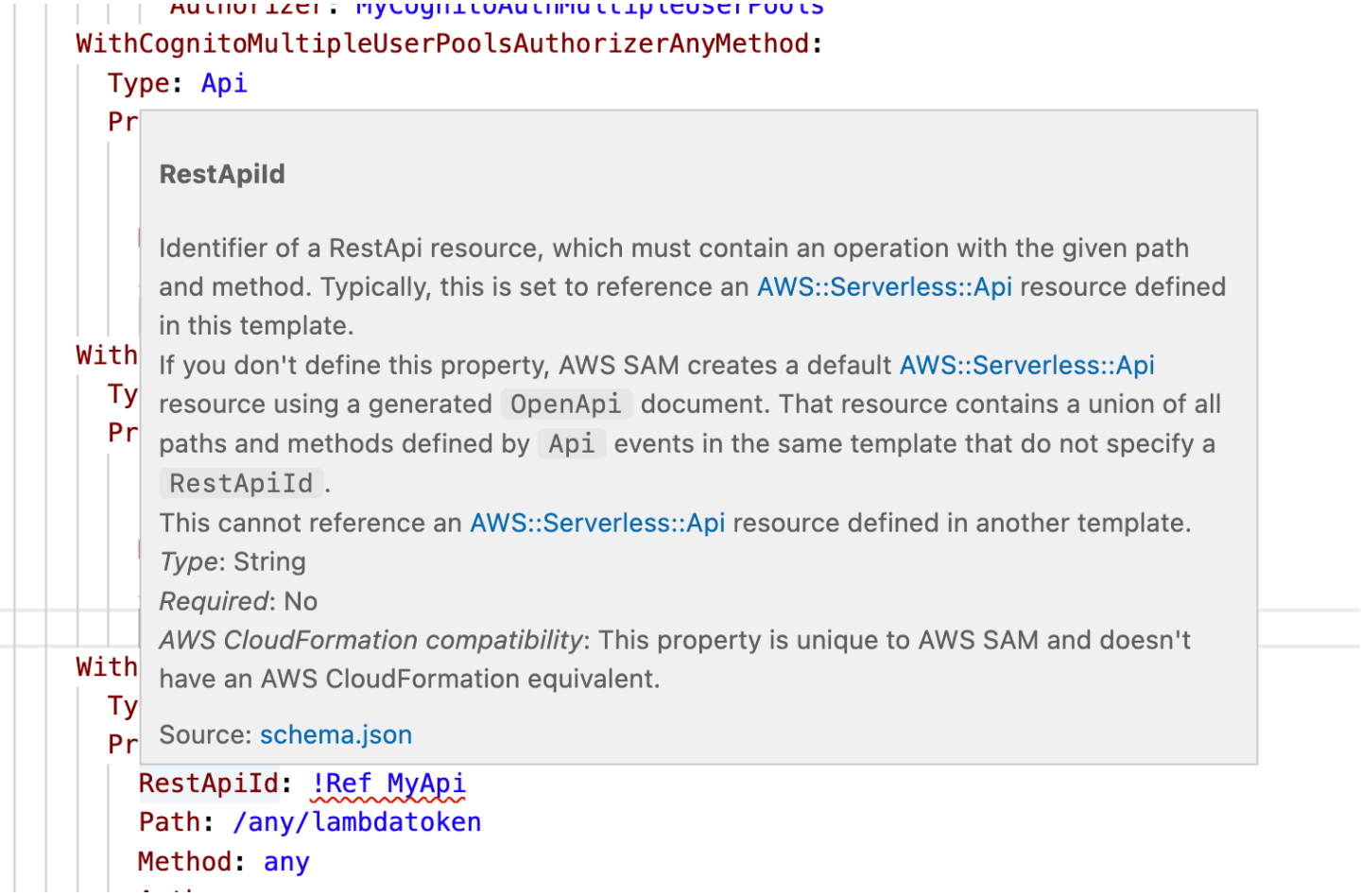
Um mit der Arbeit mit AWS Dokumenten in VS Code zu beginnen, installieren Sie die AWS Toolkit for Visual Studio Code Erweiterung von Ihrer IDE oder dem VS Code Marketplace und öffnen Sie eine beliebige AWS SAM Vorlage.

Dokumentation, Autovervollständigung und Validierung in VS Code anzeigen

Das Anzeigen von Dokumentation, automatische Vervollständigung und Validierung sind im Toolkit enthaltene Funktionen. AWS In der Abbildung unten finden Sie ein Beispiel dafür, wie diese Funktionen in VS Code aussehen.

- Um die Dokumentation aus Ihrer geöffneten AWS SAM Vorlage anzuzeigen, bewegen Sie den Mauszeiger über einen Zeileneintrag im Dokument.
- Beginnen Sie bei der automatischen Vervollständigung mit der Eingabe Ihrer AWS SAM Vorlage, um ein Pop-up mit Vorschlägen zu öffnen, die auf Ihren Eingaben basieren.
- Ihre AWS SAM Vorlage wird automatisch auf Überprüfung geprüft und Fehler werden durch ein Glühbirnensymbol hervorgehoben, das Sie für weitere Vorschläge auswählen können.

In der Abbildung unten finden Sie ein Beispiel dafür, wie diese Funktionen in VS Code aussehen.



Amazon CodeCatalyst für VS Code

Was ist Amazon CodeCatalyst?

Amazon CodeCatalyst ist ein cloudbasierter Kollaborationsraum für 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 wie Sie auf sie 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 Repositorys 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 Repositorys 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 CodeCatalystund 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 Repositorys 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 geöffnet.

1. Erweitern Sie im Toolkit for VS Code den DEVELOPER TOOLS Explorer.
2. Erweitern Sie CodeCatalystund 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 CodeCatalystSie und wählen Sie die Option Entwicklungsumgebung erstellen, um das Menü CodeCatalystEntwicklungsumgebung 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: Erzeugt 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 entsprechend Ihren spezifischen Anforderungen.
   - 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 zum Verbinden eines Drittanbieter-Repositorys mit Ihrer Entwicklungsumgebung finden Sie im Thema Verknüpfen eines Quell-Repositorys im CodeCatalystAmazon-Benutzerhandbuch.

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 arbeiten AWS 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 im 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 CodeCatalystund 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 in Ihrer `Devfile` definierten Spezifikationen entspricht.

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 CodeCatalystund 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 Entwicklungsinstanz 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 CodeCatalystund 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 damit umgeht `vscode-insiders://` URIs und nicht. `vscode://` URIs

Berechtigungen für Amazon CodeCatalyst

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

- Legen Sie Ihre eigenen Zugriffsberechtigungen für Ihre `~/.ssh/config` Datei auf `read` und fest`write`. Beschränken `write` Sie die Berechtigungen für alle anderen Benutzer.
- Legen Sie Ihre Zugriffsberechtigungen für die `~/.ssh/id_rsa` Dateien `~/.ssh/id_dsa` und auf „`read`Nur“ 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

Mit dem können Sie Remote-API-Gateway-Ressourcen in Ihrem verbundenen AWS Konto durchsuchen und ausführen AWS Toolkit for Visual Studio Code.

> **Note**
>
> Diese Funktion unterstützt kein Debugging.

Um Remote-API-Gateway-Ressourcen zu durchsuchen und auszuführen

1. Wählen Sie im AWS Explorer API Gateway, um das Menü zu erweitern. Die Remote-API-Gateway-Ressourcen werden aufgelistet.
2. Suchen Sie die API Gateway Gateway-Ressource, die Sie aufrufen möchten, öffnen Sie das zugehörige Kontextmenü (Rechtsklick) und wählen Sie dann Invoke on. AWS
3. Geben Sie im Parameterformular die Aufrufparameter an.
4. Um die Remote-API-Gateway-Ressource auszuführen, wählen Sie Invoke. Die Ergebnisse werden in der Ansicht VS Code Output angezeigt.

Verwenden AWS App Runner mit AWS Toolkit for Visual Studio Code

AWS App Runnerbietet 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 Rechendienst 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 GitHubals 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 Dienstinstanz mit der Befehlspalette von VS Code konfigurieren.

> **Note**
>
> Die mit dem AWSServiceRoleForAppRunnerDienst verknüpfte Rolle AWS App Runner ermöglicht die Ausführung der folgenden Aufgaben:
>
> - Senden Sie Protokolle an Amazon CloudWatch Logs-Protokollgruppen weiter.
> - Erstellen Sie Amazon CloudWatch Events-Regeln, um den Image-Push von Amazon Elastic Container Registry (Amazon ECR) 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 in Repositorys verwendet. https://console.aws.amazon.com/ecr/
- 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 Repositorys 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 Repositorys 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 auf verweist. GitHub AWSSie können die Verbindungen auswählen, die auf der GitHub Verbindungsseite verfügbar sind, wenn Sie Ihre Dienstinstanz 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-Dienstinstanz 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 Entwickkungsgebühr 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 GitHubals 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 Role, AppRunner ECRAccess die App Runner den Zugriff auf Amazon Elastic Container Registry (Amazon ECR) -Images in Ihrem Konto ermöglicht. 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.

AWS Anwendungsgenerator

AWS Application Builder for the AWS Toolkit for Visual Studio Code ist Ihr Leitfaden für die visuelle Erstellung von Projekten, deren lokale Bearbeitung und die Bereitstellung Ihrer Anwendungen in. AWS

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

Themen

- Mit AWS Application Builder arbeiten

Mit AWS Application Builder arbeiten

In den folgenden Abschnitten wird beschrieben, wie Sie auf AWS Application Builder in zugreifen können AWS Toolkit for Visual Studio Code. Mit Application Builder können Sie Projekte visuell erstellen, lokal iterieren und sie für bereitstellen. AWS Einen Überblick über Funktionen und mögliche Anwendungsfälle für Application Builder und Ihre AWS Lambda Erfahrungen vor Ort finden Sie im AWS YouTube Entwicklervideo *New* AWS Lambda Local IDE Experience! .

Arbeiten mit dem AWS Application Builder Explorer

Um im AWS Toolkit auf Application Builder zuzugreifen, öffnen Sie das AWS Toolkit in VS Code und erweitern Sie dann den AWS Application Builder-Explorer. Der AWS Application Builder-Explorer enthält einen Link zum Öffnen der Komplettlösung von Application Builder auf einer Registerkarte im VS Code-Editor und zeigt Ordner in Ihrem aktuellen VS Code-Arbeitsbereich an, die Ressourcen zu AWS Application Builder enthalten.

Im Application Builder-Explorer im AWS Toolkit gibt es vier project-folder-level Aktionen, auf die Sie über die Schaltflächensymbole neben Ihrem Projektordner oder durch Öffnen des Kontextmenüs für den Projektordner (Rechtsklick) für den Projektordner zugreifen können:

- Vorlagendatei öffnen: Öffnet Ihre Vorlagendatei im VS Code Explorer.
- Mit Infrastructure Composer öffnen: Öffnet Ihre Vorlagendatei mit AWS Infrastructure Composer im VS Code-Editor. Ausführliche Informationen zur Arbeit mit AWS Infrastructure Composer finden Sie unter dem Thema Was ist AWS Infrastructure Composer im AWS Infrastructure Composer Developer Guide.

- SAM-Vorlage erstellen: Öffnet das Dialogfeld „Parameter für Build angeben“ im AWS Toolkit. Sie können wählen, ob Sie Build-Flags für den Build angeben oder Standardwerte von samconfig verwenden möchten. Ausführliche Informationen zu AWS SAM Vorlagen finden Sie unter dem Thema Aufbau von Vorlagen im AWS Serverless Application ModelEntwicklerhandbuch.
- SAM-Anwendung bereitstellen: Öffnet das Dialogfeld „Bereitstellungsbefehl auswählen“ in VS Code, in dem Sie wählen können, ob Sie Ihre Anwendung bereitstellen oder synchronisieren möchten, um eine bereits bereitgestellte Anwendung zu aktualisieren. Ausführliche Informationen zur Bereitstellung von AWS SAM Anwendungen finden Sie im Thema Bereitstellen Ihrer Anwendung und Ressourcen im AWS Serverless Application ModelEntwicklerhandbuch.

Es gibt zwei Aktionen, auf die Sie über die Schaltflächensymbole neben der AWS Lambda Funktion in Ihrem Projektordner zugreifen können, oder indem Sie mit der rechten Maustaste auf die AWS Lambda Funktion klicken:

- Konfiguration für lokalen Aufruf und Debug: Öffnet das Formular für die Konfiguration von lokalem Aufrufen und Debuggen in Ihrem VS Code-Editor. Mit diesem Formular können Sie Launch-Konfigurationen des Typs erstellen, bearbeiten und ausführen:. `aws-sam` Weitere Informationen zu SAM-Debug-Konfigurationen finden Sie im Thema Konfigurationsoptionen für das Debuggen serverloser Anwendungen in diesem Benutzerhandbuch.

  > **Note**
  >
  > Derzeit wird das Debuggen einer .NET Core-Anwendung auf einer ARM64 Architektur von VS Code nicht unterstützt. Wenn Sie versuchen, eine.NET-Core-Anwendung zu debuggen, wird der folgende Fehler angezeigt:
  > `The vsdbg debugger does not currently support the arm64 architecture. Function will run locally without debug.`
  > Weitere Informationen zu diesem Problem finden Sie unter diesem VSCode-csharp-Problem im Repository. DotNet GitHub

- Funktionshandler öffnen: Öffnet Ihre Projektdatei, die den Funktionshandler enthält.

Für bereitgestellte AWS Lambda Funktionen sind 2 zusätzliche Aktionen verfügbar.

- Fernaufruf: Öffnet das Konfigurationsmenü für Fernaufruf im VS Code-Editor.
- Suchprotokolle: Öffnet das Dialogfeld „Suchprotokolle“ in VS Code.

Exemplarische Vorgehensweise für Application Builder

Die exemplarische Vorgehensweise für Application Builder ist eine step-by-step interaktive Anleitung, die Sie durch den Prozess der Erstellung einer neuen Anwendung mit AWS Application Builder führt. Sie können von zwei Stellen aus auf die exemplarische Vorgehensweise von Application Builder zugreifen: über den Application Builder-Explorer auf der Registerkarte AWS Toolkit for Visual Studio Code und über die Registerkarte VS Code Welcome. Wenn Sie im Application Builder-Explorer im AWS Toolkit die Option Exemplarische Vorgehensweise für Application Builder auswählen, wird die Komplettlösung von Application Builder auf der Registerkarte VS Code Welcome im VS Code Editor-Fenster geöffnet.

Die exemplarische Vorgehensweise von Application Builder besteht aus fünf Hauptabschnitten:

1. Installation

Im Abschnitt Installation wird überprüft, ob Sie die für Application Builder erforderlichen AWS CLI Tools und andere optionale Tools installiert haben. Wenn Sie nicht über die erforderlichen Tools verfügen oder Ihre Tools veraltet sind, werden Sie durch den Prozess der Installation der richtigen Versionen geführt.

Um zu überprüfen, ob Sie die richtigen AWS CLI und optionalen Tools installiert haben, klicken Sie auf die Schaltfläche für das AWS CLI oder ein anderes Tool, das Sie testen möchten. Nachdem Sie eine Schaltfläche ausgewählt haben, werden Ihre AWS Toolkit-Protokolle aktualisiert und VS Code zeigt eine Warnmeldung mit dem Status Ihrer Tools an. Wenn Sie Ihre Tools installieren oder aktualisieren müssen, wird die Komplettlösung von Application Builder mit den Anweisungen und Ressourcen aktualisiert, die Sie zum Fortfahren benötigen.

Ausführliche Informationen zur Installation von finden Sie im Thema Installation oder Aktualisierung auf die neueste Version des AWS CLI Themas im AWS CLIEntwicklerhandbuch. AWS CLI Ausführliche Informationen zur Installation der AWS SAM CLI finden Sie unter dem Thema AWS SAM CLI Install im AWS SAM CLI Developer Guide.

2. Wählen Sie Ihre Anwendungsvorlage

Der Abschnitt Wählen Sie Ihre Bewerbungsvorlage führt Sie durch den Prozess der Erstellung einer neuen Bewerbung anhand einer Vorlage.

Gehen Sie wie folgt vor, um eine Vorlage auszuwählen und Ihre Bewerbung zu initialisieren.

1. Wählen Sie in der exemplarischen Vorgehensweise von Application Builder den Abschnitt Wählen Sie Ihre Anwendungsvorlage aus, um eine Liste mit Vorlagenoptionen auf Ihrem Bildschirm anzuzeigen.
2. Wählen Sie eine Vorlage aus der Liste aus und klicken Sie dann auf die Schaltfläche Projekt initialisieren, um ein VS Code-Dialogfeld zu öffnen.
3. Führen Sie die Schritte im VS Code-Dialogfeld aus, um Ihre neue Anwendung zu initialisieren.
4. Die AWS Toolkit-Protokolle werden während des Initialisierungsprozesses mit dem Status Ihrer Anwendung aktualisiert.
5. Um Ihre Anwendung im Application Builder Explorer anzuzeigen, wählen Sie das Symbol Application Builder Explorer aktualisieren, um den Explorer mit Ihren Änderungen zu aktualisieren.

3. Iterieren Sie lokal

Der Abschnitt Lokal iterieren enthält Beispielbilder, die zeigen, wie Sie mit den in den VS Code- und AWS Toolkit-Explorern verfügbaren Application Builder-Funktionen iterieren können.

Weitere Informationen zu allen Application Builder-Funktionen, die in den VS Code- und AWS Toolkit-Explorern verfügbar sind, finden Sie im Abschnitt Arbeiten mit dem Application Builder-Explorer, der sich in diesem Thema im Benutzerhandbuch befindet.

4. Bereitstellen auf AWS

Der AWS Abschnitt Bereitstellen in enthält Informationen zur Konfiguration Ihrer Anmeldeinformationen AWS für die Verbindung zum Bereitstellen Ihrer Anwendung sowie Beispiele für die Bereitstellung Ihrer Anwendung mit Application Builder.

Führen Sie eines der folgenden Verfahren aus, um AWS mit Ihren vorhandenen Anmeldeinformationen aus der Anleitung zu Application Builder eine Verbindung herzustellen.

Belegschaft: Melden Sie sich AWS mit Single Sign-On an.

1. Wählen Sie in der exemplarischen Vorgehensweise von Application Builder im AWS Abschnitt Bereitstellen unter die Schaltfläche Anmeldeinformationen konfigurieren, um das Menü LOGIN im AWS Toolkit-Explorer zu öffnen.AWS
2. Wählen Sie im Menü AWS: LOGIN die Option Workforce und dann die Schaltfläche Weiter, um fortzufahren.

3. Geben Sie Ihre Start-URL in das dafür vorgesehene Feld ein, wählen Sie Ihre AWS Region aus dem Drop-down-Menü aus und klicken Sie dann auf die Schaltfläche Weiter, um fortzufahren.
4. Bestätigen Sie im Popupfenster von VS Code, dass Sie die AWS Authentifizierungsseite in Ihrem Standardbrowser öffnen möchten.
5. Führen Sie in Ihrem Standardbrowser die Authentifizierungsschritte aus. Sie werden benachrichtigt, wenn die Authentifizierung abgeschlossen ist, und Sie können Ihr Browserfenster sicher schließen.

IAM-Anmeldeinformationen: Speichern Sie Schlüssel zur Verwendung mit AWS CLI Tools.

1. Wählen Sie in der exemplarischen Vorgehensweise von Application Builder im AWS Abschnitt Bereitstellen unter die Schaltfläche Anmeldeinformationen konfigurieren, um das Menü LOGIN im AWS Toolkit-Explorer zu öffnen.AWS
2. Wählen Sie im Menü AWS: LOGIN die Option IAM-Anmeldeinformationen und anschließend die Schaltfläche Weiter aus, um fortzufahren.
3. Geben Sie einen Profilnamen in das dafür vorgesehene Feld ein, geben Sie Ihren ein **`Access Key`** und klicken Sie dann auf die Schaltfläche Weiter **`Secret Key`**, um fortzufahren.
4. VS Code zeigt den Status Ihrer Authentifizierung an und benachrichtigt Sie, wenn die Authentifizierung abgeschlossen ist oder Ihre Anmeldeinformationen ungültig sind.

Ausführliche Informationen zur Konfiguration Ihrer Anmeldeinformationen für die Bereitstellung mit dem AWS CLI finden Sie im AWS CLI Thema Konfiguration im AWS CLIEntwicklerhandbuch. Weitere Informationen zum Herstellen einer Verbindung AWS über das AWS Toolkit mit Ihren vorhandenen Anmeldeinformationen finden Sie im AWS Thema Herstellen einer Verbindung zu in diesem Benutzerhandbuch.

AWS Infrastruktur-Komponist

Sie können den verwenden AWS Toolkit for Visual Studio Code , um mit dem AWS Infrastructure Composer zu arbeiten. AWS Infrastructure 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 zu AWS Infrastructure Composer finden Sie im AWS Infrastructure Composer-Benutzerhandbuch.

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

Themen

- Arbeiten mit AWS Infrastructure Composer im Toolkit

Arbeiten mit AWS Infrastructure Composer im Toolkit

AWS Mit Infrastructure Composer for the AWS Toolkit for Visual Studio Code können Sie Anwendungen über eine interaktive Leinwand visuell entwerfen. Sie können Infrastructure Composer auch zum Visualisieren AWS CloudFormation und Ändern von AWS Serverless Application Model (AWS SAM-) Vorlagen verwenden. Während der Arbeit mit Infrastructure Composer werden Ihre Änderungen dauerhaft gespeichert, sodass Sie nahtlos zwischen der Bearbeitung von Dateien direkt im VS Code-Editor oder der interaktiven Arbeitsfläche wechseln können.

Ausführliche Informationen zu AWS Infrastructure Composer, Informationen zu den ersten Schritten und Tutorials finden Sie im AWS Infrastructure Composer-Benutzerhandbuch.

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

Zugreifen auf AWS Infrastructure Composer über das Toolkit

Es gibt drei Hauptmethoden, wie Sie über das Toolkit auf AWS Infrastructure Composer zugreifen können.

Von einer vorhandenen Vorlage aus auf AWS Infrastructure Composer zugreifen

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

Zugriff auf AWS Infrastructure Composer über das Kontextmenü (Rechtsklick)

1. Klicken Sie in VS Code mit der rechten Maustaste auf die Vorlagendatei, die Sie mit AWS Infrastructure Composer öffnen möchten.

2. Wählen Sie im Kontextmenü die Option Mit App Composer öffnen.
3. AWS Infrastructure Composer öffnet und visualisiert Ihre Vorlagendatei in einem neuen VS Code-Editor-Fenster.

Über die Befehlspalette auf AWS Infrastructure Composer zugreifen

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

AWS CDK für VS Code

> Dies ist eine Vorabveröffentlichungsdokumentation für eine Funktion in der Vorabversion. Änderungen sind vorbehalten.

Mit dem AWS CDK Explorer können Sie mit AWS Cloud Development Kit (AWS CDK)-Anwendungen oder Apps arbeiten. Ausführliche Informationen dazu finden Sie AWS CDK im AWS Cloud Development Kit (AWS CDK) Developer Guide.

AWS CDK Apps bestehen aus Bausteinen, den sogenannten Konstrukten, die Definitionen für Ihre AWS CloudFormation Stacks und die darin enthaltenen AWS Ressourcen enthalten. Mithilfe des AWS CDK Explorers können Sie die Stapel und Ressourcen visualisieren, die in Konstrukten definiert sind. AWS CDK Diese Visualisierung wird in einer Strukturansicht im Bereich Entwicklertools des Visual Studio Code (VS Code) -Editors bereitgestellt.

Dieser Abschnitt enthält Informationen zum Zugriff und zur Verwendung AWS CDK im VS Code-Editor. Es wird davon ausgegangen, dass Sie das Toolkit for VS Code bereits für Ihre lokale IDE installiert und konfiguriert haben.

Themen

- Mit AWS CDK Anwendungen arbeiten

Mit AWS CDK Anwendungen arbeiten

Dies ist eine Vorabveröffentlichungsdokumentation für eine Funktion in der Vorabversion. Änderungen sind vorbehalten.

Verwenden Sie den AWS CDK Explorer im AWS Toolkit for VS Code, um AWS CDK Anwendungen zu visualisieren und mit ihnen zu arbeiten.

Voraussetzungen

- Stellen Sie sicher, dass Ihr System die unter Installation des Toolkit for VS Code angegebenen Voraussetzungen erfüllt.
- Installieren Sie die AWS CDK Befehlszeilenschnittstelle, wie in den ersten Abschnitten von Erste Schritte mit dem AWS CDK im AWS Cloud Development Kit (AWS CDK) Entwicklerhandbuch beschrieben.

> **Important**
>
> Die 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 Sie eine Anwendung AWS CDK

Mit dem AWS Toolkit for VS Code AWS CDK Explorer können Sie die Stacks und Ressourcen verwalten, die in den CDK-Konstrukten Ihrer Apps gespeichert sind. Der AWS CDK Explorer zeigt Ihre Ressourcen anhand der in der `tree.json` Datei definierten Informationen in einer Strukturansicht an, die bei der Ausführung des Befehls erstellt wird. **`cdk synth`** Die `tree.json` Datei befindet sich standardmäßig im `cdk.out` Verzeichnis einer App.

Um mit der Verwendung des Toolkit AWS CDK Explorers zu beginnen, müssen Sie eine CDK-Anwendung erstellen.

1. Schließe die ersten Schritte des Hello World Tutorials ab, das du im AWS CDK Entwicklerhandbuch findest.

> **Important**
>
> Wenn Sie beim Lernschritt Deployment the Stack angelangt sind, halten Sie an und kehren Sie zu dieser Anleitung zurück.

> **Note**
>
> Sie können die im Tutorial bereitgestellten Befehle beispielsweise **`mkdir`**und **`cdk init`**in einer Betriebssystem-Befehlszeile oder in einem Terminalfenster im VS Code-Editor ausführen.

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. Erweitern Sie im AWS Navigationsbereich die Überschrift CDK (Vorschau). 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 Stammverzeichnis des Projekts und seinen 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, damit die Änderungen in der Strukturansicht angezeigt werden.

Führen Sie andere Operationen an einer AWS CDK App aus

Sie können den VS Code-Editor verwenden, um andere Operationen in einer CDK-App auszuführen, genauso wie Sie es mit der Befehlszeile Ihres Betriebssystems oder anderer Tools tun würden. Sie können beispielsweise die Codedateien im Editor aktualisieren und die App mithilfe eines VS Code Terminal-Fensters bereitstellen.

Um diese Art von Aktionen auszuprobieren, verwenden Sie den VS Code-Editor, um das Hello World-Tutorial im AWS CDK Entwicklerhandbuch fortzusetzen. Stellen Sie sicher, dass Sie den letzten Schritt, das Zerstören der Ressourcen der App, ausführen, damit Ihrem AWS Konto keine unerwarteten Kosten entstehen.

Mit AWS CloudFormation Stacks arbeiten

Das AWS Toolkit for Visual Studio Code bietet Unterstützung für AWS CloudFormationStapel. Mit dem Toolkit for VS Code können Sie bestimmte Aufgaben mit AWS CloudFormation Stacks ausführen, z. B. diese löschen.

Themen

- Einen AWS CloudFormation Stapel löschen
- Erstellen Sie eine AWS CloudFormation Vorlage mit dem AWS Toolkit for Visual Studio Code

Einen AWS CloudFormation Stapel löschen

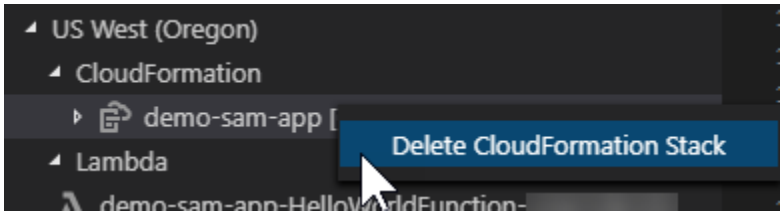
Sie können das verwenden AWS Toolkit for Visual Studio Code , um AWS CloudFormation Stapel zu löschen.

Voraussetzungen

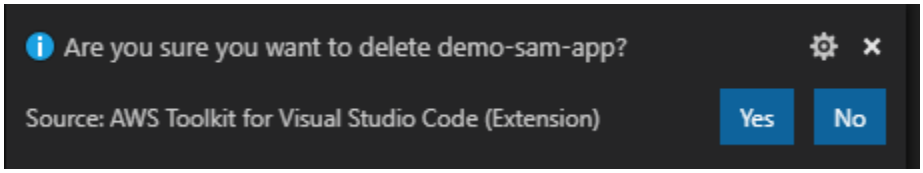
- Stellen Sie sicher, dass Ihr System die unter Installation des Toolkit for VS Code angegebenen Voraussetzungen erfüllt.
- Stellen Sie sicher, dass die von Ihnen konfigurierten Anmeldeinformationen Authentifizierung und Zugriff den entsprechenden Lese-/Schreibzugriff auf den Dienst beinhalten. AWS CloudFormation Wenn Sie im AWS Explorer unter CloudFormationeine Meldung ähnlich der Meldung „Fehler beim Laden der CloudFormation Ressourcen“ sehen, überprüfen Sie die mit diesen Anmeldeinformationen verknüpften Berechtigungen. Es dauert einige Minuten, bis sich Änderungen, die Sie an den Berechtigungen vornehmen, auf den AWS Explorer in VS Code auswirken.

Löschen Sie einen CloudFormation Stapel

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



2. Wählen Sie CloudFormation Stapel 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 Protokollen mithilfe der AWS Toolkit for Visual Studio Code

Mit Amazon CloudWatch Logs können Sie die Protokolle all Ihrer Systeme, Anwendungen und AWS Services, die Sie verwenden, in einem einzigen, hoch skalierbaren Service zentralisieren. 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 Logs? im CloudWatch Amazon-Benutzerhandbuch.

In den folgenden Themen wird beschrieben, wie Sie mit CloudWatch Logs in einem AWS Konto arbeiten können. AWS Toolkit for Visual Studio Code

Themen

- Anzeigen von CloudWatch Protokollgruppen und Protokollströmen mit dem AWS Toolkit for Visual Studio Code
- Arbeiten mit CloudWatch Protokollereignissen in Protokollstreams mithilfe der AWS Toolkit for Visual Studio Code
- CloudWatch Log-Gruppen durchsuchen
- Amazon CloudWatch protokolliert Live-Tail

Anzeigen von CloudWatch Protokollgruppen und Protokollströmen mit dem AWS Toolkit for Visual Studio Code

Ein Protokollstream ist eine Abfolge von Protokollereignissen, die dieselbe Quelle nutzen. Jede einzelne Quelle von Protokollen in 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 Protokollstreams zu einer Protokollgruppe gehören können.

Weitere Informationen finden Sie unter Working with Log Groups and Log Streams im CloudWatch Amazon-Benutzerhandbuch.

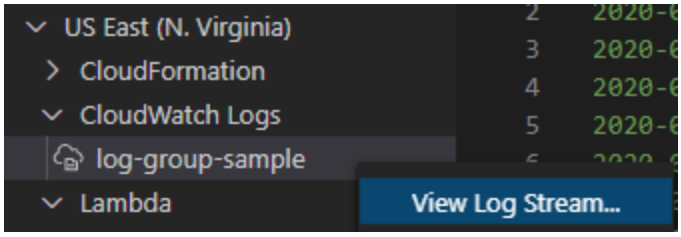
Themen

- Log-Gruppen und Log-Streams mit dem CloudWatch Logs-Knoten anzeigen

Log-Gruppen und Log-Streams mit dem CloudWatch Logs-Knoten anzeigen

1. Wählen Sie in VS Code View, Explorer aus, um den AWS Explorer zu öffnen.
2. Klicken Sie auf den Knoten CloudWatch Protokolle, um die Liste der Protokollgruppen zu erweitern.

   Die Protokollgruppen für die aktuelle AWS Region werden unter dem Knoten CloudWatch Protokolle angezeigt.
3. Um die Protokolldatenströme in einer Protokollgruppe anzuzeigen, klicken Sie mit der rechten Maustaste auf den Namen der Protokollgruppe und wählen Sie dann Protokollstream anzeigen.



4. Wählen Sie in der Befehlspalette einen Protokollstream aus der Gruppe aus, die Sie anzeigen möchten.

> **Note**
>
> In der Befehlspalette wird in jedem Stream ein Zeitstempel für das letzte Ereignis angezeigt.

   Der Log Stream-Editor wird gestartet, um die Protokollereignisse des Streams anzuzeigen.

Arbeiten mit CloudWatch Protokollereignissen in Protokollstreams mithilfe der AWS Toolkit for Visual Studio Code

Nachdem Sie den Log Steam-Editor geöffnet haben, 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, zeigt der Log Stream-Editor die Reihenfolge der Protokollereignisse dieses Streams an.

1. Um einen Log-Stream zum Anzeigen zu finden, öffnen Sie den Log Stream-Editor (siehe CloudWatch Protokollgruppen und Protokolldatenströme anzeigen).

   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 auswählen.

     > **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 und Kopieren aus dem Menü.
- Den Namen des Log-Streams kopieren: Klicken Sie mit der rechten Maustaste auf die Registerkarte des Log Stream-Editors und wählen Sie Log-Stream-Name kopieren.

> **Note**
>
> Sie können auch die Befehlspalette verwenden, um das AWS Toolkit Copy Log Stream Name auszuführen.

Speichern des Inhalts des Protokolldatenstam-Editors in einer lokalen Datei

Sie können den Inhalt des CloudWatch Logstream-Editors in eine `log` Datei auf Ihrem lokalen Computer herunterladen.

> **Note**
>
> Mit dieser Option speichern Sie nur die Protokollereignisse, die derzeit im Logstream-Editor angezeigt werden, in einer Datei. 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. Um einen Logstream zum Kopieren zu finden, öffnen Sie den Log Streams-Editor (siehe CloudWatch Protokollgruppen und Protokolldatenströme anzeigen).
2. Wählen Sie das Symbol Speichern neben der Registerkarte, auf der der Name des Log-Streams angezeigt wird.

   > **Note**
   >
   > Sie können auch die Befehlspalette verwenden, um das AWS Toolkit Save Current Log Stream Content auszuführen.

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 Protokollströmen 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 abzubrechen.
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 abzubrechen.
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.

Amazon CloudWatch protokolliert Live-Tail

Streamen Sie mit Amazon CloudWatch CloudWatch Logs Live Tail Ihre Protokollereignisse live, während sie in eine bestimmte Protokollgruppe aufgenommen werden.

Ausführliche Informationen zur Live Tail-Funktion finden Sie unter dem Thema Problembehandlung mit CloudWatch Logs Live Tail im Amazon CloudWatch Logs-Benutzerhandbuch.

Bei Live Tail-Sitzungen fallen Kosten pro Minute an. Preisinformationen finden Sie auf der Registerkarte Protokolle im Abschnitt Bezahltes Kontingent des CloudWatch Amazon-Preisleitfadens.

Starten einer Live-Tail-Sitzung über die VS Code-Befehlspalette

Gehen Sie wie folgt vor, um eine Live Tail-Sitzung von der VS Code Command Palette aus zu starten.

Ausführliche Informationen zu den Filtern und Mustern von Amazon CloudWatch Logs finden Sie im Abschnitt Filter- und Mustersyntax im CloudWatchAmazon-Benutzerhandbuch.

Starten einer Tailing-Sitzung über die Befehlspalette

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: Tail Log Group`, um das Dialogfeld „Tail log group“ in VS Code zu öffnen, und folgen Sie den Anweisungen, um fortzufahren.

   > **Note**
   >
   > Bei der ersten Aufforderung haben Sie die Möglichkeit, Ihre AWS Region zu wechseln, bevor Sie mit den nächsten Schritten fortfahren.

3. Wählen Sie in der Eingabeaufforderung Tail Log Group (1/3) die Protokollgruppe aus, die Sie verfolgen möchten.
4. Wählen Sie in der Eingabeaufforderung Include log events from... (2/3) den Log-Stream-Filter aus, der auf Ihre Protokollsitzung angewendet werden soll.
5. Geben Sie an der Eingabeaufforderung Protokollereignisfiltermuster angeben... (3/3) Ihr Filtermuster in das dafür vorgesehene Feld ein und drücken Sie dann die **`Enter`** Eingabetaste, um fortzufahren, oder die **`ESC`** Taste, um die Suche abzubrechen.
6. Nach Abschluss werden Ihre Ergebnisse in den VS Code-Editor gestreamt

   > **Note**
   >
   > Wenn eine Live-Tail-Sitzung, die im VS Code-Fenster ausgeführt wird, mit der Konfiguration eines neu übermittelten Tail Log Group-Befehls übereinstimmt, wird keine neue Sitzung gestartet. Stattdessen wird Ihre bestehende Sitzung zum aktiven Texteditor.

Starten Sie eine Live-Tail-Sitzung vom AWS Explorer aus

Gehen Sie wie folgt vor, um eine Live-Tail-Sitzung vom AWS Toolkit Explorer aus zu starten.

Starten Sie eine Tailing-Sitzung vom Explorer aus AWS

1. Erweitern Sie im AWS Toolkit-Explorer. CloudWatch
2. Öffnen Sie das Kontextmenü für die Protokollgruppe (klicken Sie mit der rechten Maustaste darauf), und wählen Sie dann Tail Log Group, um die Eingabeaufforderung für das Tailing zu öffnen.
3. Folgen Sie den Anweisungen, um fortzufahren.
4. Ihre Ergebnisse werden in den VS Code Editor gestreamt.

Eine Live-Tail-Sitzung beenden

Es gibt zwei Möglichkeiten, eine laufende Tailing-Sitzung zu beenden.

Eine Tailing-Sitzung beenden

1. Klicken Sie unten CodeLens im Textdokument mit der Aufzeichnung der Sitzung auf Beenden.
2. Schließen Sie alle Editoren, die das Sitzungs-Textdokument enthalten.

Amazon DocumentDB

Sie können Ihre Amazon DocumentDB-Cluster und -Instances direkt in VS Code mit dem AWS Toolkit for Visual Studio Code verwalten. Amazon DocumentDB (mit MongoDB-Kompatibilität) ist ein schneller, zuverlässiger und vollständig verwalteter Datenbankservice, der die Einrichtung, den Betrieb und die Skalierung von MongoDB-kompatiblen Datenbanken in der Cloud vereinfacht. Ausführliche Informationen zum Amazon DocumentDB-Service finden Sie im Amazon DocumentDB Developer Guide.

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

Themen

- Arbeiten mit Amazon DocumentDB im Toolkit

Arbeiten mit Amazon DocumentDB im Toolkit

Amazon DocumentDB (mit MongoDB-Kompatibilität) ist ein schneller, zuverlässiger und vollständig verwalteter Datenbankservice, der die Einrichtung, den Betrieb und die Skalierung von MongoDB-kompatiblen Datenbanken in der Cloud vereinfacht.

Ausführliche Informationen zu Amazon DocumentDB, Informationen zu den ersten Schritten und Tutorials finden Sie im Amazon DocumentDB Developer Guide.

In den folgenden Abschnitten wird beschrieben, wie Sie mit Amazon DocumentDB mit dem AWS Toolkit for Visual Studio Code arbeiten.

Zugreifen auf Amazon DocumentDB über das Toolkit AWS

Gehen Sie wie folgt vor, um mit dem AWS Toolkit auf Amazon DocumentDB zuzugreifen.

Zugreifen auf Amazon DocumentDB im Toolkit AWS

1. Öffnen Sie in VS Code. AWS Toolkit for Visual Studio Code
2. Erweitern Sie im AWS Toolkit den Explorer.
3. Erweitern Sie im Explorer Amazon DocumentDB, um Ihre vorhandenen Amazon DocumentDB DocumentDB-Ressourcen anzuzeigen.

Einen instanzbasierten Cluster erstellen.

Um mit der Arbeit mit Amazon DocumentDB zu beginnen, erstellen Sie einen Cluster, indem Sie das folgende Verfahren ausführen.

Einen instanzbasierten Cluster erstellen

1. Öffnen Sie im das AWS Toolkit for Visual Studio Code Kontextmenü für Amazon DocumentDB (klicken Sie mit der rechten Maustaste darauf) und wählen Sie dann Cluster erstellen aus, um das Dialogfeld Amazon DocumentDB-Cluster erstellen in VS Code zu öffnen.
2. Wählen Sie auf dem Bildschirm Clustertyp die Option Instance Based Cluster aus.
3. Geben Sie auf dem Bildschirm Clustername einen Namen für Ihren neuen Cluster ein.
4. Wählen Sie auf dem Bildschirm Engine-Version auswählen Ihre bevorzugte Amazon DocumentDB DocumentDB-Engine-Version aus.

5. Geben Sie auf den Bildschirmen Admin-Benutzername und Passwort einen Admin-Benutzernamen und ein Admin-Passwort an, um Ihren Cluster zu schützen.
6. Wählen Sie auf dem Bildschirm „Speicherverschlüsselung angeben“ aus, ob Ihr Cluster verschlüsselt werden soll oder nicht.
7. Konfigurieren Sie auf dem Bildschirm „Anzahl der Instanzen“ Ihre bevorzugte Anzahl von Instanzen.
8. Wählen Sie auf dem Bildschirm „Instance-Klasse auswählen“ Ihre bevorzugte Instance-Klasse aus und fahren Sie dann mit der Erstellung Ihres neuen Clusters fort.

> **Note**
>
> Das Hochfahren Ihres Clusters kann mehrere Minuten dauern.

Einen Cluster-Endpunkt kopieren

Gehen Sie wie folgt vor, um Ihren Amazon DocumentDB-Cluster-Endpunkt zu kopieren.

Einen Cluster-Endpunkt kopieren

1. Erweitern Sie von der aus Amazon DocumentDB AWS Toolkit for Visual Studio Code, um Ihre Amazon DocumentDB-Cluster anzuzeigen.
2. Klicken Sie mit der rechten Maustaste auf den Cluster, aus dem Sie die Verbindungsdetails kopieren möchten, und wählen Sie dann Endpunkt kopieren, um die Cluster-Endpunktinformationen in Ihre Zwischenablage zu kopieren.
3. Ihr Cluster-Endpunkt kann jetzt in Ihre Dokumente eingefügt werden.

Im Browser öffnen

Öffnen Sie Ihre Amazon DocumentDB-Cluster in der AWS Konsole, um weitere Funktionen zur Clusterverwaltung zu erhalten. Gehen Sie wie folgt vor, um die AWS Konsole für Ihren Amazon DocumentDB-Cluster in Ihrem Standard-Webbrowser zu öffnen.

Öffnen Sie Ihren Cluster in der Konsole AWS

1. Erweitern Sie von der aus Amazon DocumentDB AWS Toolkit for Visual Studio Code, um Ihre Amazon DocumentDB-Cluster anzuzeigen.

2. Klicken Sie mit der rechten Maustaste auf den Cluster, den Sie in der AWS Konsole anzeigen möchten, und wählen Sie dann Im Browser öffnen.
3. Die AWS Konsole wird im Amazon DocumentDB-Cluster in Ihrem Standard-Webbrowser geöffnet.

Erweiterung eines vorhandenen Clusters

Gehen Sie wie folgt vor, um Ihre Amazon DocumentDB-Cluster durch Hinzufügen von Instances zu skalieren.

Hinzufügen einer Instance zur Erweiterung Ihres Clusters

1. Erweitern Sie von der aus Amazon DocumentDB AWS Toolkit for Visual Studio Code, um Ihre Amazon DocumentDB-Cluster anzuzeigen.
2. Klicken Sie mit der rechten Maustaste auf den Cluster, den Sie erweitern möchten, und wählen Sie Instance hinzufügen, um das Dialogfeld Instance hinzufügen in VS Code zu öffnen.
3. Wenn Sie dazu aufgefordert werden, geben Sie einen Namen für Ihre neue Instanz in das Textfeld ein und drücken Sie dann die **`Enter`** Eingabetaste, um fortzufahren.
4. Wenn Sie dazu aufgefordert werden, wählen Sie eine Instanzklasse aus der Liste aus, um fortzufahren.
5. Der AWS Explorer zeigt den Erstellungsstatus an und aktualisiert, wenn die neue Instanz bereit ist.

Stoppen eines Clusters

Stoppen Sie Ihren Amazon DocumentDB-Cluster, indem Sie das folgende Verfahren ausführen.

> **Note**
>
> Solange Ihr Cluster gestoppt ist, sind die meisten Clusterverwaltungsfunktionen nicht verfügbar.

Ihren Amazon DocumentDB-Cluster beenden

1. Erweitern Sie von der aus Amazon DocumentDB AWS Toolkit for Visual Studio Code, um Ihre Amazon DocumentDB-Cluster anzuzeigen.

2. Wählen Sie die Schaltfläche „Cluster beenden“ neben dem Cluster, den Sie beenden möchten, oder klicken Sie mit der rechten Maustaste auf den Cluster und wählen Sie Cluster beenden.
3. Wenn Sie dazu aufgefordert werden, wählen Sie Ja, um Ihren Cluster zu beenden, oder Abbrechen, um den Stoppvorgang abzubrechen und den Cluster weiterlaufen zu lassen.
4. Der AWS Explorer zeigt den Status Ihres Clusters an und aktualisiert, wenn der Cluster gestoppt wurde.

Eine Instanz neu starten

Der Neustart einer Instanz ist nützlich, um Fehler zu beheben und kleinere Änderungen vorzunehmen, ohne dass sich dies auf Ihren gesamten Cluster auswirkt. Gehen Sie wie folgt vor, um eine Amazon DocumentDB DocumentDB-Instance neu zu starten.

Neustarten einer Cluster-Instance

1. Erweitern Sie von der aus Amazon DocumentDB AWS Toolkit for Visual Studio Code, um Ihre Amazon DocumentDB-Cluster anzuzeigen.
2. Klicken Sie mit der rechten Maustaste auf die Cluster-Instance, die Sie neu starten möchten, und wählen Sie dann Reboot Instance.
3. Wenn Sie dazu aufgefordert werden, wählen Sie Ja, um Ihre Instance neu zu starten, oder Abbrechen, um den Neustartvorgang abzubrechen und Ihre Instance gestoppt zu lassen.
4. Der AWS Explorer zeigt den Status Ihres Clusters an und aktualisiert, wenn Ihre Instance neu gestartet wurde.

Eine Instance löschen

Gehen Sie wie folgt vor, um eine Amazon DocumentDB-Cluster-Instance zu löschen.

> **Note**
>
> Das Löschen einer Instance hat keine Auswirkungen auf die Daten in Ihrem Cluster. Wenn Sie die primäre Instanz löschen, übernimmt eine der Replikatinstanzen die Rolle der beschreibbaren Instanz.

Löschen einer Clusterinstanz

1. Erweitern Sie von der aus Amazon DocumentDB AWS Toolkit for Visual Studio Code, um Ihre Amazon DocumentDB-Cluster anzuzeigen.
2. Klicken Sie mit der rechten Maustaste auf die Cluster-Instance, die Sie löschen möchten, und wählen Sie dann Löschen, um das delete-cluster-instance Bestätigungsdialogfeld in VS Code zu öffnen.
3. Folgen Sie der Bestätigungsaufforderung und drücken Sie dann die **`Enter`** Taste, um Ihre Clusterinstanz zu löschen.
4. Der AWS Explorer zeigt den Status Ihrer Cluster-Instance an und aktualisiert, wenn Ihre Instance gelöscht wurde.

Tags anzeigen, hinzufügen und entfernen

Tags werden verwendet, um Ressourcen in Ihrer Umgebung zu organisieren und nachzuverfolgen. Um die mit Ihrem Amazon DocumentDB-Cluster verknüpften Tags anzuzeigen oder zu bearbeiten, führen Sie eines der folgenden Verfahren aus.

Cluster-Tags anzeigen

1. Erweitern Sie von der aus Amazon DocumentDB AWS Toolkit for Visual Studio Code, um Ihre Amazon DocumentDB-Cluster anzuzeigen.
2. Klicken Sie mit der rechten Maustaste auf den Cluster, für den Sie Tags anzeigen möchten, und wählen Sie dann Tags... um den **`your cluster name`** Dialog „Tags für“ zu öffnen.
3. Ihre Tags werden im Dialogfenster angezeigt. Wenn Ihrem Cluster keine Tags zugeordnet sind, wird die Meldung [Keine Tags zugewiesen] angezeigt.

Hinzufügen von Tags zu Ihrem Cluster

1. Erweitern Sie von der aus Amazon DocumentDB AWS Toolkit for Visual Studio Code, um Ihre Amazon DocumentDB-Cluster anzuzeigen.
2. Klicken Sie mit der rechten Maustaste auf den Cluster, für den Sie Tags hinzufügen möchten, und wählen Sie dann Tags... um den **`your cluster name`** Dialog „Tags für“ zu öffnen.
3. Wählen Sie Tag hinzufügen... Schaltfläche, um das Dialogfeld „Tag hinzufügen“ in VS Code zu öffnen.

4. Geben Sie ein neues Tag in das Textfeld ein und drücken Sie dann die Eingabetaste, um fortzufahren.
5. Geben Sie einen Wert in das Textfeld ein und drücken Sie dann die Eingabetaste, um das Schlüssel/Wert-Paar zu Ihrem Cluster hinzuzufügen.

Tags aus Ihrem Cluster entfernen

1. Erweitern Sie von der aus Amazon DocumentDB AWS Toolkit for Visual Studio Code, um Ihre Amazon DocumentDB-Cluster anzuzeigen.
2. Klicken Sie mit der rechten Maustaste auf den Cluster, aus dem Sie Tags entfernen möchten, und wählen Sie dann Tags... um den **`your cluster name`** Dialog „Tags für“ zu öffnen.
3. Wählen Sie das Tag entfernen... Schaltfläche, um das **`your cluster name`** Dialogfeld „Tag aus“ entfernen in VS Code zu öffnen.
4. Wählen Sie das Tag, das Sie entfernen möchten, aus der bereitgestellten Liste aus, um das Tag aus Ihrem Cluster zu entfernen.

Eine Instance-Klasse ändern

Gehen Sie wie folgt vor, um die Klasse einer Amazon DocumentDB-Cluster-Instance zu ändern.

Ändern einer Instance-Klasse

1. Erweitern Sie von der aus Amazon DocumentDB AWS Toolkit for Visual Studio Code, um Ihre Amazon DocumentDB-Cluster anzuzeigen.
2. Klicken Sie mit der rechten Maustaste auf die Cluster-Instance, die Sie ändern möchten, und wählen Sie dann Modify Class... um das Dialogfeld „Instanzklasse auswählen“ in VS Code zu öffnen.
3. Wählen Sie eine neue Klasse für Ihre Instanz aus der Liste aus, um die Klasse zu aktualisieren.
4. Der AWS Explorer zeigt den Status Ihrer Cluster-Instance an und aktualisiert, wenn die Klasse Ihrer Instance aktualisiert wurde.

Amazon Elastic Compute Cloud

Mit Amazon Elastic Compute Cloud for the AWS Toolkit for Visual Studio Code können Sie Ihre EC2 Amazon-Instances von VS Code aus starten und eine Verbindung zu ihnen herstellen. Ausführliche

Informationen zu Amazon EC2 finden Sie unter Was ist Amazon EC2? Thema im Amazon Elastic Compute Cloud-Benutzerhandbuch.

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

Themen

- Arbeiten mit Amazon Elastic Compute Cloud
- Fehlerbehebung bei Amazon Elastic Compute Cloud

Arbeiten mit Amazon Elastic Compute Cloud

In den folgenden Abschnitten wird beschrieben, wie Sie mit Amazon Elastic Compute Cloud in der arbeiten AWS Toolkit for Visual Studio Code.

Voraussetzungen

Die in diesem Thema im Benutzerhandbuch beschriebenen Funktionen wurden auf EC2 Amazon-Instances mit den folgenden Betriebssystemen getestet:

- Windows 2016+

> **Note**
>
> Dieses Betriebssystem funktioniert nur, wenn ein VS Code-Terminal angeschlossen wird. Es funktioniert nicht, wenn eine vollständige VS Code-Remote-Instanz verbunden wird. Weitere Informationen zu VS Code-Terminals und Remote-Instanzen finden Sie in den Themen Erste Schritte mit dem Terminal und VS Code Remote Development in der VS Code-Dokumentation.

- Amazon Linux 2023
- Ubuntu, 22.04

Ein lokal installiertes SSH ist erforderlich, um eine Remote-Verbindung zu einer EC2 Amazon-Instance herzustellen, ist jedoch nicht erforderlich, um ein Terminal für eine EC2 Amazon-Instance zu öffnen.

Ihr EC2 Amazon-Instance-Profil muss die folgenden AWS Identity and Access Management (IAM-) Berechtigungen enthalten.

```
"ssmmessages:CreateControlChannel",
"ssmmessages:CreateDataChannel",
"ssmmessages:OpenControlChannel",
"ssmmessages:OpenDataChannel",
"ssm:DescribeAssociation",
"ssm:ListAssociations",
"ssm:UpdateInstanceInformation
```

> **Note**
>
> Die erforderlichen Berechtigungen sind in der folgenden AWS verwalteten Richtlinie enthalten.
>
> - `AmazonSSMManagedInstanceCore`
> - `AmazonSSMManagedEC2InstanceDefaultPolicy`

Bestehende EC2 Amazon-Instances anzeigen

Gehen Sie wie folgt vor, um Ihre vorhandenen EC2 Amazon-Instances im AWS Toolkit anzuzeigen.

1. Erweitern Sie im AWS Toolkit den AWS Toolkit-Explorer.
2. Erweitern Sie die Region, die die EC2 Amazon-Instances enthält, die Sie anzeigen möchten.
3. Erweitern Sie die EC2Überschrift, um Ihre vorhandenen EC2 Amazon-Instances anzuzeigen.

Eine neue EC2 Amazon-Instance starten

Es gibt 3 Möglichkeiten, eine neue EC2 Amazon-Instance mit dem AWS Toolkit zu erstellen.

Jeder Arbeitsablauf öffnet den Assistenten zum Starten einer Instance in der AWS Konsole. Ausführliche Informationen zum Starten einer neuen EC2 Amazon-Instance über den Launch an Instance Wizard finden Sie im Thema Launch an EC2 instance using the launch instance wizard in the console im Amazon Elastic Compute Cloud-Benutzerhandbuch. Um eine neue EC2 Amazon-Instance zu starten, führen Sie eines der folgenden Verfahren aus.

Starten einer neuen EC2 Amazon-Instance über die VS Code Command Palette

1. Öffnen Sie in VS Code die VS Code-Befehlspalette, indem Sie **`command + shift + P (Windows: ctrl + shift + P)`**
2. Suchen Sie in der VS Code-Befehlspalette nach dem **`AWS: Launch EC2`** Befehl und wählen Sie ihn aus, wenn er in der Liste angezeigt wird, um die Aufforderung „ EC2 Startinstanz auswählen“ in VS Code zu öffnen.
3. Wählen Sie in der Eingabeaufforderung „ EC2 Instance starten — Region auswählen“ die Region aus, in der Sie Ihre neue Instance starten möchten, und bestätigen Sie dann, dass Sie die AWS Konsole in Ihrem Standard-Webbrowser öffnen möchten.
4. Schließen Sie in der AWS Konsole Ihres Standard-Webbrowsers den Authentifizierungsprozess ab, um mit dem Assistenten zum Starten einer Instance fortzufahren.
5. Füllen Sie im Assistenten zum Starten einer Instance die erforderlichen Abschnitte aus und wählen Sie dann die Schaltfläche Instance starten, um Ihre neue EC2 Amazon-Instance zu starten.
6. Der AWS Explorer wird aktualisiert und zeigt Ihre neue EC2 Amazon-Instance an.

Eine neue EC2 Amazon-Instance über den AWS Explorer starten

1. Erweitern Sie den AWS Toolkit-Explorer und dann die Region, in der Sie die neue EC2 Amazon-Instance erstellen möchten.
2. Erweitern Sie die EC2Überschrift oder bewegen Sie den Mauszeiger darüber und wählen Sie dann das Symbol + (Launch EC2 Instance).
3. Wenn Sie dazu aufgefordert werden, bestätigen Sie, dass Sie die AWS Konsole in Ihrem Standard-Webbrowser öffnen möchten.
4. Schließen Sie in der AWS Konsole Ihres Webbrowsers den Authentifizierungsprozess ab, um mit dem Assistenten zum Starten einer Instanz fortzufahren.
5. Füllen Sie im Assistenten zum Starten einer Instance die erforderlichen Abschnitte aus und wählen Sie dann die Schaltfläche Instance starten, um Ihre neue EC2 Amazon-Instance zu starten.
6. Der AWS Explorer wird aktualisiert und zeigt Ihre neue EC2 Amazon-Instance an.

Starten einer neuen EC2 Amazon-Instance über das Kontextmenü (Rechtsklick)

1. Erweitern Sie den AWS Toolkit-Explorer und dann die Region, in der Sie die neue EC2 Amazon-Instance erstellen möchten.
2. Klicken Sie mit der rechten Maustaste EC2auf die Überschrift und wählen Sie dann Launch EC2 Instance.
3. Wenn Sie dazu aufgefordert werden, bestätigen Sie, dass Sie die AWS Konsole in Ihrem Standard-Webbrowser öffnen möchten.
4. Schließen Sie in der AWS Konsole Ihres Webbrowsers den Authentifizierungsprozess ab, um mit dem Assistenten zum Starten einer Instanz fortzufahren.
5. Füllen Sie im Assistenten zum Starten einer Instance die erforderlichen Abschnitte aus und wählen Sie dann die Schaltfläche Instance starten, um Ihre neue EC2 Amazon-Instance zu starten.
6. Der AWS Explorer wird aktualisiert und zeigt Ihre neue EC2 Amazon-Instance an.

VS Code mit einer EC2 Amazon-Instance verbinden

Es gibt drei Möglichkeiten, von VS Code aus eine Verbindung zu einer EC2 Amazon-Instance herzustellen. Um VS Code mit Ihrer EC2 Instance zu verbinden, führen Sie eines der folgenden Verfahren aus.

VS Code über die Befehlspalette mit einer EC2 Amazon-Instance verbinden

1. Öffnen Sie in VS Code die VS Code-Befehlspalette, indem Sie auf **`command + shift + P (Windows: ctrl + shift + P)`**
2. Suchen Sie in der VS Code-Befehlspalette nach dem **`AWS: Connect VS Code to EC2 instance...`** Befehl und wählen Sie ihn aus, wenn er in der Liste angezeigt wird, um die Eingabeaufforderung „ EC2Instanz auswählen“ in VS Code zu öffnen.
3. Wählen Sie in der Eingabeaufforderung „ EC2 Instanz auswählen“ die Region aus, in der sich die Instanz befindet, zu der Sie eine Verbindung herstellen möchten, und wählen Sie dann die Instanz aus, zu der Sie eine Verbindung herstellen möchten.
4. VS Code zeigt den Status an, während die Verbindung hergestellt wird.
5. Ein neues Fenster wird geöffnet, in dem Ihre EC2 Amazon-Instance angezeigt wird, wenn die Verbindung hergestellt ist.

VS Code über den AWS Explorer mit einer EC2 Amazon-Instance verbinden.

1. Erweitern Sie den AWS Toolkit-Explorer und dann die Region, die die EC2 Amazon-Instance enthält, zu der Sie eine Verbindung herstellen möchten.
2. Bewegen Sie den Mauszeiger über die EC2 Amazon-Instance und wählen Sie dann das Symbol (VS-Code mit EC2 Instance Connect).

   > **Note**
   >
   > Sie können auch das Symbol (VS Code mit EC2 Instanz Connect) aus der EC2Dienstüberschrift im AWS Explorer auswählen.

3. VS Code zeigt den Status an, während die Verbindung hergestellt wird.
4. Ein neues Fenster wird geöffnet, in dem Ihre EC2 Amazon-Instance angezeigt wird, wenn die Verbindung hergestellt ist.

VS Code über das Rechtsklick-Menü mit einer EC2 Amazon-Instance verbinden

1. Erweitern Sie den AWS Toolkit-Explorer und dann die Region, die die EC2 Amazon-Instance enthält, zu der Sie eine Verbindung herstellen möchten.
2. Klicken Sie mit der rechten Maustaste auf die EC2 Amazon-Instance, zu der Sie eine Verbindung herstellen möchten, und wählen Sie dann Connect VS Code to EC2 Instance.

   > **Note**
   >
   > Sie können auch mit der rechten Maustaste auf die EC2Dienstüberschrift im AWS Explorer klicken und die EC2Instanz Connect VS Code to auswählen.

3. VS Code zeigt den Status an, während die Verbindung hergestellt wird.
4. Ein neues Fenster wird geöffnet, in dem Ihre EC2 Amazon-Instance angezeigt wird, wenn die Verbindung hergestellt ist.

Öffnen eines Terminals für eine EC2 Amazon-Instance.

Es gibt drei Möglichkeiten, vom VS Code-Terminal aus eine Verbindung zu einer EC2 Amazon-Instance herzustellen.

VS Code über die Befehlspalette mit einer EC2 Amazon-Instance verbinden

1. Öffnen Sie in VS Code die VS Code-Befehlspalette, indem Sie auf **`command + shift + P (Windows: ctrl + shift + P)`**
2. Suchen Sie in der VS Code-Befehlspalette nach dem **`AWS:Open terminal to EC2 instance...`** Befehl und wählen Sie ihn aus, wenn er in der Liste angezeigt wird, um die Eingabeaufforderung „ EC2Instanz auswählen“ in VS Code zu öffnen.
3. Wählen Sie in der Eingabeaufforderung „ EC2 Instanz auswählen“ die Region aus, die die Instanz enthält, die Sie im Terminal öffnen möchten, und wählen Sie dann die Instanz aus.
4. VS Code zeigt den Status an, während die Verbindung hergestellt wird.
5. Das VS Code Terminal wird geöffnet, um Ihre neue Sitzung anzuzeigen, wenn die Verbindung hergestellt ist.

Öffnen einer EC2 Amazon-Instance im VS Code-Terminal vom AWS Explorer aus.

1. Erweitern Sie den AWS Toolkit-Explorer und dann die Region, die die EC2 Amazon-Instance enthält, zu der Sie eine Verbindung herstellen möchten.
2. Bewegen Sie den Mauszeiger über die EC2 Amazon-Instance und wählen Sie dann (Terminal für EC2 Instance öffnen...) Symbol.

   > **Note**
   >
   > Sie können auch das (Terminal zur EC2 Instanz öffnen...) wählen Symbol aus der EC2Service-Überschrift im AWS Explorer.

3. VS Code zeigt den Status an, während die Verbindung hergestellt wird.
4. Das VS Code Terminal wird geöffnet, um Ihre neue Sitzung anzuzeigen, wenn die Verbindung hergestellt ist.

Öffnen einer EC2 Amazon-Instance im VS Code-Terminal über das Rechtsklick-Menü

1. Erweitern Sie den AWS Toolkit-Explorer und dann die Region, die die EC2 Amazon-Instance enthält, die Sie im VS Code-Terminal öffnen möchten.
2. Klicken Sie mit der rechten Maustaste auf die EC2 Amazon-Instance, die Sie im Terminal öffnen möchten, und wählen Sie dann Terminal für EC2 Instance öffnen... .

> **Note**
>
> Sie können auch mit der rechten Maustaste auf die EC2Dienstüberschrift im AWS Explorer klicken und die Option Terminal zur EC2 Instanz öffnen auswählen... .

3. VS Code zeigt den Status an, während die Verbindung hergestellt wird.
4. Das VS Code Terminal wird geöffnet, um Ihre neue Sitzung anzuzeigen, wenn die Verbindung hergestellt ist.

Eine Amazon-Instance starten oder neu starten EC2

Es gibt drei Möglichkeiten, eine EC2 Amazon-Instance zu starten oder neu zu starten.

Eine EC2 Amazon-Instance über die Befehlspalette neu starten

1. Öffnen Sie in VS Code die VS Code-Befehlspalette, indem Sie auf **`command + shift + P (Windows: ctrl + shift + P)`**
2. Suchen Sie in der VS Code-Befehlspalette nach dem **`AWS: Reboot EC2 instance`** Befehl und wählen Sie ihn aus, wenn er in der Liste angezeigt wird, um die Eingabeaufforderung „ EC2Instanz auswählen“ in VS Code zu öffnen.

> **Note**
>
> Um eine Instanz zu starten, die nicht läuft, müssen Sie den **`AWS: Start EC2 instance`** Befehl wählen. Der **`AWS: Reboot EC2 instance`** Befehl startet nur Instanzen neu, die gerade ausgeführt werden.

3. Wählen Sie in der Eingabeaufforderung „ EC2 Instanz auswählen“ die Region aus, in der sich die Instanz befindet, die Sie starten oder neu starten möchten.
4. VS Code zeigt den Status an, während die Instance neu gestartet wird.
5. Der AWS Explorer wird aktualisiert und zeigt an, dass Ihre Instanz läuft, wenn der Neustart abgeschlossen ist.

Starten oder Neustarten einer EC2 Amazon-Instance über den Explorer AWS

1. Erweitern Sie den AWS Toolkit-Explorer und dann die Region, die die EC2 Amazon-Instance enthält, die Sie starten oder neu starten möchten.
2. Bewegen Sie den Mauszeiger über die EC2 Amazon-Instance und wählen Sie dann das Symbol (Reboot EC2 Instance).

   > **Note**
   >
   > Wenn die Instance gestoppt ist, steht nur das Symbol (EC2 Instanz starten) zur Verfügung

3. VS Code zeigt den Status an, während die Instanz neu gestartet wird.
4. Der AWS Explorer wird aktualisiert und zeigt an, dass Ihre Instanz läuft, wenn der Neustart abgeschlossen ist.

Starten oder Neustarten einer EC2 Amazon-Instance über das Rechtsklick-Menü

1. Erweitern Sie den AWS Toolkit-Explorer und dann die Region, die die EC2 Amazon-Instance enthält, die Sie starten oder neu starten möchten.
2. Klicken Sie mit der rechten Maustaste auf die EC2 Amazon-Instance, zu der Sie eine Verbindung herstellen möchten, und wählen Sie dann Reboot EC2 instance.

   > **Note**
   >
   > Wenn die Instance gestoppt ist, ist die einzige Option die EC2 Start-Instance.

3. VS Code zeigt den Status an, während die Instanz neu gestartet wird.
4. Der AWS Explorer wird aktualisiert und zeigt an, dass Ihre Instanz läuft, wenn der Neustart abgeschlossen ist.

Eine EC2 Amazon-Instance stoppen

Es gibt 3 Möglichkeiten, eine EC2 Amazon-Instance zu stoppen.

Eine EC2 Amazon-Instance über die Befehlspalette stoppen

1. Öffnen Sie in VS Code die VS Code-Befehlspalette, indem Sie auf **`command + shift + P (Windows: ctrl + shift + P)`**
2. Suchen Sie in der VS Code-Befehlspalette nach dem **`AWS: Stop EC2 instance`** Befehl und wählen Sie ihn aus, wenn er in der Liste angezeigt wird, um die Eingabeaufforderung „ EC2Instanz auswählen“ in VS Code zu öffnen.
3. Wählen Sie in der Eingabeaufforderung „ EC2 Instanz auswählen“ die Region aus, die die Instanz enthält, die Sie beenden möchten.
4. VS Code zeigt den Status an, während die Instance gestoppt wird.
5. Der AWS Explorer wird aktualisiert und zeigt an, dass Ihre Instanz gestoppt ist.

Eine EC2 Amazon-Instance im AWS Explorer beenden

1. Erweitern Sie den AWS Toolkit-Explorer und dann die Region, die die EC2 Amazon-Instance enthält, die Sie beenden möchten.
2. Bewegen Sie den Mauszeiger über die EC2 Amazon-Instance und wählen Sie dann das Symbol (Stop EC2 Instance).
3. VS Code zeigt den Status an, während die Instance beendet wird.
4. Der AWS Explorer wird aktualisiert und zeigt an, dass Ihre Instanz gestoppt wurde.

Eine EC2 Amazon-Instance über das Rechtsklick-Menü beenden

1. Erweitern Sie den AWS Toolkit-Explorer und dann die Region, die die EC2 Amazon-Instance enthält, die Sie beenden möchten.
2. Klicken Sie mit der rechten Maustaste auf die EC2 Amazon-Instance, zu der Sie eine Verbindung herstellen möchten, und wählen Sie dann Reboot EC2 instance.
3. VS Code zeigt den Status an, während die Instance beendet wird.
4. Der AWS Explorer wird aktualisiert und zeigt an, dass Ihre Instanz gestoppt wurde.

Instanz-ID kopieren

Gehen Sie wie folgt vor, um eine Instanz-ID zu kopieren.

1. Klicken Sie mit der rechten Maustaste auf die Instanz, von der Sie die ID kopieren möchten.

2. Wählen Sie Instanz-ID kopieren.
3. Die Instanz-ID wird in Ihre lokale Zwischenablage kopiert.

Namen kopieren

Gehen Sie wie folgt vor, um einen Instanznamen zu kopieren.

1. Klicken Sie mit der rechten Maustaste auf die Instanz, von der Sie den Namen kopieren möchten.
2. Wählen Sie „Instanzname kopieren“.
3. Der Instanzname wird in Ihre lokale Zwischenablage kopiert.

ARN kopieren

Gehen Sie wie folgt vor, um einen Instanz-ARN zu kopieren.

1. Klicken Sie mit der rechten Maustaste auf die Instance, von der Sie den ARN kopieren möchten.
2. Wählen Sie Instanz-ARN kopieren aus.
3. Der Instanz-ARN wird in Ihre lokale Zwischenablage kopiert.

Fehlerbehebung bei Amazon Elastic Compute Cloud

In den folgenden Abschnitten wird beschrieben, wie Sie bekannte Probleme beheben können, die bei der Arbeit mit Amazon Elastic Compute Cloud in der auftreten können AWS Toolkit for Visual Studio Code. Ausführliche Informationen zur Behebung von Problemen speziell für den EC2 Amazon-Service finden Sie im Thema Problembehandlung mit EC2 Amazon-Instances im Amazon Elastic Compute Cloud-Benutzerhandbuch.

Allgemeines Debuggen

Wenn Sie aus irgendeinem Grund auf ein Problem mit der Remoteverbindung stoßen, überprüfen Sie zunächst, ob von der AWS Konsole aus eine AWS Systems Manager Verbindung hergestellt werden kann.

Gehen Sie wie folgt vor, um über Systems Manager von der AWS Konsole aus eine Verbindung zu einer EC2 Amazon-Instance herzustellen.

1. Navigieren Sie in Ihrem Webbrowser zur AWS Konsole.
2. Schließen Sie die Authentifizierung ab, um zur AWS EC2 Konsolenseite zu gelangen.
3. Wählen Sie im EC2 Amazon-Navigationsbereich Instances aus.
4. Wählen Sie das Kästchen neben der Instance aus, zu der Sie eine Verbindung herstellen möchten.
5. Wählen Sie die Schaltfläche Connect, um den Bildschirm Mit Instanz Connect in einem neuen Browser-Tab zu öffnen.

   > **Note**
   >
   > Sie können nur dann eine Verbindung zu einer Instance herstellen, wenn sie läuft. Wenn Sie die Schaltfläche Connect nicht auswählen können, stellen Sie sicher, dass Ihre Instance läuft.

6. Wählen Sie auf dem Bildschirm Mit Instanz verbinden die Registerkarte Session Manager und dann die Schaltfläche Connect, um die Systems Manager Manager-Verbindung in Ihrem aktuellen Browser-Tab zu öffnen.

   > **Note**
   >
   > Wenn Sie Ihre Instanz vor Kurzem gestartet haben und die Option zum Herstellen einer Verbindung mit dem Systems Manager nicht verfügbar ist, müssen Sie möglicherweise einige zusätzliche Minuten warten, bis die Option verfügbar ist.

Die Zielinstanz läuft nicht

Um vom Terminal oder einer Remote-Verbindung aus eine Verbindung zu einer EC2 Amazon-Instance herzustellen, muss die Instance laufen. Bevor Sie versuchen, über das AWS Toolkit eine Verbindung zu Ihrer Instance herzustellen, starten Sie sie im AWS Explorer, AWS Management Console, oder AWS Command Line Interface.

Die Zielinstanz hat keine IAM-Rolle oder sie hat eine IAM-Rolle mit falschen Berechtigungen

Um eine Verbindung zu Ihrer EC2 Amazon-Instance herzustellen, muss dieser eine IAM-Rolle mit den richtigen Berechtigungen zugewiesen sein. Wenn Sie versuchen, eine Verbindung zu einer Instance herzustellen, der keine IAM-Rolle zugewiesen ist, werden Sie von VS Code benachrichtigt.

Wenn Sie versuchen, eine Verbindung zu einer Instance herzustellen, die über eine IAM-Rolle verfügt, aber nicht über die erforderlichen Berechtigungen verfügt, werden Sie aufgefordert, der vorhandenen IAM-Rolle die mindestens erforderlichen Aktionen als Inline-Richtlinie hinzuzufügen. Nach der Aktualisierung der Inline-Richtlinie sind Sie mit Ihrer Instance verbunden. Ausführliche Informationen zu IAM-Rollen, Berechtigungen und dem Anhängen einer Rolle an eine Instance finden Sie im EC2 Thema IAM-Rollen für Amazon im Amazon Elastic Compute Cloud-Benutzerhandbuch und im Thema Schritt 2: Instance-Berechtigungen für Session Manager überprüfen oder hinzufügen im AWS Systems Manager Manager-Benutzerhandbuch.

Das folgende Beispiel enthält die mindestens erforderlichen Aktionen.

```
"ssmmessages:CreateControlChannel",
"ssmmessages:CreateDataChannel",
"ssmmessages:OpenControlChannel",
"ssmmessages:OpenDataChannel",
"ssm:DescribeAssociation",
"ssm:ListAssociations",
"ssm:UpdateInstanceInformation
```

> **Note**
>
> Die erforderlichen Berechtigungen sind in der folgenden AWS verwalteten Richtlinie enthalten.
>
> - `AmazonSSMManagedEC2InstanceDefaultPolicy`
> - `AmazonSSMManagedInstanceCore`

Auf der Zielinstanz wird kein Systems Manager Manager-Agent ausgeführt

Dieses Problem kann aus verschiedenen Gründen auftreten. Um das Problem zu beheben, starten Sie zunächst die Instanz neu und versuchen Sie erneut, eine Verbindung herzustellen. Oder starten Sie manuell eine erste Verbindung über eine Nicht-SSM-Verbindungsmethode. Ausführlichere Informationen zu Systems Manager finden Sie im Thema Arbeiten mit dem Systems Manager Agent im AWS Systems Manager.

Beim Start gibt der EC2 Amazon-Status an, dass es läuft, aber es werden keine Verbindungen hergestellt

Wenn Sie kürzlich eine IAM-Rolle für eine Instance gestartet oder erstellt haben und keine Verbindung herstellen können, warten Sie einige weitere Minuten, bevor Sie erneut versuchen, eine Verbindung herzustellen.

Arbeiten mit Amazon Elastic Container Registry

Amazon Elastic Container Registry (Amazon ECR) ist ein AWS verwalteter Container-Registry-Service, der sicher und skalierbar ist. Auf mehrere Amazon ECR-Servicefunktionen kann über das Toolkit for VS Code Explorer zugegriffen werden.

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

Sie können auch über die VS Code-Konsole auf den gesamten Funktionsumfang von Amazon ECR 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.

Themen

- Arbeiten mit Amazon Elastic Container Registry
- Einen App Runner-Service über Amazon ECR erstellen

Arbeiten mit Amazon Elastic Container Registry

Sie können direkt vom AWS Explorer in VS Code auf den Service Amazon Elastic Container Registry (Amazon ECR) zugreifen und ihn verwenden, um ein Programm-Image in ein Amazon ECR-Repository zu übertragen. Um zu beginnen, müssen Sie die folgenden Schritte ausführen:

1. Erstellen Sie eine Dockerfile-Datei, die die zum Erstellen eines Images erforderlichen Informationen enthält.
2. Erstellen Sie ein Image aus dieser Dockerfile-Datei und markieren Sie das Image für die Verarbeitung.
3. Erstellen Sie ein Repository in Ihrer Amazon ECR-Instance.
4. Verschieben Sie das Image in Ihr Repository.

Voraussetzungen

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

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 feststellen, ob Sie berechtigt sind, auf seine Ressourcen zuzugreifen. Wir empfehlen nicht, dass Sie AWS direkt über die Anmeldeinformationen für Ihr AWS Root-Konto zugreifen. Verwenden Sie stattdessen AWS Identity and Access Management (IAM), um einen IAM-Benutzer zu erstellen und diesen Benutzer dann einer IAM-Gruppe mit Administratorberechtigungen hinzuzufügen. Sie können dann AWS mit einer speziellen URL und den Anmeldeinformationen für den IAM-Benutzer darauf zugreifen.

Wenn Sie sich für einen IAM-Benutzer registriert haben, AWS aber keinen für sich selbst erstellt haben, können Sie mithilfe der IAM-Konsole einen 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)	Verwendung von kurzfristigen Anmeldeinformationen für den Zugriff auf AWS. 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 IAM im IAM-Benutzerhandbuch.	Beachtung der Anweisungen unter Erste Schritte im AWS IAM Identity Center - Benutzerhandbuch.	Konfigurieren Sie den programmatischen Zugriff, indem Sie den AWS IAM Identity Center im AWS CLI Benutzerhandbuch zu verwendenden AWS Command Line Interface konfigurieren.
In IAM (Nicht empfohlen)	Verwendung von langfristigen Anmeldeinformationen für den Zugriff auf AWS.	Folgen Sie den Anweisungen unter Erstellen eines IAM-Benutzers für den Notfallzugriff im IAM-Benutzerhandbuch.	Konfigurieren Sie den programmatischen Zugriff unter Zugriffsschlüssel für IAM-Benutzer verwalten im IAM-Benutzerhandbuch.

Um sich als dieser neue IAM-Benutzer anzumelden, melden Sie sich von der AWS Konsole ab und verwenden Sie dann die folgende URL. In der folgenden URL, wobei your_aws_account_id Ihre AWS

Kontonummer ohne Bindestriche ist (wenn Ihre Kontonummer beispielsweise lautet, lautet Ihre AWS Konto-ID): `1234-5678-9012` AWS `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 Konto-ID enthält, können Sie einen AWS Kontoalias erstellen. Wählen Sie im IAM-Dashboard die Option Anpassen und geben Sie einen Kontoalias ein. Dies kann Ihr Firmenname sein. Weitere Informationen finden Sie unter Ihre AWS Konto-ID und deren 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 im Benutzerhandbuch zur Installation der Docker Engine Ihr bevorzugtes Betriebssystem auswählen und den Anweisungen folgen.

Installieren und konfigurieren Sie AWS CLI Version 2

Installieren und konfigurieren Sie AWS CLI Version 2, indem Sie im Benutzerhandbuch zur Installation, Aktualisierung und Deinstallation der AWS CLI Version 2 Ihr bevorzugtes Betriebssystem auswählen.

1. Erstellen einer Dockerfile-Datei

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

Erstellen einer Dockerfile-Datei

1. Verwenden Sie das Toolkit for VS Code Explorer, um zu dem Verzeichnis zu navigieren, in dem Sie Ihre Dockerfile speichern möchten.
2. Erstellen Sie eine neue Datei namens Dockerfile.

> **Note**
>
> VS Code könnte Sie auffordern, einen Dateityp oder eine Dateierweiterung auszuwählen. Wählen Sie in diesem Fall Klartext aus. Vs Code hat die Erweiterung „Dockerfile“. Wir empfehlen jedoch, diese nicht zu verwenden. Dies liegt daran, dass die Erweiterung Konflikte mit bestimmten Versionen von Docker oder anderen zugehörigen Anwendungen verursachen kann.

Bearbeiten Sie Ihr Dockerfile mit VS Code

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

Nachdem die Dateierweiterung aus Ihrer Dockerfile-Datei 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-Datei, die ein Ubuntu-18.04-Image verwendet. Die RUN-Anweisungen aktualisieren die Paketcaches. Installieren Sie Softwarepakete für den Webserver und schreiben Sie dann den Inhalt „Hello World!“ in das Dokumenten-Stammverzeichnis des Webservers. Die EXPOSE-Anweisung stellt Port 80 auf dem Container bereit, und die CMD-Anweisung startet den Webserver.

3. Speichern Sie Ihre Dockerfile-Datei.

> **Important**
>
> Stellen Sie sicher, dass Ihrem Dockerfile keine Erweiterung an den Namen angehängt ist. Eine Dockerfile-Datei mit Erweiterungen kann Konflikte mit bestimmten Versionen von Docker oder anderen zugehörigen Anwendungen verursachen.

2. Erstellen Sie Ihr Image aus Ihrem Dockerfile

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

Erstellen Sie ein Image aus Ihrem Dockerfile

1. Verwenden Sie die Docker-CLI oder eine CLI, die in Ihre Docker-Instanz integriert ist, um zu dem Verzeichnis zu navigieren, das Ihre Dockerfile enthält.
2. Führen Sie den Befehl Docker build aus, um das Image zu erstellen, das in Ihrem Dockerfile definiert ist.

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

3. Führen Sie den Befehl Docker images aus, um zu überprüfen, ob das Image korrekt erstellt wurde.

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

Example Beispielausgabe:

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

4.

> **Note**
>
> Dieser Schritt ist nicht notwendig, um Ihr Image zu erstellen oder zu übertragen, aber Sie können sehen, wie das Programm-Image funktioniert, wenn es ausgeführt wird.

Verwenden Sie den Befehl Docker run, um das neu erstellte Image auszuführen.

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

Die Option -p, die im vorherigen Beispiel angegeben wurde, ordnet den exponierten Port 80 auf dem Container dem Port 80 des Hostsystems zu. Wenn Sie Docker lokal ausführen, navigieren Sie mit Ihrem Webbrowser zu http://localhost:80. Wenn das Programm korrekt ausgeführt wurde, ein „Hello World!“ Die Anweisung wird angezeigt.

Weitere Informationen zum Befehl Docker run finden Sie in der Docker run reference auf der Docker-Website.

3. Erstellen Sie ein neues Repository

Wenn Sie das Image in Ihre Amazon-ECR-Instance hochladen möchten, erstellen Sie ein neues Repository, in dem es gespeichert werden kann.

Neues Amazon ECR-Repository erstellen

1. Wählen Sie in der VS Code-Aktivitätsleiste das AWS Toolkit-Symbol aus.
2. Erweitern Sie das AWS Explorer-Menü.
3. Suchen Sie die AWS Standardregion, die mit Ihrem AWS Konto verknüpft ist. Wählen Sie es dann aus, um eine Liste der Dienste anzuzeigen, die über das Toolkit for VS Code bereitgestellt werden.
4. Wählen Sie die Option ECR +, um mit der Erstellung eines neuen Repositorys zu beginnen.
5. Folgen Sie den Anweisungen, um den Vorgang abzuschließen.
6. Nach Abschluss des Vorgangs können Sie über den ECR-Bereich des AWS Explorer-Menüs auf Ihr neues Repository zugreifen.

4. Bilder per Push übertragen, abrufen und löschen

Nachdem Sie ein Image aus Ihrer Dockerfile-Datei erstellt und ein Repository erstellt haben, können Sie Ihr Image in Ihr Amazon-ECR-Repository übertragen. Darüber hinaus können Sie mit dem AWS Explorer mit Docker und der AWS CLI Folgendes tun:

- Ein Image aus Ihrem Repository abrufen
- Ein Bild, das in deinem Repository gespeichert ist, löschen
- Ihr Repository löschen

Authentifizieren Sie Docker mit Ihrer Standardregistrierung

Für den Datenaustausch zwischen Amazon-ECR-Instances und Docker-Instanzen ist eine Authentifizierung erforderlich. So authentifizieren Sie Docker für Ihre Registry:

1. Öffnen Sie ein Befehlszeilenbetriebssystem, das mit Ihrer AWS CLI-Instanz verbunden ist.
2. Verwenden Sie get-login-passworddiese Methode, um sich bei Ihrer privaten ECR-Registrierung zu authentifizieren.

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

> **Important**
>
> Im vorherigen Befehl müssen Sie **`region`** sowohl die als auch **`AWS_account_id`** die für Ihr AWS Konto spezifischen Informationen aktualisieren.

Taggen Sie ein Bild und übertragen Sie es in Ihr Repository

Nachdem Sie Docker mit Ihrer Instanz von authentifiziert haben AWS, übertragen Sie ein Bild in Ihr Repository.

1. Verwenden Sie den Befehl Docker images, um die Bilder anzuzeigen, die Sie lokal gespeichert haben, und um das Bild zu identifizieren, das Sie taggen möchten.

   ```
   docker images
   ```

   Example Beispielausgabe:

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

2. Markieren Sie Ihr Image mit dem Befehl Docker tag.

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

3. Pushen Sie das markierte Bild mit dem Befehl Docker tag in Ihr Repository.

   ```
   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 Ihr markiertes Bild erfolgreich in Ihr Repository hochgeladen wurde, ist es im AWS Explorer-Menü sichtbar.

Ein Bild aus Amazon ECR abrufen

- Sie können ein Bild in Ihre lokale Instance des Befehls Docker tag abrufen.

  ```
  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 Bild aus Ihrem Amazon ECR-Repository

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

1. Erweitern Sie im AWS Explorer das ECR-Menü
2. Erweitern Sie das Repository, aus dem Sie ein Bild löschen möchten
3. Wählen Sie das Bild-Tag aus, das mit dem Bild verknüpft ist, das Sie löschen möchten, indem Sie das Kontextmenü öffnen (Rechtsklick)

4. Wählen Sie „Tag löschen“... Option, um alle gespeicherten Bilder zu löschen, die mit diesem Tag verknüpft sind

Löschen Sie ein Bild mit der AWS CLI

- Sie können ein Bild auch mit dem batch-delete-image Befehl AWS ecr aus Ihrem Repository löschen.

```
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-Instance

Es gibt zwei Methoden, um ein Repository aus VS Code zu löschen. Die erste Methode besteht darin, den AWS Explorer zu verwenden.

1. Erweitern Sie im AWS Explorer das ECR-Menü
2. Wählen Sie das Repository aus, das Sie löschen möchten, indem Sie das Kontextmenü (Rechtsklick) öffnen
3. Wählen Sie Repository löschen... Option zum ausgewählten Repository

Löschen Sie ein Amazon ECR-Repository aus der CLI AWS

- Sie können ein Repository mit dem Befehl AWS ecr delete-repository löschen.

> **Note**
>
> Standardmäßig können Sie ein Repository, das Bilder enthält, nicht löschen. Das Flag --force ermöglicht dies jedoch.

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

Example Beispielausgabe:

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

Einen App Runner-Service über Amazon ECR erstellen

Das folgende Thema beschreibt, wie Sie einen AWS App Runner Service vom Knoten Amazon Elastic Container Registry (Amazon ECR) aus erstellen und starten, in der AWS Toolkit for Visual Studio Code. Ausführliche Informationen zu den Diensten AWS App Runner und Amazon ECR finden Sie in den AWS App Runnerund den Amazon ECR-Benutzerhandbüchern.

Voraussetzungen

Bevor Sie einen ECR AWS App Runner von Amazon aus im AWS Toolkit erstellen und starten können, müssen Sie die folgenden Schritte ausführen. Eine ausführliche Anleitung zur Durchführung dieser Verfahren finden Sie im Thema Arbeiten mit Amazon Elastic Container Registry in diesem Benutzerhandbuch.

1. Erstellen Sie einen `dockerfile`.
2. Erstellen Sie ein Image aus Ihrem`dockerfile`.
3. Erstellen Sie ein neues Repository.
4. Taggen Sie ein Bild und übertragen Sie es in Ihr Repository.

Einen AWS App Runner Service aus einem vorhandenen Amazon ECR-Repository erstellen

Das folgende Verfahren beschreibt, wie Sie einen AWS App Runner Service aus einem vorhandenen Amazon ECR-Repository im AWS Toolkit erstellen.

1. Erweitern Sie im AWS Explorer die Region, die das Amazon ECR-Repository enthält, aus dem Sie einen AWS App Runner Service erstellen möchten.
2. Erweitern Sie den Amazon ECR-Serviceknoten, um Ihre Amazon ECR-Repositorys anzuzeigen.
3. Öffnen Sie das Kontextmenü für das Amazon ECR-Repository oder das Repository-Image, aus dem Sie einen AWS App Runner Service erstellen möchten (klicken Sie mit der rechten Maustaste).
4. Wählen Sie im Kontextmenü die Option App Runner Service erstellen, um den AWS App Runner Erstellungsassistenten in VS Code zu öffnen
5. Geben Sie unter Geben Sie einen Port für den neuen Dienst ein (1/5) die Portnummer ein, die Sie verwenden möchten, und drücken Sie dann, **`Enter`** um fortzufahren.
6. Wählen Sie unter Umgebungsvariablen konfigurieren (2/5) die Option Datei verwenden... aus. um zu suchen, wählen Sie Ihre lokalen Dateien durchsuchen oder wählen Sie Überspringen, um diesen Schritt zu überspringen.
7. Wählen Sie unter Wählen Sie eine Rolle aus, die aus ECR abgerufen werden soll (3/5) eine bestehende IAM-Rolle aus der Liste aus.

> **Note**
>
> Die AppRunnerECRAccessRollenzugriffsrolle ist erforderlich, um einen AWS App Runner Service aus einer privaten Amazon ECR-Registrierung zu erstellen. Wenn in der Liste keine gültige Rolle verfügbar ist, wählen Sie + (Rolle erstellen...) Symbol, um automatisch eine AppRunnerECRAccessRolle zu erstellen und Ihrer Registrierung zuzuweisen.

8. Geben Sie unter Name your service (4/5) einen Namen für Ihren neuen Service ein und drücken Sie dann, **`Enter`** um fortzufahren.
9. Wählen Sie unter Instanzkonfiguration auswählen (5/5) die **`Memory`** Konfiguration **`vCPU`** und aus der Liste aus, um Ihren neuen Dienst zu erstellen.
10. Erweitern Sie im AWS Explorer den App Runner-Dienstknoten, um Ihre AWS App Runner Ressourcen anzuzeigen. Wenn Ihr neuer Dienst erfolgreich erstellt wurde, wird der Status automatisch auf Wird ausgeführt aktualisiert.

Arbeiten mit Amazon Elastic Container Service

Das AWS Toolkit for Visual Studio Code bietet eine gewisse Unterstützung für Amazon Elastic Container Service (Amazon ECS). Das Toolkit for VS Code unterstützt Sie bei bestimmten Arbeiten im Zusammenhang mit Amazon ECS, wie z. B. der Erstellung von Aufgabendefinitionen.

Themen

- IntelliSense Für Amazon ECS-Aufgabendefinitionsdateien verwenden
- Amazon Elastic Container Service Exec in AWS Toolkit for Visual Studio Code

IntelliSense Für Amazon ECS-Aufgabendefinitionsdateien verwenden

Bei der Arbeit mit Amazon Elastic Container Service (Amazon ECS) könnten Sie unter anderem Aufgabendefinitionen erstellen, wie unter Erstellen einer Aufgabendefinition im Amazon Elastic Container Service Developer Guide beschrieben. Wenn Sie das installieren AWS Toolkit for Visual Studio Code, umfasst die Installation IntelliSense Funktionen für Amazon ECS-Aufgabendefinitionsdateien.

Voraussetzungen

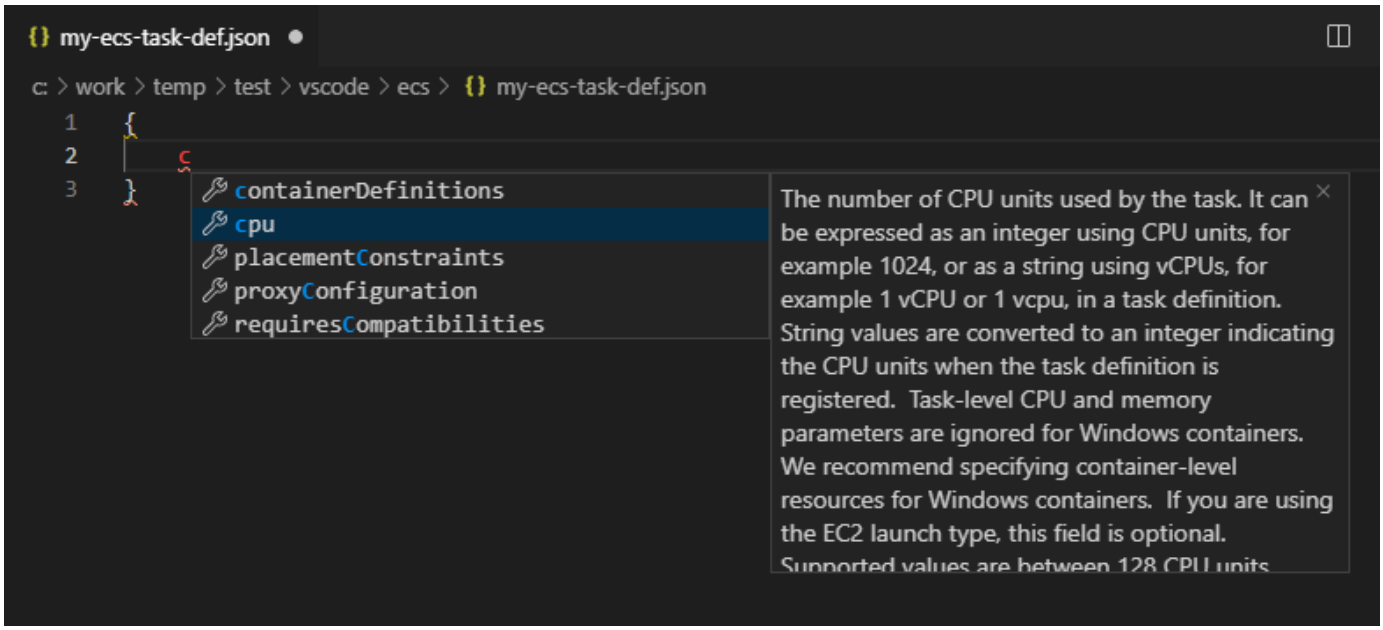
- Stellen Sie sicher, dass Ihr System die unter Installation des Toolkit for VS Code angegebenen Voraussetzungen erfüllt.

Verwendung IntelliSense in Amazon ECS-Aufgabendefinitionsdateien

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

1. Erstellen Sie eine JSON-Datei für Ihre 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 ersten geschweiften Klammern ein.
3. Geben Sie den Buchstaben „c“ ein, als ob Sie der Definition `cpu` hinzufügen wollten. Beachten Sie den sich öffnenden IntelliSense Dialog, der dem folgenden ähnelt.



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 auf Amazon gehostet werden AWS Fargate, gelten für Amazon ECS Exec unterschiedliche Versionsanforderungen.

- Wenn Sie Amazon verwenden EC2, müssen Sie ein für Amazon ECS optimiertes AMI verwenden, das nach dem 20. Januar 2021 veröffentlicht wurde, mit einer Agentenversion von 1.50.2 oder höher. Zusätzliche Informationen finden Sie im Entwicklerhandbuch Amazon ECS Optimized 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.

Mit Amazon arbeiten EventBridge

Der AWS Toolkit for Visual Studio Code (VS Code) bietet Unterstützung für Amazon EventBridge. Mit dem Toolkit for VS Code können Sie mit bestimmten Aspekten von arbeiten EventBridge, z. B. mit Schemas.

Themen

- Mit EventBridge Amazon-Schemas arbeiten

Mit EventBridge Amazon-Schemas arbeiten

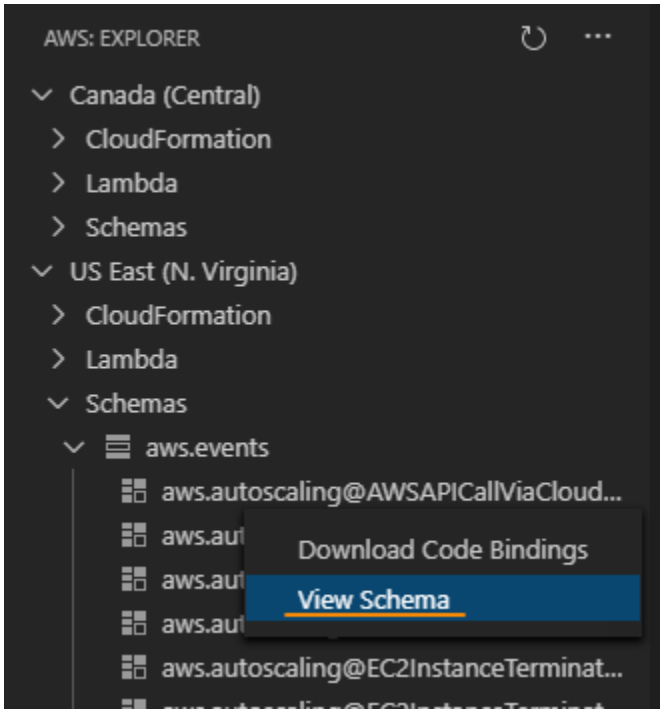
Sie können den AWS Toolkit for Visual Studio Code (VS-Code) verwenden, um verschiedene Operationen mit EventBridge Amazon-Schemas durchzuführen.

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



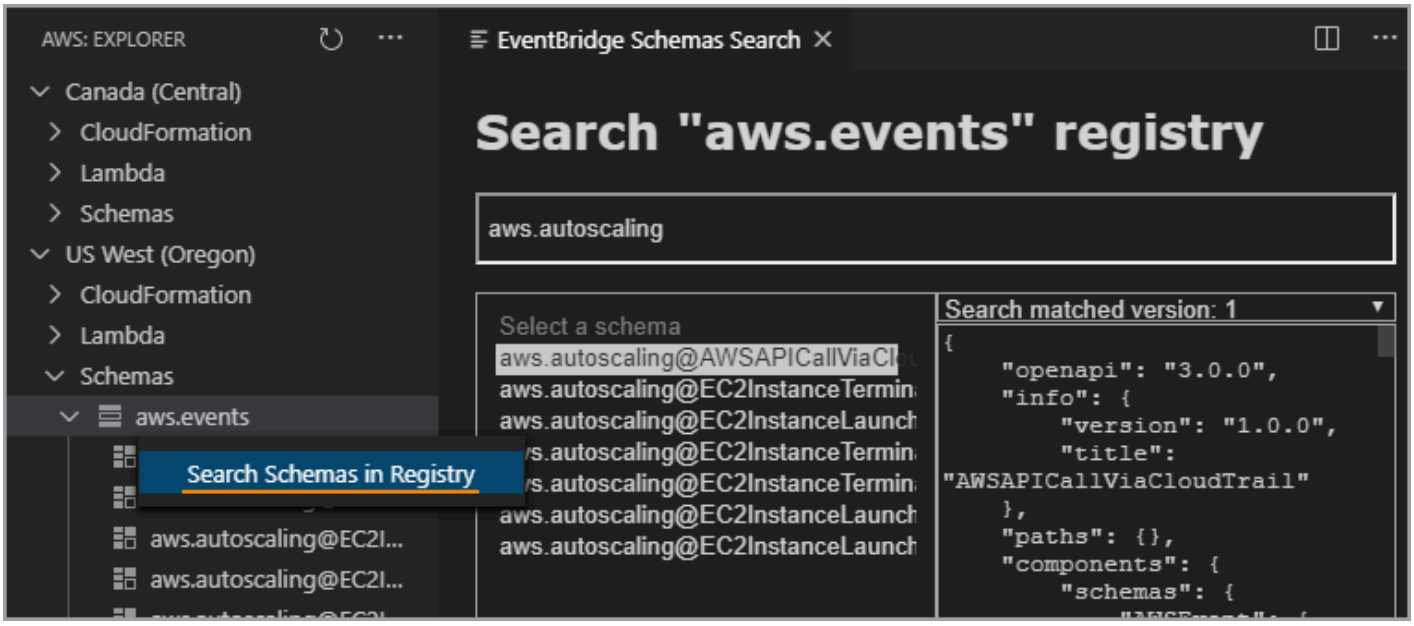
Suchen eines verfügbaren Schemas

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 Schemas Search (Schemasuche) den Titel des zu suchenden Schemas ein. 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 Richtlinienvalidierung und benutzerdefinierte Richtlinienprüfungen. Die Richtlinienvalidierung 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 Richtlinienvalidierung umfassen Sicherheitswarnungen, Fehler, allgemeine Warnungen und Richtlinienvorschläge.

Sie können auf der Grundlage Ihrer Sicherheitsstandards auch benutzerdefinierte Richtlinienprüfungen für neuen Zugriff durchführen. 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 ManagementBenutzerhandbuch.

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-Richtlinienvalidierung 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.

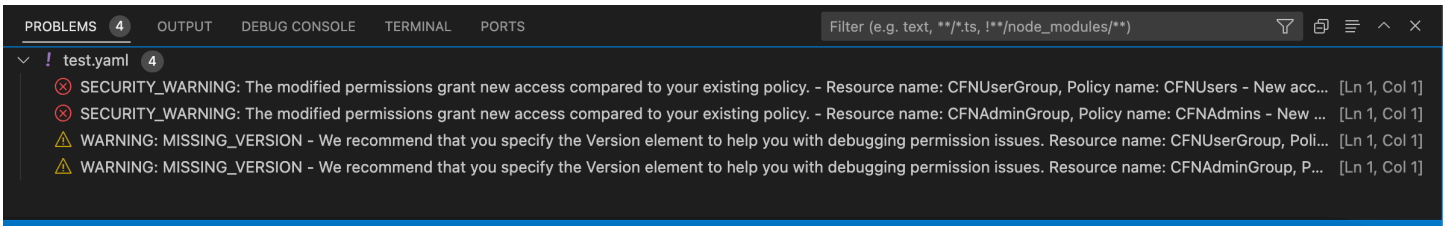
Voraussetzungen

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-Richtlinienvalidator für oder den AWS CloudFormation IAM-Richtlinienvalidator für Terraform, der für die Python-CLI-Tools erforderlich und im Fenster IAM-Richtlinienprüfungen angegeben ist.
- Konfigurieren Sie Ihre Rollenanmeldedaten. 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 4 Arten von Prüfungen:

- Richtlinie validieren
- CheckAccessNotGranted
- CheckNoNewAccess
- CheckNoPublicAccess

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

> **Note**
>
> Konfigurieren Sie Ihre AWS Rollenanmeldedaten, bevor Sie eine Prüfung durchführen. Zu den unterstützten Dateien gehören die folgenden Dokumenttypen: AWS CloudFormation Vorlagen, Terraform-Pläne und JSON-Richtliniendokumente
> Dateipfadverweise werden in der Regel von Ihrem Administrator oder Sicherheitsteam bereitgestellt und können ein Systemdateipfad oder eine 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.
> Mit jeder Überprüfung der benutzerdefinierten Richtlinien ist eine Gebühr verbunden. Einzelheiten zu den Preisen für benutzerdefinierte Policy-Checks finden Sie in der Preisübersicht für AWS IAM Access Analyzer.

Validate Policy wird ausgeführt

Die Überprüfung „Richtlinie validieren“, auch als Richtlinienvalidierung 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**`CRTL+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 „Richtlinienvalidierung 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, mit der überprüft wird, ob bestimmte IAM-Aktionen gemäß Ihrer Richtlinie nicht zulässig sind.

> **Note**
>
> Dateipfadverweise werden in der Regel von Ihrem Administrator oder Sicherheitsteam bereitgestellt und können ein Systemdateipfad oder eine 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. Es muss mindestens eine Aktion oder Ressource angegeben werden, und die Datei sollte nach dem folgenden Beispiel strukturiert werden:
>
> ```
> {"actions": ["action1", "action2", "action3"], "resources":
> ["resource1", "resource2", "resource3"]}
> ```

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-BefehlspaletteCRTL+Shift+P, indem Sie auf, suchen und dann klickenIAM 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. In das Texteingabefeld können Sie eine durch Kommas getrennte Liste eingeben, die Aktionen und Ressourcen enthält. ARNs Es muss mindestens eine Aktion oder Ressource bereitgestellt werden.
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.

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**CRTL+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.
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.

Wird ausgeführt CheckNoPublicAccess

CheckNoPublicAccess ist eine benutzerdefinierte Richtlinienüberprüfung, mit der überprüft wird, ob Ihre Richtlinie öffentlichen Zugriff auf unterstützte Ressourcentypen in Ihrer Vorlage gewährt.

Spezifische Informationen zu unterstützten Ressourcentypen finden Sie in den terraform-iam-policy-validator GitHub Repositorien cloudformation-iam-policy-validatorund.

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**CRTL+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 CheckNoPublicAccess.
5. Wählen Sie die Schaltfläche Benutzerdefinierte Richtlinienüberprüfung ausführen.
6. Überprüfen Sie im Problembereich in VS Code die Ergebnisse Ihrer Richtlinienüberprüfung. Benutzerdefinierte Richtlinienprüfungen geben ein `FAIL` Ergebnis `PASS` oder ein Ergebnis zurück.
7. 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 AWS Toolkit for Visual Studio Code

AWS IoT in AWS Toolkit for Visual Studio Code ermöglicht es Ihnen, mit dem AWS IoT Dienst zu interagieren und gleichzeitig Unterbrechungen Ihres Arbeitsablaufs in VS Code zu minimieren. Dieses Benutzerhandbuch soll Ihnen den Einstieg in die Nutzung der AWS IoT Servicefunktionen erleichtern, die in der verfügbar sind. AWS Toolkit for Visual Studio Code Weitere Informationen zum AWS IoT Service finden Sie im Entwicklerhandbuch Was ist AWS IoT?

AWS IoT Voraussetzungen

Um mit der Nutzung AWS IoT von Toolkit for VS Code zu beginnen, stellen Sie sicher, dass Ihr AWS Konto und VS Code die Anforderungen in diesen Handbüchern erfüllen:

- Informationen zu den AWS Kontoanforderungen und AWS Benutzerberechtigungen, die für den AWS IoT Dienst spezifisch sind, finden Sie im Entwicklerhandbuch Erste Schritte mit AWS IoT Core.
- Die spezifischen Anforderungen für Toolkit for VS Code finden Sie im Benutzerhandbuch zum Einrichten des Toolkit for VS Code.

AWS IoT Dinge

AWS IoT verbindet Geräte mit AWS Cloud-Diensten und Ressourcen. Sie können Ihre Geräte mithilfe AWS IoT von Objekten verbinden, die als Dinge bezeichnet werden. 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 zu diesen AWS IoT Dingen finden Sie im Entwicklerhandbuch Geräte verwalten mit AWS IoT.

AWS IoT Dinge verwalten

Das Toolkit for VS Code verfügt über mehrere Funktionen, mit denen Sie Ihr AWS IoT Ding-Management effizienter gestalten können. Auf diese Weise können Sie das VS Code-Toolkit verwenden, um Ihre AWS IoT Dinge zu verwalten:

- Create a thing
- Attach a certificate to a thing
- Detach a certificate from a thing
- Delete a thing

So erstellen Sie eine Objekt

1. Erweitern Sie im AWS Explorer die Überschrift IoT-Dienst und wählen Sie Dinge im Kontext aus (klicken Sie mit der rechten Maustaste).
2. Wählen Sie im Kontextmenü die Option Ding erstellen, um ein Dialogfeld zu öffnen.
3. Folgen Sie der Aufforderung, indem Sie einen Namen für Ihr IoT-Ding in das Feld Dingname eingeben.
4. Wenn dieser Vorgang abgeschlossen ist, wird ein Dingsymbol, gefolgt von dem von Ihnen angegebenen Namen, im Abschnitt Ding angezeigt.

So fügen Sie einem Objekt ein Zertifikat an

1. Erweitern Sie im AWS Explorer den Bereich IoT-Service.
2. Suchen Sie im Unterabschnitt Dinge nach dem Ding, an das Sie das Zertifikat anhängen möchten.
3. Wählen Sie das Ding im Kontext aus (klicken Sie mit der rechten Maustaste) und wählen Sie im Kontextmenü die Option Zertifikat anhängen, um eine Eingabeselektion mit einer Liste Ihrer Zertifikate zu öffnen.
4. Wählen Sie aus der Liste die Zertifikat-ID aus, die dem Zertifikat entspricht, das Sie an Ihr Ding anhängen möchten.
5. Wenn dieser Vorgang abgeschlossen ist, kann auf Ihr Zertifikat im AWS Explorer zugegriffen werden, und zwar als Element der Sache, an die Sie es angehängt haben.

So trennen Sie ein Zertifikat von einem Objekt

1. Erweitern Sie im AWS Explorer den IoT-Servicebereich
2. Suchen Sie im Unterabschnitt Things (Objekte) nach dem Objekt, dessen Zertifikat Sie trennen möchten.
3. Wählen Sie das Ding im Kontext aus (klicken Sie mit der rechten Maustaste) und wählen Sie im Kontextmenü die Option Zertifikat trennen aus.
4. Wenn dieser Vorgang abgeschlossen ist, wird das abgetrennte Zertifikat nicht mehr unter diesem Ding im AWS Explorer angezeigt, aber es ist weiterhin über den Unterabschnitt Zertifikate zugänglich.

So löschen Sie ein Objekt

1. Erweitern Sie im AWS Explorer den Bereich IoT-Service.
2. Suchen Sie im Unterabschnitt Dinge nach dem Ding, das Sie löschen möchten.
3. Wählen Sie das Ding im Kontext aus (klicken Sie mit der rechten Maustaste) und wählen Sie im Kontextmenü die Option Ding löschen, um es zu löschen.
4. Wenn dieser Vorgang abgeschlossen ist, ist das gelöschte Ding nicht mehr im Unterabschnitt „Dinge“ verfügbar.

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

AWS IoT Zertifikate

Zertifikate sind eine gängige Methode, um eine sichere Verbindung zwischen Ihren AWS IoT Diensten und Geräten herzustellen. 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 zu AWS IoT Zertifikaten finden Sie im Entwicklerhandbuch Authentication (IoT).

Verwalten von Zertifikaten

Das VS Code Toolkit bietet eine Vielzahl von Möglichkeiten, wie Sie Ihre AWS IoT Zertifikate direkt vom AWS Explorer aus verwalten können.

- Create a certificate
- Change a certificate status
- Attach a policy to a certificate
- Delete a certificate

Um ein Zertifikat zu erstellen AWS IoT

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

1. Erweitern Sie im AWS Explorer den Abschnitt IoT-Dienst und wählen Sie Zertifikate im Kontext aus (klicken Sie mit der rechten Maustaste).
2. Wählen Sie im Kontextmenü die Option Zertifikat erstellen aus, um ein Dialogfeld zu öffnen.
3. Wählen Sie ein Verzeichnis in Ihrem lokalen Dateisystem aus, um Ihr RSA-Schlüsselpaar und Ihr X.509-Zertifikat zu speichern.

> **Note**
>
> - Die Standarddateinamen enthalten die Zertifikat-ID als Präfix.
> - Nur das X.509-Zertifikat wird über den Dienst in Ihrem AWS Konto gespeichert. AWS IoT
> - 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, löscht das AWS Toolkit das Zertifikat aus Ihrem AWS Konto.

So ändern Sie den Status eines Zertifikats

Der Status eines einzelnen Zertifikats wird im AWS Explorer neben seiner ID angezeigt und kann wie folgt festgelegt werden: aktiv, inaktiv oder gesperrt.

> **Note**
>
> - Ihr Zertifikat muss einen aktiven Status haben, bevor Sie es verwenden können, um Ihr Gerät mit Ihrem AWS IoT Dienst zu verbinden.
> - Ein inaktives Zertifikat kann aktiviert werden, unabhängig davon, ob es zuvor deaktiviert wurde oder standardmäßig inaktiv ist.
> - Ein Zertifikat, das widerrufen wurde, kann nicht erneut aktiviert werden.

1. Erweitern Sie im AWS Explorer den Bereich IoT-Service.
2. Suchen Sie im Unterabschnitt Zertifikate das Zertifikat, das Sie ändern möchten.
3. Wählen Sie das Zertifikat im Kontext aus (klicken Sie mit der rechten Maustaste darauf), um ein Kontextmenü zu öffnen, in dem die für dieses Zertifikat verfügbaren Optionen zur Statusänderung angezeigt werden.

- Wenn ein Zertifikat den Status inactive (inaktiv) hat, wählen Sie activate (Aktivieren) aus, um den Status in active (aktiv) zu ändern.
- Hat ein Zertifikat den Status active (aktiv), wählen Sie deactivate (Deaktivieren) aus, um den Status in inactive (inaktiv) zu ändern.
- Wenn ein Zertifikat weder den Status active (aktiv) noch inactive (inaktiv) hat, wählen Sie revoke (Widerrufen) aus, um den Status in revoked (widerrufen) zu ändern.

> **Note**
>
> Jede dieser Aktionen zur Statusänderung ist auch verfügbar, wenn Sie ein Zertifikat auswählen, das an ein Objekt angehängt ist, während es im Unterabschnitt Dinge angezeigt wird.

So fügen Sie einem Zertifikat eine IoT-Richtlinie an

1. Erweitern Sie im AWS Explorer den Bereich IoT-Service.
2. Suchen Sie im Unterabschnitt Zertifikate das Zertifikat, das Sie ändern möchten.
3. Wählen Sie das Zertifikat im Kontext aus (klicken Sie mit der rechten Maustaste darauf) und wählen Sie im Kontextmenü die Option Richtlinie anhängen, um eine Eingabeselektion mit einer Liste Ihrer verfügbaren Richtlinien zu öffnen.
4. Wählen Sie die Richtlinie aus, die Sie an das Zertifikat anhängen möchten.
5. Wenn dies abgeschlossen ist, wird die von Ihnen ausgewählte Richtlinie dem Zertifikat als Untermenüelement hinzugefügt.

So trennen Sie eine IoT-Richtlinie von einem Zertifikat

1. Erweitern Sie im AWS Explorer den Bereich IoT-Service.
2. Suchen Sie im Unterabschnitt Zertifikate das Zertifikat, das Sie ändern möchten.
3. Erweitern Sie das Zertifikat und suchen Sie nach der Richtlinie, die Sie trennen möchten.
4. Wählen Sie die Richtlinie im Kontext aus (klicken Sie mit der rechten Maustaste darauf) und wählen Sie im Kontextmenü die Option Trennen aus.
5. Wenn dieser Vorgang abgeschlossen ist, ist die Richtlinie kein Element mehr, auf das Sie von Ihrem Zertifikat aus zugreifen können, sondern sie steht im Unterabschnitt Richtlinie zur Verfügung.

So löschen Sie ein Zertifikat

1. Erweitern Sie im AWS Explorer die Überschrift IoT-Dienst.
2. Suchen Sie im Unterabschnitt Zertifikate das Zertifikat, das Sie löschen möchten.
3. Wählen Sie das Zertifikat im Kontext aus (klicken Sie mit der rechten Maustaste darauf) und wählen Sie im Kontextmenü die Option Zertifikat löschen.

> **Note**
>
> Sie können ein Zertifikat nicht löschen, wenn es einem Objekt angefügt ist oder einen aktiven Status hat. Sie können ein Zertifikat löschen, dem Richtlinien angefügt sind.

AWS IoT Richtlinien

AWS IoT Kernrichtlinien werden durch JSON-Dokumente definiert, die jeweils eine oder mehrere Richtlinienerklärungen enthalten. Richtlinien definieren AWS IoT, wie AWS, und Ihr Gerät miteinander interagieren können. Weitere Informationen zum Erstellen eines Richtliniendokuments finden Sie im Entwicklerhandbuch IoT Polices.

> **Note**
>
> Benannte Richtlinien sind versioniert, sodass Sie sie rückgängig machen können. Im AWS Explorer werden Ihre IoT-Richtlinien im Unterabschnitt Richtlinien im IoT-Dienst aufgeführt. Sie können die Versionen der Richtlinien anzeigen, indem Sie eine Richtlinie erweitern. Die Standardversion ist mit einem Sternchen gekennzeichnet.

Verwalten von Richtlinien

Das Toolkit for VS Code bietet Ihnen verschiedene Möglichkeiten, Ihre AWS IoT Servicerichtlinien zu verwalten. Auf diese Weise können Sie Ihre Richtlinien direkt im AWS Explorer in VS Code verwalten oder ändern:

- Create a policy
- Upload a new policy version
- Edit a policy version
- Change the policy version defualt
- Change the policy version defualt

Um eine AWS IoT Richtlinie zu erstellen

> **Note**
>
> Sie können im AWS Explorer eine neue Richtlinie erstellen, aber das JSON-Dokument, das die Richtlinie definiert, muss bereits in Ihrem Dateisystem vorhanden sein.

1. Erweitern Sie im AWS Explorer den Bereich IoT-Service.

2. Wählen Sie im Kontext den Unterabschnitt Richtlinien aus (klicken Sie mit der rechten Maustaste darauf) und wählen Sie Richtlinie aus Dokument erstellen aus, um das Eingabefeld Richtlinienname zu öffnen.
3. Geben Sie einen Namen ein und folgen Sie den Anweisungen, um ein Dialogfeld 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 verfügbar, wenn der Vorgang abgeschlossen ist.

Um eine neue AWS IoT Richtlinienversion hochzuladen

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 mit dem AWS Explorer eine neue Version zu erstellen.

1. Erweitern Sie im AWS Explorer den Bereich IoT-Service.
2. Erweitern Sie den Unterabschnitt Richtlinien, um Ihre AWS IoT Richtlinien anzuzeigen
3. Wählen Sie im Kontext (mit der rechten Maustaste) 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 im Explorer zugegriffen werden. AWS

Um eine AWS IoT Richtlinienversion zu bearbeiten

Ein Richtliniendokument 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 aus dem AWS Explorer in Ihren AWS IoT Dienst hochladen.

1. Erweitern Sie im AWS Explorer den Bereich IoT-Service.

2. Erweitern Sie den Unterabschnitt Richtlinien und suchen Sie die Richtlinie, die Sie aktualisieren möchten. Richtlinie aus Dokument erstellen, um das Eingabefeld Richtlinienname zu öffnen.
3. Erweitern Sie die Richtlinie, die Sie aktualisieren möchten, und wählen Sie dann im Kontext (Rechtsklick) die Richtlinienversion aus, die Sie bearbeiten möchten.
4. Wählen Sie im Kontextmenü die Option Ansicht aus, um die Richtlinienversion in VS Code zu öffnen
5. Wenn das Richtliniendokument geöffnet ist, nehmen Sie 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 sie mit dem AWS Explorer zu verfolgen, wiederholen Sie die im Upload a new policy version Verfahren beschriebenen Schritte.

So wählen Sie eine neue Standardversion einer Richtlinie aus

1. Erweitern Sie im AWS Explorer den Bereich IoT-Service.
2. Erweitern Sie den Unterabschnitt Policies (Richtlinien) und suchen Sie die Richtlinie, die Sie aktualisieren möchten.
3. Erweitern Sie die Richtlinie, die Sie aktualisieren möchten, und wählen Sie dann im Kontext (mit der rechten Maustaste) die Richtlinienversion aus, die Sie festlegen möchten, und wählen Sie Als Standard festlegen aus.
4. Wenn dieser Vorgang abgeschlossen ist, wird neben der neuen Standardversion, die Sie ausgewählt haben, ein Stern angezeigt.

So löschen Sie -Richtlinien

> **Note**
>
> Bevor Sie eine Richtlinie oder eine Richtlinienversion löschen können, müssen bestimmte Bedingungen erfüllt sein.

- Sie können eine Richtlinie nicht löschen, wenn sie an ein Zertifikat angehängt ist.
- Sie können eine Richtlinie nicht löschen, wenn sie nicht standardmäßige Versionen hat.
- Sie können die Standardversion einer Richtlinie nur löschen, wenn eine neue Standardversion ausgewählt oder die gesamte Richtlinie gelöscht wurde.
- Bevor Sie eine gesamte Richtlinie löschen können, müssen zuerst alle nicht standardmäßigen Versionen dieser Richtlinie gelöscht werden.

1. Erweitern Sie im AWS Explorer den Bereich IoT-Service.
2. Erweitern Sie den Unterabschnitt Policies (Richtlinien) und suchen Sie die Richtlinie, die Sie aktualisieren möchten.
3. Erweitern Sie die Richtlinie, die Sie aktualisieren möchten, und wählen Sie dann im Kontext (mit der rechten Maustaste) die Richtlinienversion aus, die Sie löschen möchten, und wählen Sie Löschen aus.
4. Wenn eine Version gelöscht wird, ist sie im Explorer nicht mehr sichtbar.
5. Wenn für eine Richtlinie nur noch die Standardversion übrig ist, können Sie die übergeordnete Richtlinie im Kontext auswählen (mit der rechten Maustaste klicken) und Löschen wählen, um sie zu löschen.

AWS Lambda Funktionen

Die AWS Toolkit for Visual Studio Code bietet umfassende Unterstützung für AWS Lambda Funktionen, sodass Sie direkt von VS Code aus erstellen, testen und bereitstellen können.

Lambda ist ein vollständig verwalteter, ereignisgesteuerter Rechendienst, der Ihren Code als Reaktion auf Ereignisse von über 200 AWS Diensten und software-as-a-service (SaaS-) Anwendungen automatisch ausführt. Ausführliche Informationen zum AWS Lambda Service finden Sie im Developer Guide. AWS Lambda

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

Themen

- Mit AWS Lambda Funktionen arbeiten

- AWS Lambda Debuggen aus der Ferne
- AWS Lambda console zu IDE

Mit AWS Lambda Funktionen arbeiten

Das AWS Toolkit for Visual Studio Code ermöglicht Ihnen, mit Ihren AWS Lambda Funktionen in Ihrer lokalen VS Code-Umgebung zu arbeiten. Mit dem AWS Toolkit können Sie Ihre Lambda-Funktionen erstellen, bearbeiten, testen, debuggen und bereitstellen, ohne die IDE verlassen zu müssen. Ausführliche Informationen zum AWS Lambda Service finden Sie im Developer Guide. AWS Lambda

In den folgenden Abschnitten wird beschrieben, wie Sie mit der Arbeit mit Lambda-Funktionen in beginnen. AWS Toolkit for Visual Studio Code

> **Note**
>
> Wenn Sie bereits Lambda-Funktionen mit dem erstellt haben AWS Management Console, können Sie diese über das Toolkit aufrufen. Darüber hinaus können Sie Ihre Lambda-Funktionen über das AWS Lambda console in VS Code öffnen. Weitere Informationen finden Sie unter dem Thema AWS Lambda console zu IDE in diesem Benutzerhandbuch. Um eine neue Lambda-Funktion in VS Code zu erstellen, folgen Sie den Schritten, die im Thema Erstellen einer neuen serverlosen Anwendung (lokal) in diesem Benutzerhandbuch beschrieben sind.

Voraussetzungen

Die folgenden Bedingungen müssen erfüllt sein, um mit dem AWS Lambda Dienst im Toolkit arbeiten zu können. AWS

- Die neueste Version von AWS Toolkit for Visual Studio Code wird mit Ihren AWS Anmeldeinformationen installiert und eingerichtet.
- Ihre AWS Identity and Access Management (IAM) verwalteten Berechtigungen und Richtlinien sind so konfiguriert, dass sie mit dem AWS Lambda Service funktionieren. Ausführliche Informationen zur Konfiguration Ihrer Berechtigungen und zur Erstellung einer kompatiblen AWS verwalteten Richtlinie finden Sie im AWS Lambda Thema AWS Identity and Access Management für im AWS Lambda Entwicklerhandbuch.

- Sie verfügen über bestehende AWS Lambda Funktionen oder sind mit der Erstellung einer solchen vertraut. Anweisungen zum Erstellen einer Lambda-Funktion finden Sie im Thema Erstellen Sie Ihre erste Lambda-Funktion im AWS Lambda Entwicklerhandbuch.

Eine Lambda-Funktion aufrufen

Gehen Sie wie folgt vor, um eine Lambda-Funktion von Ihrem AWS Konto in VS Code aufzurufen.

1. Erweitern Sie von der AWS Toolkit for Visual Studio Code aus den AWS Explorer.
2. Erweitern Sie im AWS Explorer Lambda, um Ihre Lambda-Ressourcen anzuzeigen.
3. Öffnen Sie das Kontextmenü für die Lambda-Funktion (klicken Sie mit der rechten Maustaste), die Sie aufrufen möchten, und wählen Sie dann In der Cloud aufrufen oder wählen Sie das Symbol In der Cloud aufrufen, um das Konfigurationsmenü für Fernaufruf in VS Code zu öffnen.
4. Geben Sie im Konfigurationsmenü „Remote Invoke“ Ihre Payload-Einstellungen an und fügen Sie alle zusätzlichen Informationen hinzu, die für das Ereignis erforderlich sind.

   > **Note**
   >
   > Der erste Aufrufprozess wird möglicherweise gestartet, sobald Sie im Explorer Invoke in the Cloud auswählen. AWS Die Ausgabe wird auf der Registerkarte OUTPUT des VS Code-Terminals angezeigt.

5. Wählen Sie die Taste Remote Invoke, um Ihre Funktion aufzurufen. Die Ausgabe wird auf der Registerkarte OUTPUT des VS Code-Terminals angezeigt.

Löschen einer Lambda-Funktion

Gehen Sie wie folgt vor, um eine Lambda-Funktion zu löschen.

> **Warning**
>
> Verwenden Sie dieses Verfahren nicht, um Lambda-Funktionen zu löschen, die mit AWS CloudFormation verknüpft sind. Diese Funktionen müssen über Ihren AWS CloudFormation Stack gelöscht werden.

1. Erweitern Sie von der AWS Toolkit for Visual Studio Code aus den AWS Explorer.

2. Erweitern Sie im AWS Explorer Lambda, um Ihre Lambda-Ressourcen anzuzeigen.
3. Klicken Sie mit der rechten Maustaste auf die Lambda-Funktion, die Sie löschen möchten, und wählen Sie dann Löschen.
4. Bestätigen Sie, dass Sie Ihre Funktion löschen möchten, wenn Sie dazu aufgefordert werden.

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

Laden Sie eine Lambda -Funktion herunter.

Sie können Code von einer Remote-Lambda-Funktion in Ihren VS Code-Arbeitsbereich herunterladen, um ihn zu bearbeiten und zu debuggen.

> **Note**
>
> Um Ihre Lambda-Funktion herunterzuladen, müssen Sie in einem VS Code-Workspace mit einem zugänglichen Ordner arbeiten und das AWS Toolkit unterstützt diese Funktion nur mit Lambda-Funktionen, die Node.js und Python-Laufzeiten verwenden.

1. Erweitern Sie von der aus den AWS Toolkit for Visual Studio Code Explorer. AWS
2. Erweitern Sie im AWS Explorer Lambda, um Ihre Lambda-Ressourcen anzuzeigen.
3. Klicken Sie mit der rechten Maustaste auf die Lambda-Funktion, die Sie herunterladen möchten, und wählen Sie dann Herunterladen.
4. Ihre Lambda-Funktion wird im VS Code-Editor geöffnet und im AWS Explorer angezeigt, wenn der Download abgeschlossen ist. Das AWS Toolkit erstellt auch eine Startkonfiguration im VS Code Run-Panel, mit der Sie die Lambda-Funktion lokal ausführen und debuggen können. AWS Serverless Application Model Weitere Informationen zur Verwendung AWS SAM finden Sie unter. the section called “Eine serverlose Anwendung über eine Vorlage ausführen und debuggen (lokal)”

Bereitstellung von Updates für neue Lambda-Funktionen

Sie können Updates für neue Lambda-Funktionen von einem nicht spezifizierten, temporären Speicherort auf Ihrem lokalen Computer aus bereitstellen.

> **Note**
>
> Wenn es Änderungen an Ihren Lambda-Dateien gibt, die nicht bereitgestellt wurden, werden Sie durch das M-Symbol neben den geänderten Dateien im VS Code-Editor und im Explorer benachrichtigt. AWS

Bereitstellung über den VS Code-Editor

1. Öffnen Sie eine Datei aus Ihrer Lambda-Funktion im VS Code-Editor und nehmen Sie dann eine Änderung an der Datei vor.
2. Manuelles Speichern über das VS Code-Hauptmenü oder durch Drücken von **`option+s`** (Mac) **`ctrl+s`** (Windows).
3. VS Code fordert Sie automatisch auf, Ihre Änderungen in der Cloud bereitzustellen. Wählen Sie die Schaltfläche Bereitstellen, um die Bereitstellung zu bestätigen.
4. VS Code informiert Sie über den Status Ihrer Bereitstellung und benachrichtigt Sie, wenn der Vorgang abgeschlossen ist.

Bereitstellung über den Explorer AWS

1. Öffnen Sie eine Datei aus Ihrer Lambda-Funktion im VS Code-Editor und nehmen Sie dann eine Änderung an der Datei vor.
2. Erweitern Sie im AWS Toolkit den AWS Explorer.
3. Erweitern Sie im AWS Explorer die AWS Region mit der Lambda-Funktion, für die Sie Änderungen bereitstellen möchten.
4. Erweitern Sie AWS in der Region Lambda und navigieren Sie zu der Funktion, für die Sie Änderungen bereitstellen möchten.
5. Wählen Sie im Schnellmenü neben Ihrer Funktion das Symbol Speichern und Bereitstellen Ihres Codes aus.
6. VS Code informiert Sie über den Status Ihrer Bereitstellung und benachrichtigt Sie, wenn der Vorgang abgeschlossen ist.

Updates für bestehende Lambda-Funktionen hochladen

Die folgenden Verfahren beschreiben, wie Sie lokale Änderungen hochladen, die an Ihren vorhandenen Lambda-Funktionen vorgenommen wurden. Diese Funktion unterstützt Uploads mit jeder von Lambda unterstützten Laufzeit.

> ⚠ Warning
>
> Bevor Sie Ihre Lambda-Funktion hochladen, sollten Sie Folgendes beachten:
>
> - Beim Aktualisieren von Code auf diese Weise wird die AWS SAM CLI nicht für die Bereitstellung verwendet oder ein AWS CloudFormation Stack erstellt
> - Das AWS Toolkit validiert den Code nicht. Validieren Sie Ihren Code und testen Sie Ihre Funktion (en), bevor Sie Änderungen in die Cloud hochladen.

Ein Zip-Archiv hochladen

1. Erweitern Sie von AWS Toolkit for Visual Studio Code der aus den AWS Explorer.
2. Erweitern Sie im AWS Explorer Lambda, um Ihre Lambda-Ressourcen anzuzeigen.
3. Klicken Sie mit der rechten Maustaste auf die Lambda-Funktion, in die Sie Ihre Änderungen hochladen möchten, und wählen Sie dann Lambda hochladen... um das Menü „Upload-Typ auswählen“ zu öffnen.
4. Wählen Sie ZIP-Archiv, um es `ZIP Archive` in Ihrem lokalen Verzeichnis zu finden.
5. Wenn Sie dazu aufgefordert werden, bestätigen Sie den Upload, um den Upload der ausgewählten Datei zu start`ZIP Archive`en.
6. Der Status Ihres Uploads wird in VS Code angezeigt und Sie werden benachrichtigt, wenn der Upload-Vorgang abgeschlossen ist.

Ein Verzeichnis hochladen, ohne es zu erstellen

1. Erweitern Sie von AWS Toolkit for Visual Studio Code der aus den AWS Explorer.
2. Erweitern Sie im AWS Explorer Lambda, um Ihre Lambda-Ressourcen anzuzeigen.
3. Klicken Sie mit der rechten Maustaste auf die Lambda-Funktion, in die Sie Ihre Änderungen hochladen möchten, und wählen Sie dann Lambda hochladen... um das Menü „Upload-Typ auswählen“ zu öffnen.

4. Wählen Sie Verzeichnis, um zum Bildschirm Verzeichnis erstellen zu gelangen.
5. Wählen Sie im Fenster Verzeichnis erstellen die Option Nein, um ein lokales Verzeichnis für den Upload auszuwählen.
6. Wenn Sie dazu aufgefordert werden, bestätigen Sie den Upload, um das ausgewählte Verzeichnis hochzuladen.
7. Der Status Ihres Uploads wird in VS Code angezeigt und Sie werden benachrichtigt, wenn der Upload-Vorgang abgeschlossen ist.

Ein Verzeichnis mit einem Build hochladen

> **Note**
>
> Achten Sie auf Folgendes:
>
> - Für dieses Verfahren ist die AWS Serverless Application Model CLI erforderlich.
> - Das AWS Toolkit benachrichtigt Sie vor dem Upload, dass kein passender Handler gefunden werden kann.
> - Um den an Ihre Lambda-Funktion angehängten Handler zu ändern, verwenden Sie den AWS Lambda console oder den AWS Command Line Interface.

1. Erweitern Sie von AWS Toolkit for Visual Studio Code der aus den AWS Explorer.
2. Erweitern Sie im AWS Explorer Lambda, um Ihre Lambda-Ressourcen anzuzeigen.
3. Klicken Sie mit der rechten Maustaste auf die Lambda-Funktion, in die Sie Ihre Änderungen hochladen möchten, und wählen Sie dann Lambda hochladen... um das Menü „Upload-Typ auswählen“ zu öffnen.
4. Wählen Sie Verzeichnis, um zum Bildschirm Verzeichnis erstellen zu gelangen.
5. Wählen Sie im Bildschirm Verzeichnis erstellen die Option Ja und wählen Sie dann ein lokales Verzeichnis für den Upload aus.
6. Wenn Sie dazu aufgefordert werden, bestätigen Sie den Upload, um mit dem Erstellen und Hochladen des ausgewählten Verzeichnisses zu beginnen.
7. Der Status Ihres Uploads wird in VS Code angezeigt und Sie werden benachrichtigt, wenn der Upload-Vorgang abgeschlossen ist.

Konvertieren Sie Ihre Lambda-Funktion in ein Projekt AWS SAM

Gehen Sie wie folgt vor, um Ihre Lambda-Funktion in einen AWS SAM Stack umzuwandeln.

> **Warning**
>
> Derzeit wird bei der Konvertierung einer Lambda-Funktion in ein Projekt nur eine Teilmenge von Ressourcen unterstützt. AWS SAM Um fehlende Ressourcen nach einer Konvertierung zu finden, überprüfen Sie die Lambda-Konsole und fügen Sie sie manuell zu Ihrer AWS SAM Vorlage hinzu. Weitere Informationen zu unterstützten und nicht unterstützten Ressourcen finden Sie im Thema Support für Ressourcentypen im AWS CloudFormation Entwicklerhandbuch.

1. Erweitern Sie im AWS Toolkit den AWS Explorer.
2. Erweitern Sie im AWS Explorer die AWS Region mit der Lambda-Funktion, die Sie in ein AWS SAM Projekt konvertieren möchten.
3. Erweitern Sie AWS in der Region Lambda und navigieren Sie zu der Funktion, die Sie in einen AWS SAM Stack konvertieren möchten.
4. Wählen Sie im Schnellmenü neben Ihrer Lambda-Funktion das Symbol In SAM-Anwendung konvertieren, um Ihr lokales Dateisystem zu durchsuchen und einen Speicherort für Ihr neues AWS SAM Projekt anzugeben.
5. Nachdem Sie einen Speicherort angegeben haben, beginnt das AWS Toolkit mit der Konvertierung Ihrer Lambda-Funktion in ein AWS SAM Projekt. VS Code informiert Sie über den Status des Prozesses.

   > **Note**
   >
   > Dieser Vorgang kann einige Minuten dauern.

6. Wenn Sie von VS Code dazu aufgefordert werden, geben Sie einen Stack-Namen ein und drücken Sie dann die **`Enter`** Taste, um fortzufahren.
7. VS Code informiert Sie weiterhin über den Status Ihres Projekts, benachrichtigt Sie dann, wenn der Vorgang abgeschlossen ist, und öffnet Ihr neues AWS SAM Projekt als VS Code-Arbeitsbereich.

AWS Lambda Debuggen aus der Ferne

Der AWS Toolkit for Visual Studio Code ermöglicht es Ihnen, Ihre AWS Lambda Funktionen, die in der Cloud ausgeführt werden, direkt in VS Code zu debuggen. Mit dem AWS Lambda Remote-Debugging können Sie laufende Funktionen überprüfen, Breakpoints setzen, Variablen untersuchen und das Debuggen Schritt für Schritt durchführen, ohne den bestehenden Entwicklungsablauf zu ändern.

In den folgenden Abschnitten wird beschrieben, wie Sie mit Lambda-Remote-Debugging in der arbeiten. AWS Toolkit for Visual Studio Code

So funktioniert Lambda-Remote-Debugging

Das AWS Toolkit ermöglicht das Remote-Debugging, indem es Ihre Lambda-Funktionen vorübergehend mit einer zusätzlichen Lambda-Debugging-Schicht modifiziert und das Timeout-Limit für Lambda-Aufrufe auf 900 Sekunden verlängert. Mithilfe von Secure Tunneling wird eine sichere Verbindung zwischen Ihrem lokalen Debugger und der Lambda-Laufzeitumgebung hergestellt AWS IoT . Diese Verbindung ermöglicht es Ihnen, Ihre lokalen Code-Breakpoints zu verwenden, um die Funktion bei der Remoteausführung Schritt für Schritt zu durchlaufen. Nach Abschluss Ihrer Debugging-Sitzung werden alle temporären Änderungen automatisch auf ihre ursprünglichen Einstellungen zurückgesetzt.

Erste Schritte

Unterstützte Laufzeiten

Die folgenden Laufzeiten werden vom Lambda-Remote-Debugging unterstützt.

- Python (Amazon Linux 2023)
- Java
- JavaScript/Node.js (Amazon Linux 2023)

Voraussetzungen

Bevor Sie beginnen, müssen die folgenden Voraussetzungen erfüllt sein.

- Sie müssen gültige AWS Anmeldeinformationen im AWS Toolkit konfiguriert haben. Weitere Informationen zur Installation des AWS Toolkits und zur Konfiguration Ihrer Anmeldeinformationen finden Sie im Thema Erste Schritte in diesem Benutzerhandbuch.

- Eine Lambda-Funktion wurde für Ihr AWS Konto bereitgestellt. Einzelheiten zur Bereitstellung einer Lambda-Funktion finden Sie im Thema Erstellen Sie Ihre erste Lambda-Funktion im AWS LambdaEntwicklerhandbuch.
- Sie benötigen die entsprechenden AWS Identity and Access Management (IAM-) Richtlinien und Berechtigungen, um Ihre Funktion debuggen zu können. Weitere Informationen zu Lambda-Berechtigungen finden Sie im AWS Lambda Thema AWS verwaltete Richtlinien im AWS LambdaDeveloper Guide. Im Folgenden finden Sie ein Beispiel für eine Richtlinie, die die Mindestberechtigungen für die Arbeit mit Lambda-Remote-Debugging im AWS Toolkit enthält.

> **Note**
>
> Das Remote-Debugging wird über Secure Tunneling aktiviert. AWS AWS IoT Dadurch kann Ihr lokaler Debugger eine sichere Verbindung zur Lambda-Laufzeitumgebung herstellen.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "lambda:ListFunctions",
        "lambda:GetFunction",
        "lambda:GetFunctionConfiguration",
        "lambda:GetLayerVersion",
        "lambda:UpdateFunctionConfiguration",
        "lambda:InvokeFunction",
        "lambda:PublishVersion",
        "lambda:DeleteFunction",
        "iot:CloseTunnel",
        "iot:OpenTunnel",
        "iot:RotateTunnelAccessToken",
        "iot:ListTunnels"
      ],
      "Resource": "*"
    }
  ]
}
```

Zugreifen auf Lambda-Remote-Debugging

Es gibt zwei Hauptpfade für den Zugriff auf das Lambda-Remote-Debugging im AWS Toolkit: den AWS Explorer oder den Application Builder-Explorer. Vom AWS Explorer aus können Sie über Ihre Knoten auf das Lambda-Remote-Debugging zugreifen AWS Lambda . Über den Application Builder Explorer können Sie über Ihre lokalen AWS SAM Projekte auf das Lambda-Remote-Debugging zugreifen.

Zugriff auf Lambda-Remote-Debugging vom Explorer aus AWS

1. Öffnen Sie in VS Code die AWS Toolkit-Erweiterung.
2. Erweitern Sie im AWS Toolkit den AWS Explorer.
3. Erweitern Sie im Explorer den Lambda-Knoten.
4. Navigieren Sie zu der Funktion, die Sie debuggen möchten, und wählen Sie dann im Kontextmenü das Symbol Invoke remote aus, um den Konfigurationsbildschirm Remote Invoke zu öffnen.

Zugriff auf Lambda-Remote-Debugging über den Application Builder Explorer.

1. Öffnen Sie in VS Code die AWS Toolkit-Erweiterung.
2. Erweitern Sie im AWS Toolkit den Application Builder Explorer.
3. Erweitern Sie im Explorer das `AWS SAM` Projekt, das das Lambda-Projekt enthält, das Sie debuggen möchten.
4. Erweitern Sie die bereitgestellte `Lambda` Funktion, die Sie debuggen möchten.
5. Navigieren Sie zur Funkfernbedienung und wählen Sie dann im Kontextmenü das Symbol Aus der Ferne aufrufen, um den Konfigurationsbildschirm für Fernzugriff zu öffnen.

Arbeiten mit Lambda-Remote-Debugging

In den folgenden Abschnitten wird beschrieben, wie Sie mit Lambda-Remote-Debugging in der arbeiten. AWS Toolkit for Visual Studio Code

> **Note**
>
> Lambda-Funktionen haben ein 5-Layer-Limit und ein kombiniertes Limit von 250 MB für Funktionscode und alle angehängten Schichten. Für die Ausführung von Lambda-Remote-Debugging ist mindestens eine freie Ebene erforderlich.

Eine Debugging-Sitzung einrichten

Bevor Sie beginnen, konfigurieren Sie Ihre Debugging-Sitzung, indem Sie das folgende Verfahren ausführen.

1. Öffnen Sie das Konfigurationsmenü Remote Invoke, indem Sie die im vorherigen Abschnitt beschriebenen Verfahren Accessing Lambda Remote Debug from the AWS Explorer oder Accessing Lambda remote debug from the Application Builder Explorer abschließen.
2. Wählen Sie im Konfigurationsmenü Remote Invoke das Kontrollkästchen Remote Debugging, um die Remote-Debugging-Eigenschaften anzuzeigen.
3. Geben Sie den lokalen Root-Pfad zu Ihrer lokalen Handler-Datei an.

   > **Note**
   >
   > Der lokale Root-Pfad ist der Speicherort Ihres Quellcodes, der der bereitgestellten Lambda-Funktion entspricht. Wenn Sie mit einer bereitgestellten Funktion im Application Builder Explorer arbeiten, wird Ihr lokaler Stammpfad automatisch erkannt.
   > Wenn Sie den Quellcode nicht lokal gespeichert haben, wählen Sie die Schaltfläche Remotecode herunterladen, um den Quellcode Ihrer Lambda-Funktion abzurufen. Dadurch wird Ihr `handler file` VS-Code-Editor geöffnet.

4. Geben Sie im Abschnitt Payload an, wo Ihre Testereignisdaten abgerufen werden.

Breakpoints setzen und debuggen

Legen Sie Haltepunkte fest und beginnen Sie mit dem Debuggen, indem Sie das folgende Verfahren ausführen.

1. Klicken Sie `handler file` im VS Code-Editor auf den Seitenrand, um Haltepunkte an den Zeilennummern zu setzen, in denen Sie das Debuggen unterbrechen möchten.

2. Wenn Sie mit den Breakpoints zufrieden sind, kehren Sie zum Konfigurationsmenü Remote Invake zurück, um zu überprüfen, ob Ihre Einstellungen korrekt konfiguriert sind, und wählen Sie dann die Schaltfläche Remote Invoke, um mit dem Debuggen zu beginnen.
3. Das AWS Toolkit aktualisiert Ihre Lambda-Funktion mit Debugging-Funktionen, richtet einen sicheren Tunnel für die Debugging-Sitzung ein, ruft Ihre Funktion mit der angegebenen Payload auf und unterbricht dann den Prozess, wenn er einen Breakpoint erreicht.
4. Verwenden Sie bei einer Pause am Breakpoint den Bereich AUSFÜHREN UND DEBUGGEN, um Ihre VARIABLEN, CALL STACK und BREAKPOINTS anzuzeigen.

Ihre Funktion aktualisieren und testen

Gehen Sie wie folgt vor, um Ihren Code zu ändern und Änderungen mit einer schnellen Bereitstellung zu testen.

1. Wenn Ihre Debugging-Sitzung aktiv ist, nehmen Sie `handler file` im VS Code-Editor Änderungen an Ihrer vor.
2. Speichern Sie Ihre Änderungen (**`Command+S on macOS`**,**`Ctrl+S on Windows`**)
3. Wenn Sie dazu aufgefordert werden, bestätigen Sie, dass Sie mit der Implementierung Ihrer Änderungen fortfahren möchten. Das AWS Toolkit aktualisiert Ihre Lambda-Funktion mit dem geänderten Code.
4. Fahren Sie mit dem Debuggen und Testen Ihrer Änderungen fort, indem Sie neue Breakpoints setzen und erneut auf die Schaltfläche Remote Invoke klicken.

> **Note**
>
> Alternativ können Sie die Option Debugger anhängen in den VS Code-Debugging-Steuerelementen deaktivieren und die Schaltfläche Fernaufruf wählen, um Ihre Funktion ohne Debuggen auszuführen.

Beenden einer Debugging-Sitzung

Jede der folgenden Optionen beendet Ihre Remote-Debugging-Sitzung und entfernt die Debug-Ebene aus Ihrem Projekt.

- Wählen Sie im Konfigurationsbildschirm für den Remote-Aufruf die Option Debug-Setup entfernen.

- Wählen Sie das Symbol zum Trennen der Verbindung aus den VS Code-Debugging-Steuerelementen.
- Schließen von `handler file` im VS Code-Editor.

> **Note**
>
> Beachten Sie die folgenden Punkte:
>
> - Die Lambda-Debug-Schicht wird nach 60 Sekunden Inaktivität automatisch entfernt. Die Zählung beginnt, wenn Ihr letzter Aufruf abgeschlossen ist.
> - Wenn Sie während des Debugging-Vorgangs Codeänderungen an Ihren infrastructure-as-code (IaC) verwalteten (AWS SAM AWS CDK,, Terraform) Funktionen vorgenommen haben, speichern Sie diese in Ihrem lokalen Projekt und erwägen Sie, Ihr Quellcodeverwaltungs-Repository zu aktualisieren. Nicht gespeicherte Änderungen werden überschrieben, wenn Ihre IaC-Funktion erneut bereitgestellt wird.
> - Wenn Sie temporäre Änderungen nur zu Debugging-Zwecken vorgenommen haben, sollten Sie Ihre Funktion möglicherweise über Ihre Quellcodeverwaltung erneut bereitstellen, um sicherzustellen, dass sie mit Ihrem Produktionscode übereinstimmt.

Problembehandlung und erweiterte Anwendungsfälle

Wenn Ihre Debug-Sitzung fehlschlägt, starten Sie den Fehlerbehebungsprozess, indem Sie die folgenden Schritte ausführen.

1. Aktualisieren Sie das AWS Toolkit auf die neueste Version.
2. Aktualisieren Sie die Webansicht, indem Sie die Webansicht für die Konfiguration des Remote-Aufrufs schließen und erneut öffnen.
3. Starten Sie VS Code neu, indem Sie es vollständig schließen und erneut öffnen.
4. Öffnen Sie die VS Code-Befehlspalette und geben Sie den Befehl ein. Wählen Sie ihn aus**`AWS: Reset Lambda Remote Debugging Snapshot`**, wenn er in den Ergebnissen angezeigt wird, um Ihren Lambda-Remote-Debugging-Snapshot zurückzusetzen.
5. Wenn Sie das Problem nicht beheben können, reichen Sie ein Problem an Issues ein.AWS Toolkit for Visual Studio Code GitHub

Anwendungsfall für Fortgeschrittene: Konfiguration der Codesignatur

Das Remote-Debugging erfordert das Anhängen einer Debug-Ebene an Ihre Lambda-Funktion. Wenn für Ihre Funktion die Codesignaturkonfiguration aktiviert und durchgesetzt wurde, kann das AWS Toolkit die Debug-Ebene nicht automatisch an Ihre Funktion anhängen.

Es gibt zwei Optionen, um das Problem mit der Konfiguration der Codesignatur zu lösen.

- Entfernen Sie vorübergehend die Codesignatur.
- Verwenden Sie eine signierte Debug-Ebene.

Codesignatur vorübergehend entfernen

Aktualisieren Sie die Codesignaturkonfiguration anhand einer Einstellung und aktivieren Sie sie dann wieder`UntrustedArtifactOnDeployment : Warn`, wenn der Debugging-Vorgang `Enforced` abgeschlossen ist.

Weitere Informationen finden Sie in der UpdateCodeSigningConfigReferenz in der API-Referenz.AWS Lambda

Verwenden einer signierten Debug-Ebene

1. Erweitern Sie im AWS Toolkit unter Lambda Remote Debugging den Abschnitt mit der zusätzlichen Konfiguration von Remote Debug.
2. Kopieren Sie im Abschnitt Zusätzliche Konfiguration für Remote-Debug Ihren Region-Layer-ARN aus dem Feld Layer-Override.
3. Verwenden Sie von der aus den folgenden Befehl AWS CLI, um die Layer-Version herunterzuladen`aws lambda get-layer-version-by-arn --arn` **`layer-arn`**, und ersetzen Sie layer-arn durch Ihren Layer-ARN. Detaillierte Anweisungen zum Herunterladen der signierten Debug-Ebene finden Sie in der Referenz get-layer-version-by-arn in der Befehlsreferenz.AWS CLI
4. Signieren Sie den Layer mit Ihrer Codesignatur-Konfiguration und veröffentlichen Sie ihn in Ihrem Konto. Anleitungen zum Signieren und Veröffentlichen finden Sie im Thema Codesignatur für Ihre AWS SAM Anwendung einrichten im AWS Serverless Application Model Entwicklerhandbuch.
5. Nachdem der Layer signiert und in Ihrem Konto veröffentlicht wurde, kehren Sie zum Abschnitt mit der zusätzlichen Konfiguration für das Remote-Debuggen von Lambda Remote Debugging

zurück und geben Sie dann den neuen Layer-ARN in das Feld Layer-Override ein. Wenn der Vorgang abgeschlossen ist, verwendet das Lambda-Remote-Debugging Ihre signierte Ebene anstelle der Standardebene.

Unterstützte -Regionen

Der folgende Fehler tritt auf, wenn eine Region das Remote-Debugging nicht unterstützt.

```
Region ${region} doesn't support remote debugging yet
```

Im Folgenden finden Sie eine Liste der unterstützten Regionen.

- ap-east-1
- ap-northeast-1
- ap-northeast-2
- ap-south-1
- ap-southeast-1
- ap-southeast-2
- ca-central-1
- eu-central-1
- eu-north-1
- eu-west-1
- eu-west-2
- eu-west-3
- me-central-1
- me-south-1
- sa-east-1
- us-east-1
- us-east-2
- us-west-1
- us-west-2

Lambda RequestEntityTooLargeException

Lambda-Funktionen haben ein 5-Layer-Limit und ein kombiniertes Limit von 250 MB für Funktionscode und alle angehängten Schichten. Die Remote-Debugging-Schicht ist ungefähr 40 MB groß, was dazu führen kann, dass Ihre Funktion diese Grenze überschreitet, wenn Sie über ein großes Funktionspaket oder mehrere Ebenen verfügen. Weitere Informationen finden Sie im RequestEntityTooLargeException Themenabschnitt Lambda: InvalidParameterValueException oder im AWS Lambda Developer Guide.

In der folgenden Liste werden Möglichkeiten zur Behebung und Korrektur dieses Fehlers beschrieben.

- Reduzieren Sie die Funktionsgröße: Optimieren Sie Ihren Funktionscode und entfernen Sie unnötige Abhängigkeiten.
- Unbenutzte Ebenen entfernen: Entfernen Sie beim Debuggen vorübergehend nicht benötigte Ebenen.
- Verwenden Sie externe Abhängigkeiten: Verschieben Sie große Abhängigkeiten in externen Speicher wie Amazon S3 und laden Sie sie zur Laufzeit.

Kontingent für sicheres IoT-Tunneling überschritten

Im Folgenden finden Sie ein Beispiel für den Fehler „Tunnelkontingent überschritten“, der auftritt, wenn Sie das Tageslimit für AWS IoT sichere Tunnelverbindungen beim Lambda-Remote-Debugging erreicht haben.

```
Error creating/reusing tunnel: LimitExceededException: Exceeded quota of Lambda
 debugging tunnels
```

AWS IoT Für Secure Tunneling-Verbindungen gelten die folgenden Kontingente:

- Für sicheres IoT-Tunneling auf kostenloser Stufe sind 10 Verbindungen pro Tag vorgesehen.
- Jeder Tunnel unterstützt eine VS Code-Instanz für bis zu 12 Stunden.
- Das Kontingent gilt pro AWS Konto und Tag.

Wenn der AWS IoT Secure Tunneling-Fehler auftritt, warten Sie, bis das Kontingent täglich zurückgesetzt wird, oder wenden Sie sich an den AWS Support, um eine Erhöhung des Kontingentlimits zu beantragen. Kontaktinformationen für den AWS Support finden Sie im Support-

Kontaktportal.AWS Ausführliche Informationen zu AWS IoT Secure Tunneling finden Sie unter dem Thema AWS IoT Secure Tunneling im Developer Guide.AWS IoT

Debuggen von TypeScript Lambda-Funktionen mit Quellzuordnungen

In den folgenden Abschnitten wird beschrieben, wie Sie Ihre TypeScript Lambda-Funktionen mit Quellzuordnungen debuggen.

Voraussetzungen

Um Ihre TypeScript Lambda-Funktionen zu debuggen, müssen die folgenden Voraussetzungen erfüllt sein.

- Sie TypeScript müssen die aktivierte Quellzuordnungsoption erfüllen. Weitere Informationen finden Sie im Thema zur JavaScript Quellzuordnungsunterstützung in der VS Code-Dokumentation.
- Um mit Inline-Quellzuordnungen arbeiten zu können, müssen alle Änderungen in die Cloud übertragen werden.

Konfiguration

Gehen Sie wie folgt vor, um das Lambda-Remote-Debugging für TypeScript Lambda-Funktionen im AWS Toolkit zu konfigurieren.

1. Erweitern Sie im AWS Toolkit den Explorer. AWS
2. Erweitern Sie im Explorer den Lambda-Knoten.
3. Navigieren Sie zu der Funktion TypeScript, für die Sie konfigurieren möchten, und wählen Sie dann das Symbol „Aus der Ferne aufrufen“ aus dem Kontextmenü, um den Konfigurationsbildschirm für Fernzugriff zu öffnen.
4. Aktivieren Sie das Remote-Debugging, indem Sie das Kontrollkästchen Remote-Debugging aktivieren.
5. Konfigurieren Sie Ihren lokalen Root-Pfad, indem Sie auf das Verzeichnis verweisen, das Ihren enthält. `TypeScript handler file`

   > **Note**
   >
   > `TypeScript handler file`Hier legen Sie Ihre Debugging-Haltepunkte fest.

6. Erweitern Sie Remote Debug — Zusätzliche Konfigurationseinstellungen.

7. Aktivieren Sie die Quellenzuordnung, indem Sie das Kontrollkästchen Quellenzuordnung aktivieren.
8. Setzen Sie das Feld Out-Dateien auf das lokale Verzeichnis Ihrer Lambda-Funktionskopie.

   Example

   Wenn `app.js` und drin `app.map` sind`.aws-sam/build/HelloWorldFunction`, legen Sie den Speicherort `/Users/`**`user`**`/`**`project`**`/aws-sam/build/`**`HelloWorldFunction`**`/*` für die Out-Dateien fest.

   > **Note**
   >
   > Der Out-Dateipfad sollte ein absoluter Pfad sein.

9. Wenn Sie mit den Einstellungen zufrieden sind, klicken Sie auf die Schaltfläche Fernzugriff, um mit dem Debuggen Ihrer TypeScript Funktion zu beginnen.

AWS Lambda console zu IDE

Mit AWS Lambda console der To-IDE-Funktion können Sie Ihre AWS Lambda Funktionen von AWS Lambda console in VS Code herunterladen. Wenn Sie mit Ihren Lambda-Funktionen in VS Code arbeiten, haben Sie Zugriff auf andere lokale Entwicklungsoptionen wie AWS Serverless Application Model (AWS SAM) und die. AWS Cloud Development Kit (AWS CDK)

Weitere Informationen zu finden Sie im AWS Lambda Entwicklerhandbuch. AWS Lambda Informationen zu den ersten Schritten mit Ihrer Lambda-Funktion im AWS Toolkit finden Sie im Thema Arbeiten mit AWS Lambda Funktionen in diesem Benutzerhandbuch. In den folgenden Abschnitten wird beschrieben, wie Sie Ihren Workflow von der Lambda-Konsole nach VS Code verschieben. Ausführliche Informationen zum Verschieben Ihrer Lambda-Funktionen von der Lambda-Konsole zu VS Code, einschließlich der ersten Schritte zur Arbeit mit der Lambda-Konsole, finden Sie im Thema Lokales Entwickeln von Lambda-Funktionen mit VS Code im Entwicklerhandbuch.AWS Lambda

Übergang von der Konsole zur lokalen Entwicklung

Gehen Sie wie folgt vor, um eine Lambda-Funktion über die Lambda-Konsole in VS Code zu öffnen:

1. Öffnen Sie in Ihrem Webbrowser die Lambda-Konsole.

2. Wählen Sie in der Lambda-Konsole die Funktion aus, die Sie in VS Code öffnen möchten.
3. Navigieren Sie in der Funktionsansicht zur Registerkarte Codequelle.
4. Wählen Sie auf der Registerkarte Codequelle die Option In VS Code öffnen aus.

Arbeiten mit Ihrer Lambda-Funktion in VS Code

Wenn Ihre Lambda-Funktion in VS Code über die Lambda-Konsole geöffnet wird:

- VS Code wird automatisch auf Ihrem lokalen Computer gestartet.
- Ihre Lambda-Funktion wird als VS Code-Workspace geöffnet.
- Ihr Lambda `handler file` wird im VS Code Editor geöffnet.

> **Note**
>
> Wenn `handler file` im Workspace keine ordnungsgemäß konfigurierte Datei vorhanden ist, wird keine Datei im VS Code-Editor geöffnet.

Wenn Sie Ihre Lambda-Funktion in VS Code über die Lambda-Konsole öffnen, können Sie auf alle vorhandenen AWS Toolkit-Lambda-Funktionen zugreifen, einschließlich der Möglichkeit, Funktionscode mit vollständiger Sprachunterstützung zu bearbeiten, lokale Tests, Remote-Debugging, Bereitstellungsunterstützung und Abhängigkeitsmanagement. Weitere Informationen zu den im AWS Toolkit unterstützten Lambda-Funktionen finden Sie im Inhaltsverzeichnis des AWS LambdaDienstes in diesem Benutzerhandbuch.

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 unter dem AWS Thema Verbindung herstellen 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 ausführen AWS 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 arbeiten AWS 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 Dateien AWS 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 AWS Toolkit-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 arbeiten AWS Toolkit for Visual Studio Code. Einzelheiten zur Arbeit mit Amazon S3 S3-Ressourcen, wie z. B. S3-Buckets AWS 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 kopieren AWS 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 bietet Unterstützung für. AWS Toolkit for Visual Studio Code AWS-Serverless-Anwendung In den folgenden Themen werden die ersten Schritte zum Erstellen von und Arbeiten mit AWS Serverless Application Model (AWS SAM) -Anwendungen anhand von beschrieben AWS Toolkit for Visual Studio Code.

Themen

- Erste Schritte mit serverlosen Anwendungen
- Arbeiten mit AWS Serverless Land
- Lambda-Funktionen direkt aus dem Code ausführen und debuggen
- Ausführen und Debuggen lokaler Amazon API Gateway Ressourcen
- Konfigurationsoptionen für das Debuggen von serverless Anwendungen
- Problembehandlung 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 CodeLensAbschnitt der VS Code-Dokumentation

IAM-Berechtigungen für serverlose Anwendungen

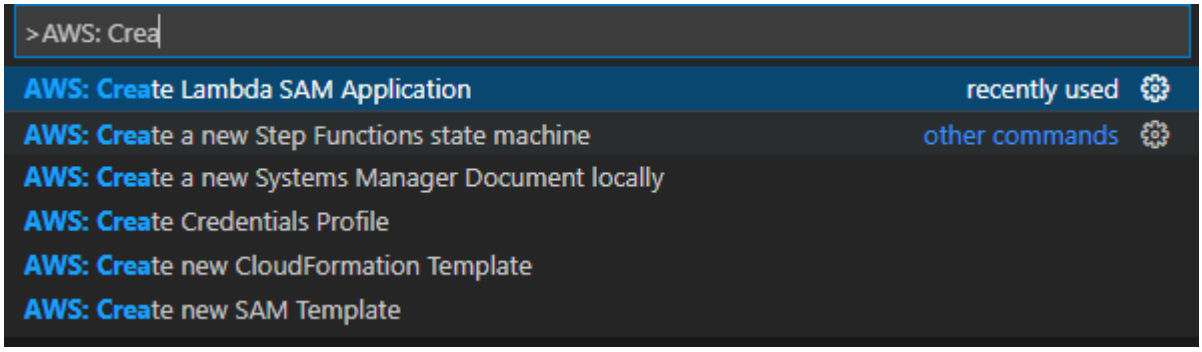
Im Toolkit for VS Code benötigen Sie ein Anmeldeinformationsprofil, das die AWS Identity and Access Management (IAM-) Berechtigungen enthält, die für die Bereitstellung und Ausführung serverloser Anwendungen erforderlich sind. Sie benötigen entsprechenden Lese-/Schreibzugriff auf die folgenden Dienste: 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 AWS Toolkit 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 x86_64-basierten Standardumgebung 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 basierten Umgebungen ausgeführt werden können. ARM64 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-nodejs`. 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 CodeLensCode-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 verwenden`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“.

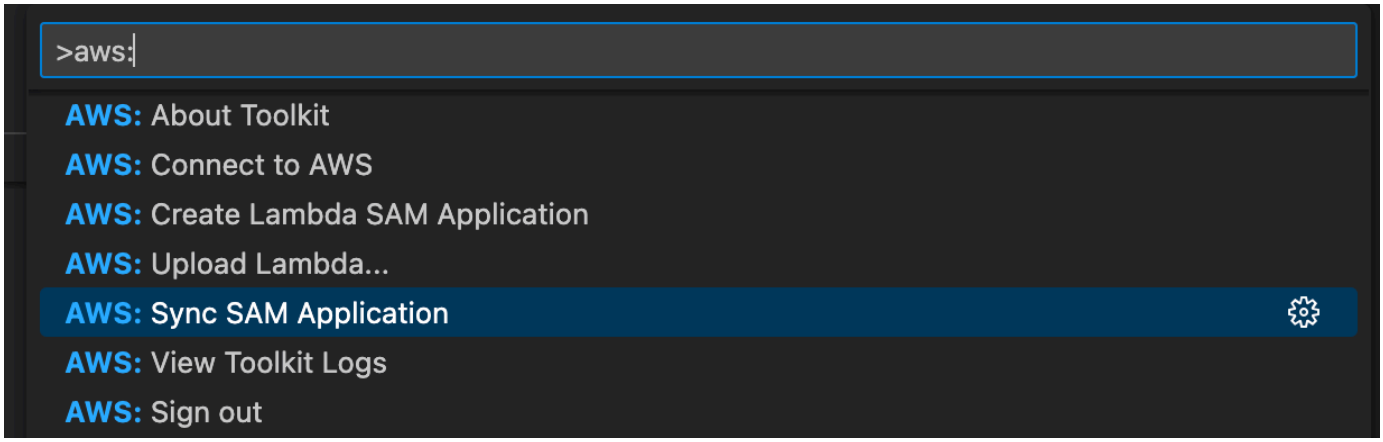
   Wenn die Debugging-Sitzungen gestartet werden, wird die DEBUGKONSOLE zeigt die Debugging-Ausgabe an und zeigt alle von der Lambda Funktion zurückgegebenen Werte an. (Beim Debuggen von AWS SAM Anwendungen wird das AWS Toolkit im Bedienfeld „Ausgabe“ als Ausgabekanal ausgewählt.)

Anwendungen werden synchronisiert AWS SAM

Das AWS Toolkit for Visual Studio Code führt den AWS SAM CLI-Befehl aus`sam 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 AWSund 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.

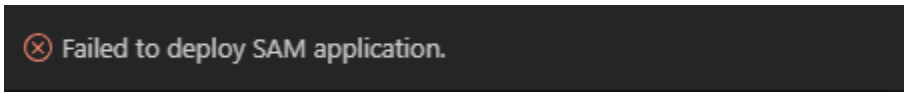
> **Important**
>
> Ihr Amazon S3 S3-Bucket muss die folgenden Anforderungen erfüllen:
>
> - Der Bucket muss sich in der Region befinden, mit der Sie synchronisieren.
> - Der Amazon-S3-Bucket-Name muss unter den in Amazon S3 vorhandenen Bucket-Namen global eindeutig sein.

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.



> **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 AWS Explorer aufgeführt. Weitere Informationen zum Aufrufen der Lambda-Funktion, die als Teil der Anwendung erstellt wurde, finden Sie im Mit AWS Lambda Funktionen arbeiten Thema in diesem Benutzerhandbuch.

Löschen einer serverlosen Anwendung aus dem AWS Cloud

Das Löschen einer serverlosen Anwendung beinhaltet das Löschen des AWS CloudFormation Stacks, den Sie zuvor in der 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 AWS Toolkit-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 .
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.

Arbeiten mit AWS Serverless Land

AWS Serverless Land in the AWS Toolkit for Visual Studio Code ist eine Sammlung von Funktionen, die Sie beim Erstellen ereignisgesteuerter Architekturen unterstützen. In den folgenden Themenabschnitten wird beschrieben, wie Sie mit Serverless Land im Toolkit arbeiten. AWS Ausführliche Informationen zu Serverless Land finden Sie in der Serverless Land-Webanwendung.

Zugreifen auf Serverless Land

Es gibt drei Haupteinstiegspunkte für den Zugriff auf Serverless Land im AWS Toolkit:

- Die VS Code-Befehlspalette
- Der AWS Toolkit-Explorer
- Der AWS Toolkit Application Builder-Explorer

Serverless Land über die VS Code-Befehlspalette öffnen

Gehen Sie wie folgt vor, um Serverless Land über die VS Code Command Palette zu öffnen.

1. Öffnen Sie in VS Code die Befehlspalette, indem Sie **`option+shift+p`** (Mac) oder **`control+shift+p`** (Windows) drücken.
2. Geben Sie **`AWS Create application with Serverless template`** in der VS Code-Befehlspalette in die Suchleiste ein.
3. Wählen Sie AWS: Anwendung mit serverloser Vorlage erstellen, wenn sie in der Liste angezeigt wird.
4. Wenn der Vorgang abgeschlossen ist, öffnet sich der Serverless Land-Assistent und der Bildschirm Wählen Sie ein Muster für Ihre Anwendung (1/5) in VS Code aus.

Serverless Land über den Toolkit Explorer öffnen. AWS

Gehen Sie wie folgt vor, um Serverless Land vom AWS Toolkit Explorer aus zu öffnen.

1. Erweitern Sie im AWS Toolkit-Explorer die Region, in der Sie Serverless Land öffnen möchten.
2. Öffnen Sie das Kontextmenü für den Lambda-Knoten (Rechtsklick).
3. Wählen Sie im Kontextmenü die Option Anwendung mit serverloser Vorlage erstellen aus.
4. Der Serverless Land-Assistent öffnet in VS Code den Bildschirm Select a Pattern for Your Application (1/5), wenn der Vorgang abgeschlossen ist.

Serverless Land vom Application Builder Explorer aus öffnen

Gehen Sie wie folgt vor, um Serverless Land vom AWS Toolkit Application Builder Explorer aus zu öffnen.

1. Navigieren Sie im AWS Toolkit Explorer zum Application Builder Explorer.
2. Klicken Sie mit der rechten Maustaste auf den Application Builder-Explorer und wählen Sie im Kontextmenü die Option Anwendung mit serverloser Vorlage erstellen aus.
3. Wenn der Vorgang abgeschlossen ist, wird der Serverless Land-Assistent in VS Code mit dem Bildschirm „Wählen Sie ein Muster für Ihre Anwendung (1/5)“ geöffnet.

Eine Anwendung mit einer serverlosen Vorlage erstellen

Gehen Sie wie folgt vor, um eine Anwendung mit einer serverlosen Vorlage zu erstellen.

1. Wählen Sie auf dem Bildschirm „Serverless Land Wizard Select a Pattern for your application“ (1/5) ein Muster für die Basis Ihrer Anwendung aus.

   > **Note**
   >
   > Um eine Vorschau und weitere Details zu einem bestimmten Pattern anzuzeigen, klicken Sie auf das Symbol „In Serverless Land öffnen“ neben dem Pattern, das Sie anzeigen möchten. Das Serverless Land Pattern wird in Ihrem Standard-Webbrowser geöffnet.

2. Wählen Sie auf dem Bildschirm Select Runtime (2/5) eine Laufzeit für Ihr Projekt aus.
3. Wählen Sie auf dem Bildschirm Select IaC (3/5) eine IaC-Option für Ihr Projekt aus.
4. Wählen Sie auf dem Bildschirm „Projektstandort auswählen“ (4/5) einen Speicherort für Ihr Projekt aus.
5. Geben Sie auf dem Bildschirm Projektnamen eingeben (5/5) einen Namen für Ihre neue Anwendung ein.
6. Ihre neue Anwendung wird im VS Code Explorer angezeigt und Ihr Projekt `readme.md` wird im VS Code Editor geöffnet, wenn der Vorgang abgeschlossen ist.

   > **Note**
   >
   > Nachdem Ihre neue Anwendung erstellt wurde, finden Sie zusätzliche Aktionen, die für Ihren Anwendungstyp spezifisch sind, in der `readme.md` Datei. Darüber hinaus können

Ihre AWS Serverless Application Model (AWS SAM) -Anwendungen mit AWS Application Builder für lokale Tests, Debugging und mehr geöffnet werden.
Einzelheiten zur Arbeit mit Application Builder im AWS Toolkit finden Sie im Thema Arbeiten mit dem AWS Application Builder Explorer in diesem Benutzerhandbuch.

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 CodeLensFunktion 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)	Die Suchfunktion bietet zudem die folgenden Funktionen: • Es ist eine öffentliche Funktion einer öffentlichen Klasse. • Sie hat einen oder zwei Parameter. Bei zwei Parametern muss der zweite Parameter die `ILambdaContext` Schnittstelle implementieren. • Es hat eine `*.csproj` Datei in seinem übergeordneten Ordner innerhalb des VS Code-Workspace-Ordners. Die Erweiterung ms-dotnettools.csharp (oder jede Erweiterung, die Sprachsymbole für C# bereitstellt) ist installiert und aktiviert.

Sprache/Laufzeit	Kriterien für die Identifizierung von Lambda-Funktionen anhand von Indikatoren CodeLens
JavaScript/(Node.js 12.x, 14.x) TypeScript	Die Suchfunktion bietet zudem die folgenden Funktionen: • 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)	Die Suchfunktion bietet zudem die folgenden Funktionen: • Es ist eine Top-Level-Funktion. • Es hat eine `requirements.txt` Datei in seinem übergeordneten Ordner innerhalb des VS Code-Workspace-Ordners. Die Erweiterung ms-python.python (oder jede Erweiterung, die Sprachsymbole für Python bereitstellt) ist installiert und aktiviert.

Sprache/Laufzeit	Kriterien für die Identifizierung von Lambda-Funktionen anhand von Indikatoren CodeLens

Java (8, 8.al2, 11)	Die Suchfunktion bietet zudem die folgenden Funktionen:
	• Es ist eine öffentliche Funktion einer öffentlichen, nicht abstrakten Klasse.
	• Sie hat einen, zwei oder drei Parameter:
	  • Ein Parameter: Der Parameter kann alles sein.
	  • Zwei Parameter: Parameter müssen a sein `java.io.InputStream` und ein `java.io.OutputStream` ODER, der letzte Parameter muss a sein`com.amazonaws.services.lambda.runtime.Context` .
	  • Drei Parameter: Parameter müssen a `java.io.InputStream` sein `java.io.OutputStream` und der letzte Parameter muss a sein`com.amazonaws.services.lambda.runtime.Context` .
	• Es hat eine `build.gradle` (Gradle) oder `pom.xml` (Maven) -Datei in seinem übergeordneten Ordner innerhalb des VS Code-Workspace-Ordners.
	
	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.
	
	Das vscjava. vscode-java-debugDie Erweiterung (oder eine beliebige Erweiterung, die einen

Sprache/Laufzeit	Kriterien für die Identifizierung von Lambda-Funktionen anhand von Indikatoren CodeLens
	Java-Debugger bereitstellt) ist installiert und aktiviert.
Gehe zu (1.x)	Die Suchfunktion bietet zudem die folgenden Funktionen: • Es ist eine Top-Level-Funktion. • Es benötigt zwischen 0 und 2 Argumente. Bei zwei Argumenten muss das erste Argument implementiere `context.Context` . • Es gibt zwischen 0 und 2 Argumente zurück. Wenn es mehr als 0 Argumente gibt, muss das letzte Argument implementiert werden`error`. • Es hat eine `go.mod` Datei im VS Code Workspace-Ordner. Die Erweiterung golang.go ist installiert, konfiguriert und aktiviert.

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, aus`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 AWS Toolkit im Bedienfeld „Ausgabe“ als Ausgabekanal ausgewählt.)

Ausführen und Debuggen lokaler Amazon API Gateway Ressourcen

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

> **Note**
>
> API Gateway unterstützt zwei Typen APIs: REST und HTTP. Die API-Gateway-Funktion unterstützt jedoch AWS Toolkit for Visual Studio Code nur REST APIs. Manchmal APIs werden HTTP als „API Gateway V2" bezeichnet APIs.

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: Rufen Sie den Handler-Quellcode (.js-, .cs- oder .py-Datei) in Ihrem AWS SAM Projekt auf, bewegen Sie den Mauszeiger über den Lambda-Handler und wählen Sie Debug-Konfiguration hinzufügen. CodeLens Wählen Sie dann im Menü den Eintrag mit der Bezeichnung API-Ereignis aus.
   - Option 2: Bearbeiten `launch.json` und erstellen Sie eine neue Startkonfiguration mit 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. Wählen Sie im VS Code Run-Bereich die Startkonfiguration (`myConfig`im obigen Beispiel benannt) aus.
3. (Optional) Fügen Sie Ihrem Lambda Projektcode Haltepunkte hinzu.
4. Geben Sie F5 ein oder wählen Sie im Bedienfeld „Ausführen“ die Option „Abspielen“.
5. Zeigen Sie im Ausgabefenster die Ergebnisse an.

Konfiguration

Wenn Sie den `invokeTarget.target`Eigenschaftswert`api` 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.

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

Wie AWS Verbindungsinformationen bereitgestellt werden. Weitere Informationen finden Sie in der Tabelle AWS mit den Verbindungseigenschaften („aws“) im Konfigurationsoptionen für das Debuggen von serverless Anwendungen Abschnitt.

sam

Wie die AWS SAM CLI die Anwendung erstellt. Weitere Informationen finden Sie in der AWS SAM-CLI-Eigenschaftstabelle („sam“) im Konfigurationsoptionen für das Debuggen von serverless Anwendungen Abschnitt.

Konfigurationsoptionen für das Debuggen von serverless Anwendungen

Wenn Sie die `launch.json` Datei öffnen, um Debug-Konfigurationen zu bearbeiten, können Sie die VS IntelliSenseCode-Funktion verwenden, um gültige Eigenschaften anzuzeigen und automatisch zu vervollständigen. Um IntelliSense im Editor auszulösen, drücken Sie Strg + Leertaste.

IntelliSense ermöglicht es Ihnen, Eigenschaften für den direkten Aufruf von Lambda-Funktionen oder mit der AWS SAM Vorlage zu finden und zu definieren. Sie können auch Eigenschaften für `"lambda"` (wie die Funktion ausgeführt wird), `"sam"` (wie die AWS SAM CLI die Anwendung erstellt) und `"aws"` (wie AWS Verbindungsinformationen bereitgestellt werden) definieren.

AWS SAM: Direkter Lambda-Handler-Aufruf//Template-basierter Lambda-Aufruf

Property (Eigenschaft)	Description (Beschreibung)
`type`	Gibt an, welche Erweiterung die Startkonfiguration verwaltet . Immer auf eingestellt, `aws-sam` um die AWS SAM CLI zum lokalen Erstellen und Debuggen zu verwenden.
`name`	Gibt einen leserfreundlichen Namen an, der in der Liste Konfiguration des Debuggen Startstarts angezeigt wird.

Property (Eigenschaft)	Description (Beschreibung)
`request`	Gibt den Konfigurationstyp an, der von der angegebenen Erweiterung (`aws-sam`) enthalten. Immer auf `direct-invoke` gesetzt, um die Lambda-Funktion zu starten.

Property (Eigenschaft)	Description (Beschreibung)
`invokeTarget`	Gibt den Eintrittspunkt für den Aufruf der Ressource an. Um die Lambda Funktion direkt aufzurufen, legen Sie Werte für die folgenden`invokeTarget` -Felder: • `target` – Eingestellt auf `code`. • `lambdaHandler` — Der Name des aufzurufenden Lambda -Funktionshandlers. • `projectRoot` — Der Pfad für die Anwendungsdatei, die den Lambda-Funktionshandler enthält. • `architecture` — Prozessorarchitektur der emulierte n Umgebung, in der Ihre lokale SAM-Lambda-Anwendu ng ausgeführt wird. Für bestimmte Laufzeiten können Sie `arm64` statt der Standardarchitektur wählen. `x86_64` Weitere Informationen finden Sie unter Eine neue serverlose Anwendung (lokal) erstellen. Um die Lambda-Ressourcen mit der AWS SAM Vorlage aufzurufen, legen Sie Werte für die folgenden `invokeTarget` Felder fest: • `target` – Eingestellt auf `template`. • `templatePath` — Der Pfad zur AWS SAM Vorlagendatei. • `logicalId` — Der Ressourcenname des `AWS::Lamb da::Function` oder `AWS::Serverless::Function` , der aufgerufen werden soll. Sie finden den Ressourcennamen in der Vorlage im YAML-Format AWS SAM . Beachten Sie, dass die `PackageType: Image` in der AWS SAM Vorlage definierten Funktionen AWS Toolkit implizit als Image-basierte Lambda-Funktionen erkennt. Weitere Informationen finden Sie unter Lambda-Bereitstellungspakete im AWS Lambda Developer Guide.

Lambda (**`"lambda"`**) Eigenschaften

Property (Eigenschaft)	Description (Beschreibung)

`environmentVariables`	Übergibt Betriebsparameter an Ihre Lambda-Funktion. Wenn Sie beispielsweise in einen Amazon S3 S3-Bucket schreiben, konfigurieren Sie den Bucket-Namen als Umgebungsvariable, anstatt den Bucket-Namen fest zu codieren.
	
	Note
	Wenn Sie Umgebungsvariablen für eine serverlose Anwendung angeben, müssen Sie Konfigurationen sowohl zur AWS SAM Vorlage (`template.yaml`) als auch zur `launch.json` Datei hinzufügen.
	Beispiel für die Formatierung einer Umgebungsvariablen in der AWS SAM Vorlage:
	`Resources:`
	`HelloWorldFunction:`
	`Type: AWS::Serverless::Function`
	`Properties:`
	`CodeUri: hello-world/`
	`Handler: app.lambdaHandlerN10`
	`Runtime: nodejs10.x`
	`Environment:`
	`Variables:`
	`SAMPLE1: Default Sample 1 Value`
	
	Beispiel für die Formatierung einer Umgebungsvariablen in der `launch.json` Datei:
	`"environmentVariables": {`
	`"SAMPLE1": "My sample 1 value"`
	`}`

`payload`	Stellt zwei Optionen für die Ereignisnutzlast bereit, die Sie Ihrer Lambda -Funktion als Eingabe bereitstellen.

Property (Eigenschaft)	Description (Beschreibung)
	• `"json"`: JSON-formatierte Schlüssel-Wert-Paare, welche die Ereignisnutzlast definieren. • `"path"`: Ein Pfad zu der Datei, die als Ereignisnutzlast verwendet wird.
`memoryMB`	Gibt Megabyte (MB) an Speicher an, der für die Ausführung einer aufgerufenen Lambda-Funktion bereitgestellt wird.
`runtime`	Gibt die Laufzeit an, die die Lambda-Funktion verwendet . Weitere Informationen finden Sie unter AWS Lambda -L aufzeiten.
`timeoutSec`	Legt die zulässige Zeit in Sekunden fest, bevor die Debug-Sit zung das Timeout überschreitet.

Property (Eigenschaft)	Description (Beschreibung)
`pathMappings`	Gibt an, wo sich der lokale Code im Verhältnis zu seiner Ausführung im Container befindet. Standardmäßig ist das Toolkit for VS Code `localRoot` auf das Code-Stammverzeichnis der Lambda-Funktion im lokalen Workspace und `remoteRoot` auf, das Standardarbeitsverzeichnis für Code`/var/task`, der in Lambda ausgeführt wird, festgelegt. Wenn das Arbeitsverzeichnis im Dockerfile oder mit dem `WorkingDirectory` Parameter in der AWS CloudFormation Template-Datei geändert wird, muss mindestens ein `pathMapping` Eintrag angegeben werden, damit der Debugger erfolgreich lokal gesetzte Breakpoints dem Code zuordnen kann, der im Lambda-Container läuft. Beispiel für die Formatierung in der Datei: `pathMappings` `launch.json` `"pathMappings": [ { "localRoot": " ${workspaceFolder}/sam-app/HelloWorldFunction ", "remoteRoot": " /var/task " } ]` Vorbehalte: • Für Image-basierte Lambda-Funktionen von.NET-Funktionen muss der `remoteRoot` Eintrag das Build-Verzeichnis sein. • Für Node.js-basierte Lambda-Funktionen können Sie nur einen einzigen Pfadzuordnungseintrag angeben.

Das Toolkit for VS Code verwendet die AWS SAM CLI, um serverlose Anwendungen lokal zu erstellen und zu debuggen. Sie können das Verhalten von AWS SAM CLI-Befehlen mithilfe der Eigenschaften der `"sam"` Konfiguration in der `launch.json` Datei konfigurieren.

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

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

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

AWS Eigenschaften von connection (**`"aws"`**)

Property (Eigenschaft)	Description (Beschreibung)	Standardwert
`credentials`	Wählt ein bestimmtes Profil (z. B.`profile:default`) aus Ihrer Anmeldein formationsdatei aus, um AWS Anmeldeinformationen abzurufen.	Die AWS Anmeldeinformation en, die Ihre vorhandene gemeinsam genutzte AWS Konfigurationsdatei oder Datei mit gemeinsam genutzten AWS Anmeldeinformationen für das Toolkit for VS Code bereitstellt.
`region`	Legt die AWS Region des Dienstes fest (z. B. us-east-1).	Die AWS Standardregion, die dem aktiven Anmeldein formationsprofil zugeordnet ist.

Beispiel: Konfiguration zum Starten der Vorlage

Hier ist ein Beispiel für eine Startkonfigurationsdatei für ein 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: Konfiguration für den Start von Code

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": {}
            }
        }
    ]
}
```

Problembehandlung bei serverlosen Anwendungen

In diesem Thema werden häufig auftretende Fehler beschrieben, die beim Erstellen serverloser Anwendungen mit dem Toolkit for VS Code auftreten können, sowie deren Behebung.

Themen

- Wie kann ich eine Datei samconfig.toml mit einer SAM-Startkonfiguration verwenden?
- Fehler: "RuntimeError: Container existiert nicht“
- Fehler: „docker.errors. APIError: 500 Serverfehler... Sie haben Ihr Pull-Rate-Limit erreicht.“
- Fehler: „500-Serverfehler: Mounting C:\Users\...“
- Bei Verwendung von WSL sind Webansichten (z. B. das Formular „Invoke on AWS“) fehlerhaft
- Debuggen einer TypeScript Anwendung, aber Breakpoints funktionieren nicht

Wie kann ich eine Datei samconfig.toml mit einer SAM-Startkonfiguration verwenden?

Geben Sie den Speicherort Ihrer SAM-CLI samconfig.toml an, indem Sie das `--config-file` Argument in der `sam.localArguments` Eigenschaft Ihrer Startkonfiguration konfigurieren. Wenn sich die Datei samconfig.toml beispielsweise auf der obersten Ebene Ihres Workspace befindet:

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

Fehler: "RuntimeError: Container existiert nicht“

Der `sam build` Befehl kann diesen Fehler anzeigen, wenn Ihr System nicht über genügend Speicherplatz für den Docker-Container verfügt. Wenn auf Ihrem Systemspeicher nur 1 bis 2 GB Speicherplatz verfügbar sind, schlägt `sam build` dies möglicherweise während der Verarbeitung fehl, auch wenn der Systemspeicher vor dem Start des Builds nicht vollständig voll ist. Weitere Informationen finden Sie in dieser GitHub Ausgabe.

Fehler: „docker.errors. APIError: 500 Serverfehler... Sie haben Ihr Pull-Rate-Limit erreicht.“

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

```
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-Dienst mit Ihren Docker Hub-Anmeldeinformationen authentifiziert wurde.

Fehler: „500-Serverfehler: Mounting C:\Users\..."

Windows-Benutzern wird beim Debuggen AWS SAM von Anwendungen möglicherweise dieser Docker-Mount-Fehler angezeigt:

```
Fetching lambci/lambda:nodejs10.x Docker container image......
2019-07-12 13:36:58 Mounting C:\Users\<username>\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 ...
```

Versuchen Sie, die Anmeldeinformationen für Ihre gemeinsam genutzten Laufwerke (in den Docker-Einstellungen) zu aktualisieren.

Bei Verwendung von WSL sind Webansichten (z. B. das Formular „Invoke on AWS") fehlerhaft

Dies ist ein bekanntes VS-Code-Problem für Benutzer von Cisco VPN. Weitere Informationen finden Sie in diesem GitHub Problem.

In diesem WSL-Tracking-Problem wird eine Problemumgehung vorgeschlagen.

Debuggen einer TypeScript Anwendung, aber Breakpoints funktionieren nicht

Dies passiert, wenn es keine Quellzuweisung gibt, um die kompilierte JavaScript Datei mit der TypeScript Quelldatei zu verknüpfen. Um dies zu korrigieren, öffnen Sie Ihre `tsconfig.json` Datei und stellen Sie sicher, dass die folgende Option und der folgende Wert festgelegt sind:`"inlineSourceMap": true`.

Arbeiten mit Systems Manager Automation-Dokumenten

AWS Systems Manager bietet Ihnen Transparenz und Kontrolle über Ihre Infrastruktur AWS. Systems Manager bietet eine einheitliche Benutzeroberfläche, mit der Sie Betriebsdaten von mehreren AWS Services anzeigen und betriebliche Aufgaben AWS ressourcenübergreifend automatisieren können.

Ein Systems-Manager-Dokument definiert die Aktionen, die Systems Manager auf Ihren verwalteten Instances durchführt. Ein Automatisierungsdokument ist eine Art von Systems Manager Manager-Dokument, das Sie für allgemeine Wartungs- und Bereitstellungsaufgaben wie das Erstellen oder Aktualisieren eines Amazon Machine Image (AMI) verwenden. In diesem Thema wird beschrieben, wie Sie Automatisierungsdokumente mit erstellen, bearbeiten, veröffentlichen und löschen AWS Toolkit for Visual Studio Code.

Themen

- Annahmen und Voraussetzungen
- IAM-Berechtigungen für Dokumente zur Automatisierung von Systems Manager
- Ein neues Systems Manager Automation-Dokument erstellen
- Ein vorhandenes Systems Manager Automation-Dokument öffnen
- Bearbeiten eines Systems Manager Automation-Dokuments
- Veröffentlichen eines Systems Manager Automation-Dokuments
- Löschen eines Systems Manager Automation-Dokuments
- Ausführen eines Systems Manager Automation-Dokuments
- Problembehandlung bei Systems Manager Automation-Dokumenten im Toolkit for VS Code

Annahmen und Voraussetzungen

Bevor Sie beginnen, stellen Sie Folgendes sicher:

- Sie haben Visual Studio Code und die neueste Version von installiert AWS Toolkit for Visual Studio Code. Weitere Informationen finden Sie unter Installation des AWS Toolkit for Visual Studio Code.
- Sie sind mit Systems Manager vertraut. Weitere Informationen finden Sie im AWS Systems Manager -Benutzerhandbuch.
- Sie kennen die Anwendungsfälle von Systems Manager Automation. 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 benötigen Sie ein Anmeldeinformationsprofil, das die AWS Identity and Access Management (IAM-) Berechtigungen enthält, die zum Erstellen, Bearbeiten, Veröffentlichen

und Löschen 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 Anmeldeinformationsprofils finden Sie unter. AWS IAM-Anmeldeinformationen

Ein neues Systems Manager Automation-Dokument erstellen

Sie können ein neues Automatisierungsdokument in `JSON` oder `YAML` mit Visual Studio Code erstellen. Wenn Sie ein neues Automatisierungsdokument erstellen, wird es in einer Datei ohne Titel angezeigt. Sie können Ihre Datei benennen und in VS Code speichern, der Name der Datei ist jedoch für sie nicht sichtbar. AWS

Um ein neues Automation-Dokument zu erstellen

1. Öffnen Sie VS Code.
2. Wählen Sie im Menü Ansicht die Option Befehlspalette, um die Befehlspalette zu öffnen.
3. Geben Sie in der Befehlspalette AWS Toolkit Ein neues Systems Manager Manager-Dokument lokal erstellen ein.

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 Automation-Dokument in VS Code erscheint nicht automatisch in AWS. Sie müssen es veröffentlichen, AWS bevor Sie es ausführen können.

Ein vorhandenes Systems Manager Automation-Dokument öffnen

Sie verwenden den AWS Explorer, um nach vorhandenen Systems Manager Automation-Dokumenten zu suchen. Wenn Sie ein vorhandenes Automation-Dokument öffnen, wird es in VS Code als Datei ohne Titel angezeigt.

Um Ihr Automation-Dokument zu öffnen

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

> **Note**
>
> Wenn Sie ein Automation-Dokument lokal als Datei in VS Code speichern, wird es nicht in AWS angezeigt. Es muss AWS vor der Ausführung veröffentlicht werden.

Bearbeiten eines Systems Manager Automation-Dokuments

Wenn Sie Eigentümer von Automatisierungsdokumenten sind, werden diese in der Kategorie „Mein Eigentum“ der Systems Manager Manager-Dokumente im AWS Explorer angezeigt. Sie können

Eigentümer von Automatisierungsdokumenten sein, die bereits in VS Code vorhanden sind AWS, und Sie können Eigentümer neuer oder aktualisierter Dokumente sein, in denen Sie zuvor in VS Code veröffentlicht haben. AWS

Wenn Sie ein Automation-Dokument zur Bearbeitung in VS Code öffnen, können Sie damit mehr tun als in der AWS Management Console. Zum Beispiel:

- Es gibt eine Schemavalidierung für beide `JSON` `YAML` Formate.
- Im Dokumenteditor stehen Auszüge zur Verfügung, mit denen Sie beliebige Arten von Automatisierungsschritten erstellen können.
- Es gibt Unterstützung für die automatische Vervollständigung verschiedener Optionen in `JSON` und`YAML`.

Arbeiten mit Versionen

Systems Manager Automation-Dokumente verwenden Versionen für das Änderungsmanagement. Sie können die Standardversion für ein Automation-Dokument 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 Standardversion festlegen.

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

Veröffentlichen eines Systems Manager Automation-Dokuments

Nachdem Sie Ihr Automation-Dokument in VS Code bearbeitet haben, können Sie es in veröffentlichen AWS.

Um Ihr Automation-Dokument zu veröffentlichen

1. Öffnen Sie das Automatisierungsdokument, das Sie veröffentlichen möchten, wie unter beschriebenEin vorhandenes Systems Manager Automation-Dokument öffnen.

2. Nehmen Sie die Änderungen vor, die veröffentlicht werden sollen. Weitere Informationen finden Sie unter Bearbeiten eines Systems Manager Automation-Dokuments.
3. Wählen Sie oben rechts in der geöffneten Datei das Upload-Symbol.
4. Wählen Sie im Dialogfeld für den Veröffentlichungsworkflow die AWS Region aus, in der Sie das Automatisierungsdokument veröffentlichen möchten.
5. Wenn Sie ein neues Dokument veröffentlichen, wählen Sie Quick Create. Andernfalls wählen Sie Quick Update, um ein vorhandenes Automation-Dokument in dieser AWS Region zu aktualisieren.
6. Geben Sie den Namen für dieses Automatisierungsdokument ein.

Wenn Sie ein Update für ein vorhandenes Automatisierungsdokument 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, werden die AWS Ressourcen, die beim Start erstellt oder geändert wurden, nicht gelöscht.

Um Ihr Automation-Dokument zu löschen

1. Öffnen Sie VS Code.
2. Wählen Sie in der linken Navigationsleiste, ob Sie AWSden AWS Explorer öffnen möchten.
3. Öffnen Sie im AWS Explorer für Systems Manager das Kontextmenü (Rechtsklick) für das Dokument, das Sie löschen möchten, und wählen Sie Dokument löschen.

Ausführen eines Systems Manager Automation-Dokuments

Sobald Ihr Automation-Dokument veröffentlicht wurde AWS, können Sie es ausführen, um Aufgaben in Ihrem Namen in Ihrem AWS Konto auszuführen. Um Ihr Automatisierungsdokument auszuführen,

verwenden Sie den AWS Management Console, den Systems Manager APIs AWS CLI, den oder den AWS -Tools für PowerShell. Anweisungen zum Ausführen eines Automatisierungsdokuments finden Sie unter Ausführen einer einfachen Automatisierung im AWS Systems Manager Benutzerhandbuch.

Wenn Sie alternativ eines der AWS SDKs mit dem Systems Manager APIs zum Ausführen Ihres Automatisierungsdokuments verwenden möchten, finden Sie weitere Informationen in den AWS SDK-Referenzen.

> **Note**
>
> Durch die Ausführung eines Automatisierungsdokuments können neue Ressourcen erstellt werden AWS und es können Abrechnungskosten anfallen. Wir empfehlen Ihnen dringend, dass Sie wissen, was Ihr Automatisierungsdokument in Ihrem Konto erstellen wird, bevor Sie es starten.

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

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

Beim Speichern eines Automation-Dokuments in VS Code wird das Automation-Dokument nicht veröffentlicht AWS. Weitere Informationen zum Veröffentlichen Ihres Automatisierungsdokuments finden Sie unter Veröffentlichen eines Systems Manager Automation-Dokuments.

Das Veröffentlichen meines Automation-Dokuments ist mit einem Berechtigungsfehler fehlgeschlagen.

Stellen Sie sicher, dass Ihr AWS Anmeldeinformationsprofil über die erforderlichen Berechtigungen zum Veröffentlichen von Automatisierungsdokumenten verfügt. Ein Beispiel für eine Berechtigungsrichtlinie finden Sie unter IAM-Berechtigungen für Dokumente zur Automatisierung von Systems Manager.

Ich habe mein Automation-Dokument veröffentlicht AWS, aber ich sehe es nicht in der AWS Management Console.

Vergewissern Sie sich, dass Sie das Dokument in derselben AWS Region veröffentlicht haben, in der Sie gerade surfen AWS Management Console.

Ich habe mein Automation-Dokument gelöscht, aber mir werden immer noch die Ressourcen in Rechnung gestellt, die es erstellt hat.

Durch das Löschen eines Automatisierungsdokuments werden die Ressourcen, die es erstellt oder geändert hat, nicht gelöscht. Sie können die AWS Ressourcen, die Sie erstellt haben, in der AWS Billing Management Console identifizieren, Ihre Gebühren überprüfen und dort auswählen, welche Ressourcen gelöscht werden sollen.

AWS Step Functions

Mit können Sie Workflows (auch Zustandsmaschinen genannt) erstellen AWS Step Functions, um verteilte Anwendungen zu erstellen, Prozesse zu automatisieren, Microservices zu orchestrieren und Daten- und Machine-Learning-Pipelines zu erstellen. In den folgenden Themen wird beschrieben, wie Sie mit dem AWS Step Functions arbeiten. AWS Toolkit for Visual Studio Code Ausführliche Informationen zum AWS Step Functions Dienst finden Sie im AWS Step FunctionsEntwicklerhandbuch.

Themen

- Arbeiten mit AWS Step Functions
- Mit AWS Step Functions Workflow Studio arbeiten

Arbeiten mit AWS Step Functions

In den folgenden Abschnitten wird beschrieben, wie Sie mit AWS Step Functions `Amazon State Language (ASL)` Dateien arbeiten, die State-Machine-Definitionen im AWS Toolkit enthalten. Ausführliche Informationen zu AWS Step Functions Zustandsmaschinen finden Sie im Thema Erfahren Sie mehr über Zustandsmaschinen in Step Functions im AWS Step FunctionsEntwicklerhandbuch.

Zustandsmaschinen von Step Functions anzeigen

Gehen Sie wie folgt vor, um Ihre vorhandenen `ASL` Dateien mit State-Machine-Definitionen im AWS Toolkit-Explorer anzuzeigen.

1. Erweitern Sie im AWS Toolkit-Explorer den Bereich, der die `ASL` Datei enthält, die Sie anzeigen möchten.
2. Erweitern Sie die Überschrift Step Functions.

3. Ihre ASL Dateien werden im AWS Explorer angezeigt.

Eine Step Functions Functions-Zustandsmaschine erstellen

Im AWS Toolkit können Sie eine neue Step Functions Functions-Zustandsmaschine aus einer Datei erstellen oder eine Vorlage verwenden. Das folgende Verfahren beschreibt, wie Sie eine Step Functions Functions-Zustandsmaschine aus einer Datei erstellen. Einzelheiten zum Erstellen eines SFN; -Zustandsmaschinen anhand einer Vorlage finden Sie im Abschnitt Vorlagen für Zustandsmaschinen weiter unten in diesem Thema im Benutzerhandbuch.

> **Note**
>
> Um mit Step Functions in VS Code arbeiten zu können, muss die Erweiterung Ihrer `Amazon State Language` (ASL) -Datei, die Ihre State Machine-Definition enthält`asl.json`, mit`asl.yml`, oder `.asl.yaml` enden.
> Standardmäßig werden relevante Step Functions Functions-Dateien in Workflow Studio geöffnet. Ausführliche Informationen zur Arbeit in Workflow Studio mithilfe des AWS Toolkits finden Sie im Thema Arbeiten mit Workflow Studio in diesem Benutzerhandbuch.

1. Erstellen Sie in Ihrem Workspace in VS Code eine neue Datei.
2. Benennen Sie Ihre Datei und geben Sie die Dateierweiterung als `asl.json`asl.yml, oder an`.asl.yaml`.
3. Nach der Erstellung öffnet das AWS Toolkit die neue Datei in AWS Step Functions Workflow Studio.
4. Wählen Sie in Workflow Studio im Utility-Menü die Schaltfläche Speichern, um Ihre neue ASL Datei zu speichern.

Eine Step Functions Functions-Zustandsmaschine aus einer Vorlage erstellen

Im AWS Toolkit können Sie eine Step Functions Functions-Zustandsmaschine aus einer Vorlage erstellen. Der Vorlagenprozess erstellt eine ASL Datei, die eine State-Machine-Definition enthält und so einen Ausgangspunkt für Ihr Projekt darstellt. Das folgende Verfahren beschreibt, wie Sie eine Step Functions Functions-Zustandsmaschine aus einer Vorlage im AWS Toolkit erstellen.

1. Erweitern Sie im AWS Toolkit-Explorer die Region, in der Sie eine Step Functions Functions-Zustandsmaschine erstellen möchten.

2. Öffnen Sie das Kontextmenü für Step Functions (mit der rechten Maustaste) und wählen Sie Neue Step Functions Functions-Zustandsmaschine erstellen, um den Assistenten zum Auswählen einer Startvorlage (1/2) in VS Code zu öffnen.
3. Wählen Sie im Assistenten Startvorlage auswählen (1/2) den Vorlagentyp für Ihre Step Functions Functions-Zustandsmaschine aus, um fortzufahren.
4. Wählen Sie auf dem Bildschirm Vorlagenformat auswählen (2/2) entweder YAML oder JSON als Vorlagenformat aus.
5. Eine neue ASL Datei, die Ihre State Machine-Definition enthält, wird im VS Code Editor geöffnet.

Eine Step Functions Functions-Zustandsmaschine herunterladen

Gehen Sie wie folgt vor, um eine remote gespeicherte Step Functions Functions-Zustandsmaschine auf Ihre lokale Instanz von VS Code herunterzuladen.

1. Erweitern Sie im AWS Toolkit-Explorer die Region, die den Step Functions Functions-Zustandsmaschine enthält, den Sie herunterladen möchten.
2. Erweitern Sie Step Functions, klicken Sie dann mit der rechten Maustaste auf die Step Functions Functions-Zustandsmaschine, die Sie herunterladen möchten, und wählen Sie Definition herunterladen... .
3. Geben Sie einen Ort an, an dem Ihre Step Functions Functions-Zustandsmaschine lokal gespeichert werden soll, um fortzufahren.
4. Die Step Functions Functions-Zustandsmaschine wird in Workflow Studio geöffnet, wenn der Vorgang abgeschlossen ist.

Änderungen an einer Step Functions Functions-Zustandsmaschine speichern

Das folgende Verfahren beschreibt, wie Sie Änderungen speichern, die Sie an Ihrer Step Functions Functions-Zustandsmaschine vorgenommen haben.

> **Note**
>
> In Workflow Studio vorgenommene Änderungen werden mit Ihrer lokalen Datei synchronisiert, bleiben jedoch nicht gespeichert, bis Ihre Arbeit im VS Code-Editor oder Workflow Studio gespeichert wird. Wenn Ihre lokale Datei geändert und gespeichert wird, während Workflow Studio geöffnet ist und in Ihrer ASL Datei keine Fehler festgestellt wurden, erhalten Sie in Workflow Studio eine Erfolgsmeldung, wenn der Speichervorgang

abgeschlossen ist. Wenn Ihre lokale Datei jedoch ungültiges `JSON` oder enthält `YAML` und Sie versuchen zu speichern, schlägt die Synchronisierung Ihrer lokalen Datei fehl und Sie erhalten eine Warnmeldung in Workflow Studio.

1. Navigieren Sie in einer geöffneten `ASL` Datei, die eine State-Machine-Definition in Workflow Studio enthält, zu den Utility-Schaltflächen.
2. Wählen Sie die Schaltfläche Save (Speichern) aus.
3. VS Code benachrichtigt Sie, wenn die Datei gespeichert wurde.

Ausführen einer Step Functions Functions-Zustandsmaschine

Das folgende Verfahren beschreibt, wie Sie eine Step Functions Functions-Zustandsmaschine im AWS Toolkit ausführen.

1. Erweitern Sie im AWS Toolkit-Explorer die Region, die den Step Functions Functions-Zustandsmaschine enthält, den Sie ausführen möchten.
2. Erweitern Sie Step Functions und klicken Sie dann mit der rechten Maustaste auf die Step Functions Functions-Zustandsmaschine, die Sie ausführen möchten.
3. Wählen Sie im Kontextmenü die Option Ausführung starten, um den Startvorgang zu starten.
4. Der Status des Starts wird im AWS Toolkit-Ausgabefenster in VS Code angezeigt.

Mit Codefragmenten arbeiten

Codefragmente sind automatisierte Vorschläge, die auf der Grundlage des Codes generiert werden, an dem Sie gerade arbeiten. Gehen Sie wie folgt vor, um mit Codefragmenten mit Step Functions im Toolkit zu arbeiten.

> **Note**
>
> Um mit Step Functions Functions-Codefragmenten in VS Code arbeiten zu können, muss die Erweiterung Ihrer `ASL` Datei, die Ihre State Machine-Definition enthält `.asl.json`, mit `.asl.yml`, oder enden. `.asl.yaml`
> Standardmäßig werden Ihre relevanten Step Functions Functions-Dateien in Workflow Studio geöffnet.

1. Öffnen Sie in VS Code eine ASL Datei mit der State-Machine-Definition, die Sie ändern möchten, oder erstellen Sie eine neue ASL Datei.
2. Wechseln Sie in Workflow Studio in den Codemodus, wenn Sie sich im Entwurfsmodus befinden.
3. Platzieren Sie im Workflow Studio-Code-Editor den Cursor auf der `"States"` Eigenschaft.
4. Drücken Sie**`control + space`**, um das Menü mit den Codefragmenten zu öffnen. Weitere Eigenschaften können durch Drücken aufgerufen werden **`control + space`** und basieren auf dem. `"State"` `"Type"`
5. Wählen Sie den gewünschten Codeausschnitt aus der Liste aus.

Code-Validierung

Während Sie in Workflow Studio an Step Functions arbeiten, identifiziert die Codevalidierung aktiv Fehler und macht Vorschläge für Folgendes:

- Fehlende Eigenschaften
- Inkorrekte Werte
- Nicht terminaler Zustand
- Nicht vorhandene Zustände, auf die verwiesen wird

Mit AWS Step Functions Workflow Studio arbeiten

In den folgenden Abschnitten wird beschrieben, wie Sie mit AWS Step Functions Workflow Studio in der arbeiten AWS Toolkit for Visual Studio Code. Ausführliche Informationen zu AWS Step Functions Workflow Studio finden Sie unter dem Thema Entwickeln von Workflows im AWS Step FunctionsEntwicklerhandbuch

Workflow Studio öffnen

In der folgenden Liste werden die verschiedenen Pfade beschrieben, die Ihnen zum Öffnen von Workflow Studio in VS Code zur Verfügung stehen.

> **Note**
>
> Um mit Workflow Studio in VS Code arbeiten zu können, muss die Erweiterung Ihrer `Amazon State Language` (ASL) -Datei, die Ihre Zustandsmaschinen-Definition enthält`asl.json`,

mit `asl.yml` oder `asl.yaml` enden. Einzelheiten zum Herunterladen oder Erstellen einer neuen Zustandsmaschinen-Definition im AWS Toolkit finden Sie in den Abschnitten Herunterladen von Zustandsmaschinen und Erstellen einer Zustandsmaschine im AWS Step Functions Thema Arbeiten mit dieses Benutzerhandbuchs.

- Öffnen Sie im AWS Explorer das Kontextmenü für eine ASL Datei, die eine Zustandsmaschinen-Definition enthält (klicken Sie mit der rechten Maustaste darauf), und wählen Sie dann In Workflow Studio öffnen aus.
- Wählen Sie in einer geöffneten ASL Datei, die eine State-Machine-Definition enthält, das Symbol Mit Workflow Studio öffnen, das sich neben den Registerkarten im VS Code-Editor-Fenster befindet.
- Wählen Sie in einer geöffneten ASL Datei, die eine State-Machine-Definition enthält, den CodeLens Befehl Mit Workflow Studio öffnen, der sich oben in der Datei befindet.
- Durch Schließen und erneutes Öffnen einer ASL Datei, die eine State-Machine-Definition enthält, wird die Datei automatisch wieder in Workflow Studio geöffnet, sofern das standardmäßige Workflow Studio nicht manuell deaktiviert wird.

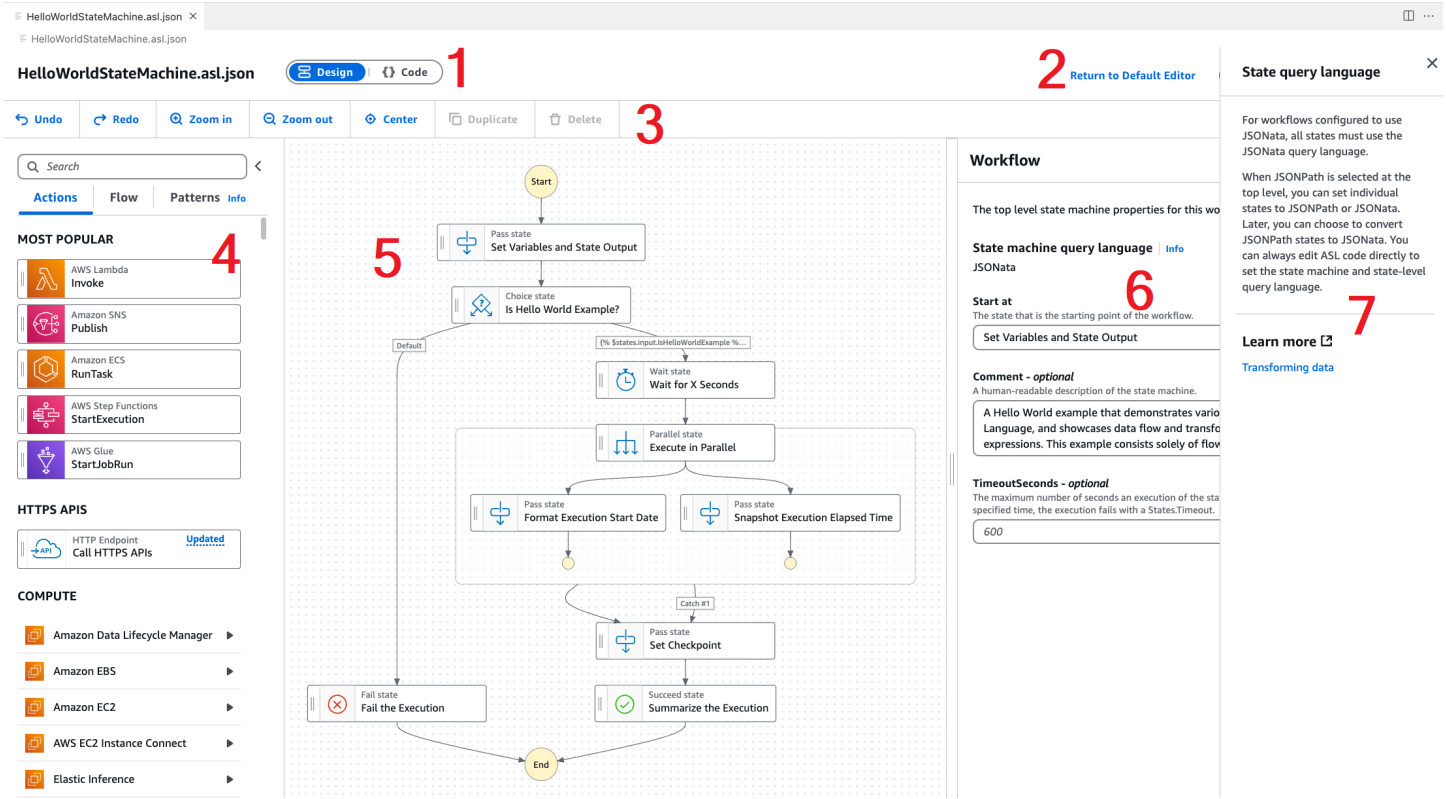
Entwurfsmodus und Codemodus

Workflow Studio bietet zwei Modi für die Arbeit mit Ihren ASL Dateien, die eine Zustandsmaschinen-Definition enthalten: den Entwurfsmodus und den Codemodus. Der Entwurfsmodus bietet eine grafische Oberfläche zur Visualisierung Ihrer Workflows beim Erstellen von Prototypen. Der Codemodus verfügt über einen integrierten Code-Editor, in dem Sie die ASL Definitionen in Ihren Workflows anzeigen, schreiben und bearbeiten können.

> **Note**
>
> Ausführliche Informationen zu den einzelnen Abschnitten der Benutzeroberfläche sowohl im Entwurfs- als auch im Codemodus finden Sie im Thema Verwenden von Workflow Studio im AWS Step FunctionsEntwicklerhandbuch. Nicht alle Funktionen von Workflow Studio sind im AWS Toolkit verfügbar, wie z. B. der Konfigurationsmodus.

Die Benutzeroberfläche des Designmodus besteht aus 7 Hauptbereichen, die in der folgenden Abbildung beschrieben und beschrieben sind.

1. Modusschaltflächen: Schaltflächen zum Umschalten zwischen Design - und Codemodus.
2. Hilfsschaltflächen: Eine Reihe von Schaltflächen zum Ausführen von Aufgaben, z. B. zum Beenden von Workflow Studio, zum Speichern Ihrer Workflows oder zum Exportieren ASL von Definitionen in eine JSON- oder YAML-Datei.
3. Design-Symbolleiste: Werkzeugleiste mit einer Reihe von Schaltflächen, mit denen allgemeine Aktionen wie Rückgängigmachen, Löschen und Zoom-Steuerung ausgeführt werden können.
4. Statusbrowser: Browser, der drag-and-drop Status für Ihre Workflow-Arbeitsfläche enthält. Status sind in Registerkarten organisiert und als Aktionen, Ablauf und Muster definiert.
5. Die Arbeitsfläche und das Workflow-Diagramm: Eine visuelle Darstellung Ihres Workflows, in der Sie Status löschen, neu organisieren und für die Konfiguration auswählen können.
6. Inspektor-Panel: Zeigt die Eigenschaften jedes auf der Leinwand ausgewählten Status an und bearbeitet sie. Abhängig vom Status, der im Arbeitsablaufdiagramm auf der Arbeitsfläche ausgewählt wurde, werden die Registerkarten mit zustandsspezifischen Optionen für Konfiguration, Eingabe/Ausgabe, Variablen und Fehlerbehandlung gefüllt.
7. Informationslinks: Öffnet ein Fenster mit Kontextinformationen, wenn Sie Hilfe benötigen. Diese Panels enthalten auch Links zu verwandten Themen im AWS Step FunctionsDeveloper Guide.



Verwendung von Tests mit nur einem Status während des Entwurfs

Über die Teststatus-Benutzeroberfläche von Workflow Studio können Sie die einzelnen Zustände Ihrer Zustandsmaschine testen. Dazu gehört die Möglichkeit, Statuseingaben bereitzustellen, Variablen festzulegen AWS SAM und beides vorzunehmen sowie AWS CloudFormation Definitionen zu ersetzen.

Weitere Informationen zu Infrastructure as Code (IaC), Ressourcendefinitionen und Datentransformation finden Sie in den Themen Verwenden AWS SAM zum Erstellen von Step Functions-Workflows und Transformieren von Daten mit JSONata in Step Functions im AWS Step FunctionsEntwicklerhandbuch.

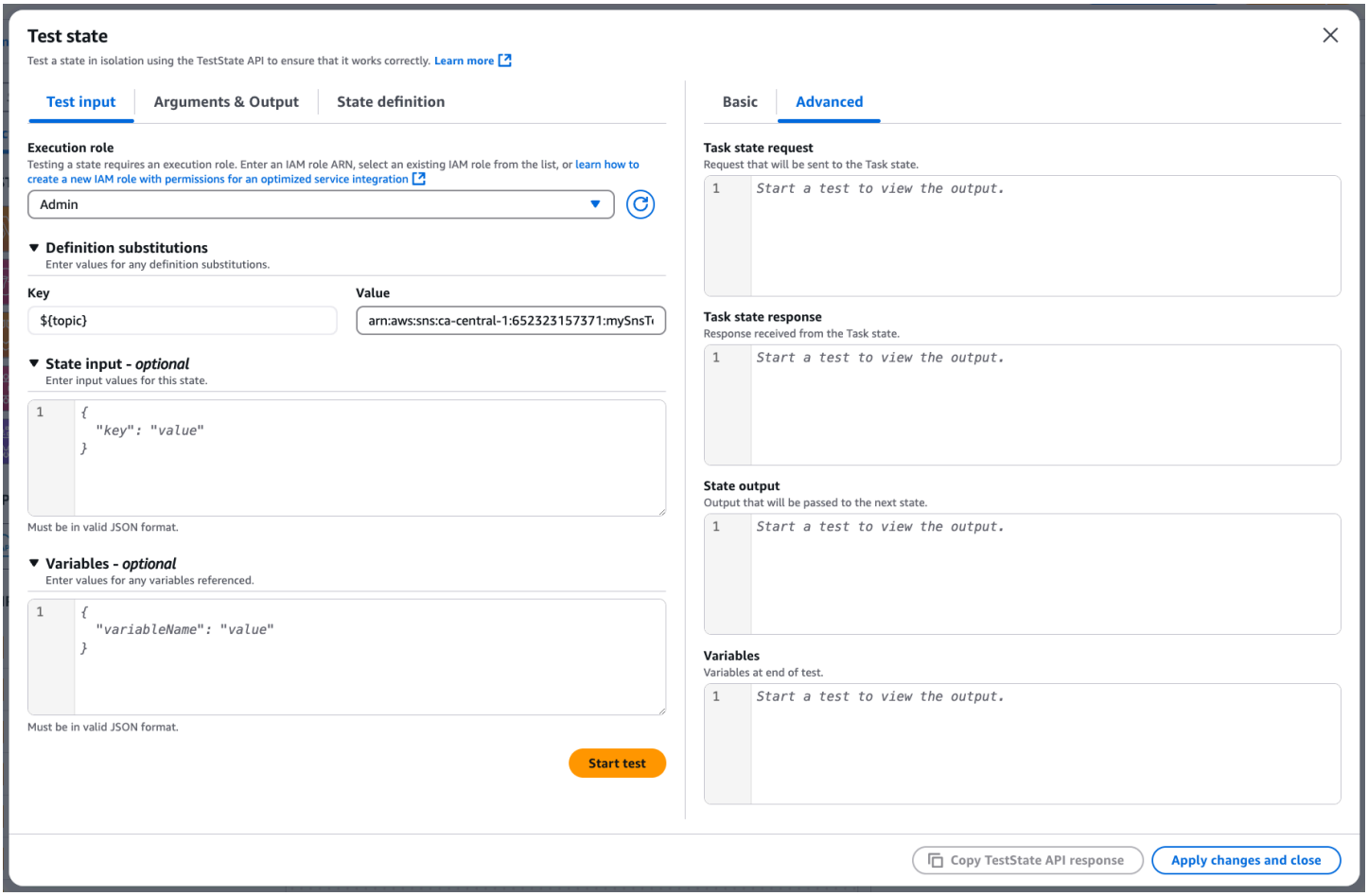
Das folgende Verfahren beschreibt, wie Sie die Teststatus-Benutzeroberfläche in Workflow Studio öffnen.

Öffnen der Teststatus-Benutzeroberfläche

1. Navigieren Sie auf der Registerkarte Designmodus in Workflow Studio zur Leinwand und wählen Sie einen Status aus, um ihn im Inspektorfenster zu öffnen.
2. Wählen Sie im Inspektorfenster die Schaltfläche Teststatus.
3. Die Benutzeroberfläche für den Teststatus wird in VS Code geöffnet.

Die Teststatus-Benutzeroberfläche besteht aus drei Hauptregistern: Testeingabe, Argumente und Ausgabe, Statusdefinition. Die Registerkarte „Testeingabe“ enthält 3 zusätzliche Felder, in denen Sie Eingaben für den Status angeben, Variablen festlegen und Definitionsersetzungen anhand Ihrer AWS SAM Vorlagen angeben können. AWS CloudFormation Auf der Registerkarte Bundesstaatendefinition können Sie den Arbeitsablauf anpassen und erneut testen. Wenn Sie mit der Ausführung der Tests fertig sind, können Sie Änderungen an Ihrer State-Machine-Definition anwenden und speichern.

Der folgende Screenshot zeigt die Teststatus-Benutzeroberfläche, die eine Themenressourcendefinition enthält.



Workflow Studio wird standardmäßig deaktiviert

Standardmäßig ist Workflow Studio der Standardeditor für ASL Dateien, die eine State-Machine-Definition enthalten. Sie können die Standardeinstellung deaktivieren, indem Sie Ihre `settings.json` Datei in Ihrem lokalen `.vscode` Verzeichnis ändern. Wenn Sie Workflow Studio standardmäßig deaktivieren, können Sie immer noch über die Methoden darauf zugreifen, die im Abschnitt Workflow Studio öffnen in diesem Thema aufgeführt sind.

Gehen Sie wie folgt vor, um Ihre `settings.json` Datei in VS Code zu bearbeiten.

1. Öffnen Sie in VS Code die Befehlspalette, indem Sie **`option+shift+p`** (Mac) oder **`ctrl+shift+p`** (Windows) drücken.
2. Geben Sie **`Open User Settings (JSON)`** in der VS Code-Befehlspalette das Suchfeld ein und wählen Sie die Option aus, wenn sie in der Liste angezeigt wird.
3. Fügen Sie `settings.json` in Ihrem Editor die folgende Änderung zu Ihrer Datei hinzu.

```
                {
                    "workbench.editorAssociations": {
                    // Use all the following overrides or a specific one for a
certain file type
                    "*.asl.json": "default",
                    "*.asl.yaml": "default",
                    "*.asl.yml": "default"
                    }
                }
```

4. Speichern Sie Ihre Änderungen `settings.json` und aktualisieren Sie VS Code oder starten Sie ihn neu.

Arbeiten mit Threat Composer

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.

In den folgenden Themen wird beschrieben, wie Sie mit Threat Composer arbeiten 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 ein und drücken Sie dann die **`enter`** Eingabetaste`file 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 ein und drücken Sie dann die **`enter`** Eingabetaste`file 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 auf AWS Dienste, die standardmäßig im AWS Explorer aufgeführt sind, können Sie auch zu Ressourcen wechseln und aus Hunderten von Ressourcen auswählen, die Sie der Benutzeroberfläche hinzufügen möchten. In ist eine Ressource eine Entität AWS, mit der Sie arbeiten können. Zu den Ressourcen, die hinzugefügt werden können, gehören Amazon AppFlow, Amazon Kinesis Data Streams, AWS IAM-Rollen, Amazon VPC und Amazon-Distributionen. CloudFront

Nachdem Sie Ihre Auswahl getroffen haben, können Sie zu Ressourcen navigieren und den Ressourcentyp erweitern, 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:

- Sehen Sie sich eine Liste der vorhandenen Ressourcen an, die in der aktuellen AWS Region für diesen Ressourcentyp verfügbar sind.
- Zeigen Sie eine schreibgeschützte Version der JSON-Datei an, die eine Ressource beschreibt.
- Kopieren Sie die Ressourcenkennung für die Ressource.
- Sehen Sie sich die AWS Dokumentation an, in der der Zweck des Ressourcentyps und das Schema (in den Formaten JSON und YAML) für die Modellierung einer Ressource erläutert werden.
- 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 ist AWS Toolkit for Visual Studio Code die Option zum Erstellen, Bearbeiten und Löschen von Ressourcen eine experimentelle Funktion. Da experimentelle Funktionen noch getestet und aktualisiert werden, kann es zu Problemen im Zusammenhang mit der Benutzerfreundlichkeit kommen. Und experimentelle Funktionen können AWS Toolkit for Visual Studio Code ohne vorherige Ankündigung aus dem entfernt werden.
> Um die Verwendung experimenteller Funktionen für Ressourcen zu ermöglichen, öffnen Sie den Bereich Einstellungen in Ihrer VS Code-IDE, erweitern Sie Erweiterungen und wählen Sie AWS Toolkit.
> Wählen Sie unter AWS Toolkit-Experimente aus jsonResourceModification, ob Sie Ressourcen erstellen, aktualisieren und löschen können.
> Weitere Informationen finden Sie unter Arbeiten mit experimentellen Funktionen.

IAM-Berechtigungen für den Zugriff auf Ressourcen

Für den Zugriff auf die Ressourcen, die den AWS Diensten zugeordnet sind, benötigen Sie spezielle AWS Identity and Access Management Berechtigungen. 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 Dienstressourcen benötigt eine IAM-Entität Berechtigungen, damit das Toolkit for VS Code AWS Cloud Control API-Operationen in ihrem Namen aufrufen kann. Cloud-Control-API-Vorgänge ermöglichen dem IAM-Benutzer oder der IAM-Rolle, auf die Remote-Ressourcen zuzugreifen und diese zu aktualisieren.

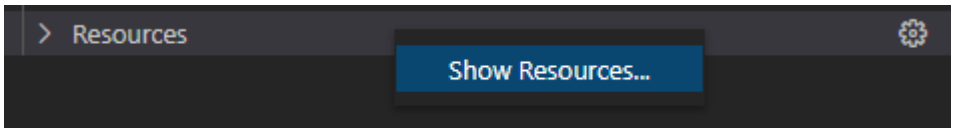
Der einfachste Weg, Berechtigungen zu erteilen, besteht darin PowerUserAccess, die AWS verwaltete Richtlinie über die Toolkit-Schnittstelle an die IAM-Entität anzuhängen, die diese API-Operationen aufruft. Diese verwaltete Richtlinie erteilt eine Reihe von Berechtigungen zum Ausführen von Anwendungsentwicklungsaufgaben, 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 von Ressourcen und Interagieren mit bereits vorhandenen Ressourcen

1. Klicken Sie im AWS Explorer mit der rechten Maustaste auf Ressourcen und wählen Sie Ressourcen anzeigen.

   In einem Bereich wird eine Liste der Ressourcentypen angezeigt, die zur Auswahl stehen.

2. Wählen Sie im Auswahlbereich die Ressourcentypen aus, die Sie dem AWS Explorer hinzufügen möchten, und drücken Sie die EINGABETASTE oder wählen Sie zur Bestätigung OK.

   Die von Ihnen ausgewählten Ressourcentypen werden unter Ressourcen aufgeführt.

   > **Note**
   >
   > Wenn Sie dem AWS Explorer bereits einen Ressourcentyp hinzugefügt haben und dann das Kontrollkästchen für diesen Typ deaktivieren, wird er nach dem Klicken auf OK

nicht mehr unter Ressourcen aufgeführt. Nur die aktuell ausgewählten Ressourcentypen werden im AWS Explorer angezeigt.

3. Um die Ressourcen anzuzeigen, die bereits für einen Ressourcentyp vorhanden sind, erweitern Sie den Eintrag für diesen Typ.

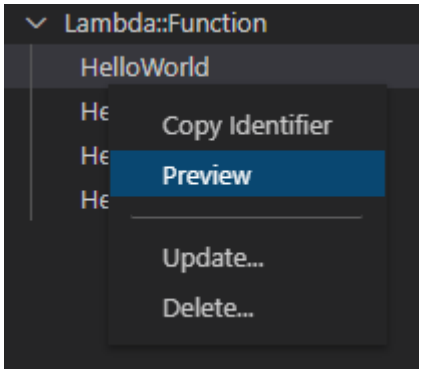
   Eine Liste der verfügbaren Ressourcen wird unter ihrem Ressourcentyp angezeigt.

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:

   - Ressourcen-ID kopieren: Kopiert die ID für die spezifische Ressource in die Zwischenablage. (Beispielsweise kann die `AWS Toolkit:DynamoDB::Table`-Ressource mit der `TableName`-Eigenschaft identifiziert werden.)
   - Preview (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 auf der rechten Seite des Editors auf das Aktualisierungssymbol klicken. Um eine Ressource zu aktualisieren, müssen Sie die erforderlichen Optionen ??? aktiviert haben.
   - Update: Bearbeiten Sie die JSON-formatierte Vorlage für die Ressource in einem VS-Code-Editor. Weitere Informationen finden Sie unter Ressourcen erstellen und bearbeiten.
   - Löschen: Löschen Sie die Ressource, indem Sie den Löschvorgang in einem angezeigten Dialogfeld bestätigen. (Das Löschen von Ressourcen ist derzeit ??? in dieser Version von AWS Toolkit for Visual Studio Code.)

   > **Warning**
   >
   > Wenn Sie eine Ressource löschen, können alle AWS CloudFormation Stacks, die diese Ressource verwenden, nicht aktualisiert werden. Um diesen Aktualisierungsfehler zu beheben, müssen Sie entweder die Ressource neu erstellen oder den Verweis darauf in der AWS CloudFormation Vorlage des Stacks entfernen. Weitere Informationen finden Sie in diesem Knowledge-Center-Artikel.



Ressourcen erstellen und bearbeiten

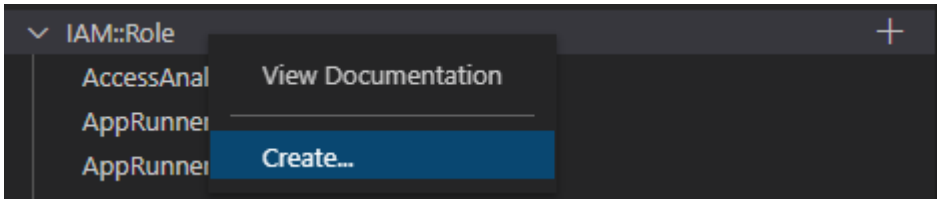
> **Important**
>
> Die Erstellung und Aktualisierung von Ressourcen ist derzeit ein ??? in dieser Version des AWS Toolkit for Visual Studio Code.

Zum Erstellen einer neuen Ressource muss der Liste Ressourcen ein Ressourcentyp hinzugefügt und anschließend eine Vorlage im JSON-Format bearbeitet werden, die die Ressource sowie ihre Eigenschaften und Attribute definiert.

Beispielsweise wird eine Ressource, die zum `AWS Toolkit:SageMaker::UserProfile` Ressourcentyp gehört, mit einer Vorlage definiert, die ein Benutzerprofil für Amazon SageMaker AI Studio erstellt. Die Vorlage, die diese Benutzerprofilressource definiert, muss dem Ressourcentypschema für `AWS Toolkit:SageMaker::UserProfile` entsprechen. Wenn die Vorlage beispielsweise aufgrund fehlender oder falscher Eigenschaften nicht dem Schema entspricht, kann die Ressource nicht erstellt oder aktualisiert werden.

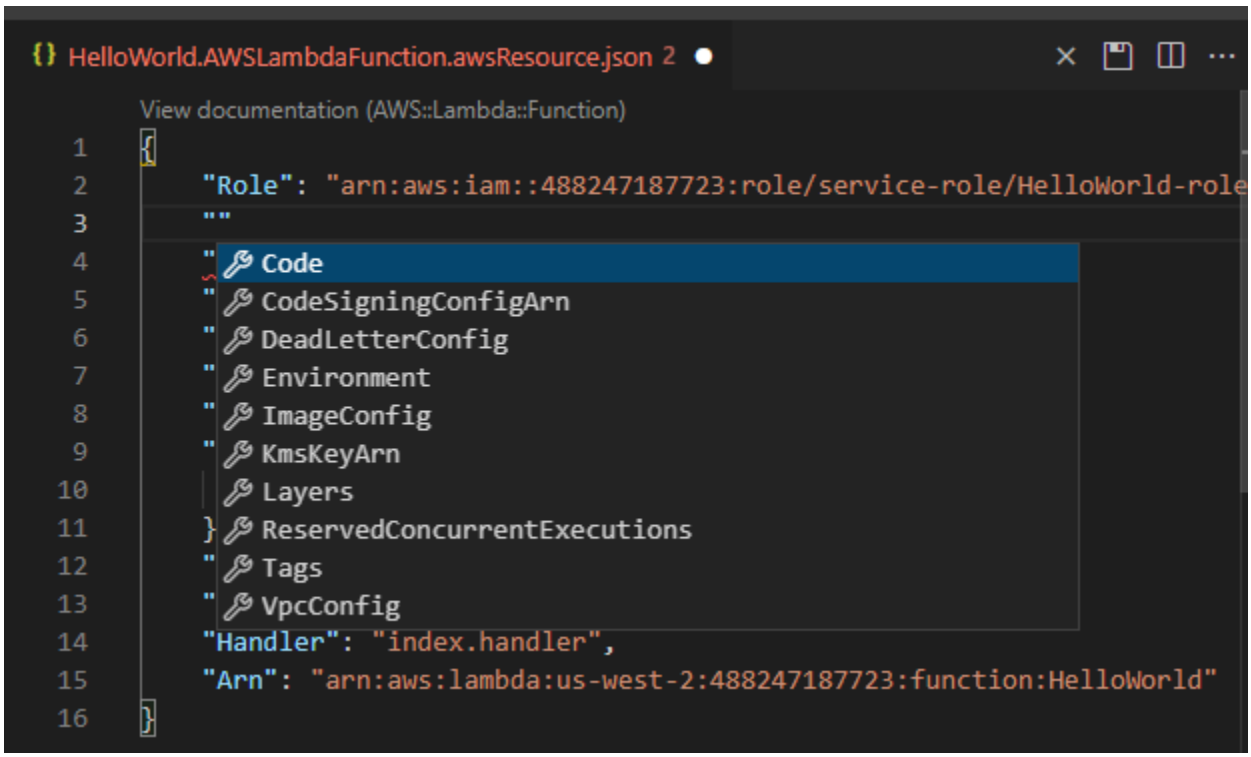
1. Fügen Sie den Ressourcentyp für die Ressource hinzu, die Sie erstellen möchten, indem Sie mit der rechten Maustaste auf Ressourcen klicken und Ressourcen anzeigen wählen.
2. Nachdem der Ressourcentyp unter Ressourcen hinzugefügt wurde, wählen Sie das Plussymbol („+“), um die Vorlagendatei in einem neuen Editor zu öffnen.

   Sie können auch mit der rechten Maustaste auf den Namen des Ressourcentyps klicken und Erstellen wählen. Sie können auch auf Informationen zur Modellierung der Ressource zugreifen, indem Sie Dokumentation anzeigen wählen.



3. Beginnen Sie im Editor damit, Eigenschaften zu definieren, aus denen sich die Ressourcenvorlage zusammensetzt. Die AutoVervollständigen-Funktion schlägt passende Eigenschaftsnamen für das Schema Ihrer Vorlage vor. Wenn Sie mit der Maus auf einen Eigenschaftstyp zeigen, wird in einem Bereich eine Beschreibung dessen, wofür er verwendet wird, angezeigt. Ausführliche Informationen zum Schema erhalten Sie, wenn Sie Dokumentation anzeigen wählen.

   Jeder Text, der nicht dem Ressourcenschema entspricht, wird durch eine rote Wellenlinie gekennzeichnet.



4. Nachdem Sie Ihre Ressource deklariert haben, klicken Sie auf das Symbol Speichern, um Ihre Vorlage zu validieren und die Ressource in der Remote-Cloud zu speichern. AWS

   Wenn Ihre Vorlage die Ressource im Einklang mit ihrem Schema definiert, wird eine Bestätigungsmeldung für die Ressourcenerstellung angezeigt. (Ist die Ressource bereits vorhanden, bestätigt die Meldung, dass die Ressource aktualisiert wurde.)

Nachdem die Ressource erstellt wurde, wird sie der Liste unter der Ressourcentypüberschrift hinzugefügt.

5. Wenn Ihre Datei Fehler enthält, wird eine Meldung mit dem Hinweis angezeigt, dass die Ressource nicht erstellt oder aktualisiert werden konnte. Wählen Sie „Protokolle anzeigen“, um die Vorlagenelemente zu identifizieren, die Sie korrigieren müssen.

Problembehandlung bei AWS Toolkit for Visual Studio Code

Die folgenden Abschnitte enthalten allgemeine Informationen zur Fehlerbehebung zu den Diensten aus dem Toolkit AWS Toolkit for Visual Studio Code und zur Arbeit mit den AWS Diensten aus dem Toolkit. Informationen zu Problemen, die sich speziell auf die Behebung von SAM-Problemen im AWS Toolkit beziehen, finden Sie unter dem Thema Problembehandlung für serverlose Anwendungen in diesem Benutzerhandbuch.

Themen

- Bewährte Methoden zur Fehlerbehebung
- Profil... konnte in der Konfigurationsdatei nicht gefunden werden
- SAM-JSON-Schema: Das Schema in der Datei template.yaml kann nicht geändert werden

Bewährte Methoden zur Fehlerbehebung

Im Folgenden werden bewährte Methoden zur Behebung von AWS Toolkit for Visual Studio Code Problemen empfohlen. Ausführliche Informationen darüber, wie Sie zu der beitragen können AWS Toolkit for Visual Studio Code, finden Sie im AWS Toolkit for Visual Studio Code Thema Beiträge zum AWS Toolkit for Visual Studio Code GitHub Repository.

- Versuchen Sie, Ihr Problem oder Ihren Fehler erneut zu erstellen, bevor Sie einen Bericht senden.
- Machen Sie sich während des Wiederherstellungsvorgangs detaillierte Notizen zu jedem Schritt, jeder Einstellung und jeder Fehlermeldung.
- Sammeln Sie Ihre AWS Toolkit-Debug-Logs. Eine ausführliche Beschreibung, wie Sie Ihre AWS Toolkit-Debug-Logs finden, finden Sie im Abschnitt So finden Sie Ihre AWS Logs in diesem Benutzerhandbuch.
- Suchen Sie im Bereich AWS Toolkit for Visual Studio Code Probleme des Repositorys nach offenen Anfragen und bekannten Lösungen oder melden Sie Ihr ungelöstes Problem. AWS Toolkit for Visual Studio Code GitHub

> **Note**
>
> Das folgende Verfahren beschreibt, wie Sie Ihre AWS Toolkit-Debug-Logs einsehen können. Der Vorgang zum Anzeigen Ihrer Amazon Q-Debug-Logs ist identisch, außer dass Sie Amazon Q: View Logs aus der VS Code Command Palette wählen.

So finden Sie Ihre AWS Toolkit for Visual Studio Code Debug-Protokolle

1. Öffnen Sie im VS-Code die Befehlspalette, indem Sie **`Cmd + Shift + P`** oder **`Ctrl + Shift + P`** (Windows) drücken und **`AWS View Logs`** in das Suchfeld eingeben.
2. Wählen Sie Protokolle AWS anzeigen, um Ihre AWS Toolkit-Protokolle im Ausgabefenster des VS Code-Terminals zu öffnen.
3. Erweitern Sie im Ausgabefenster des VS Code-Terminals das Zahnradsymbolmenü und wählen Sie Debug.
4. Erweitern Sie das Zahnradsymbolmenü erneut und wählen Sie Als Standard festlegen.
5. Öffne die Befehlspalette erneut, indem du auf **`Cmd + Shift + P`** oder **`Ctrl + Shift + P`** (Windows) drückst und nach „Entwickler: Fenster neu laden“ suchst **`Reload Window`** und dann auswählst.
6. VS Code wird neu geladen und im Ausgabefenster des VS Code-Terminals werden Ihre aktualisierten AWS Toolkit-Debug-Protokolle angezeigt.

Profil... konnte in der Konfigurationsdatei nicht gefunden werden

Problem

> **Note**
>
> Dieses Problem betrifft nur die `~/.aws/config` Datei und nicht die `~/.aws/credentials` Datei. Ausführliche Informationen zu den Dateien für AWS Konfiguration und AWS Anmeldeinformationen finden Sie im Thema Gemeinsam genutzte Konfigurationsdateien und Anmeldeinformationen im Referenzhandbuch für AWS SDK und Tools.

Bei der Auswahl von Anmeldeinformationen wird in den AWS Toolkit-Protokollen eine Meldung mit dieser Struktur angezeigt:`Profile` *`name`* `could not be found in shared credentials file`.

Im Folgenden finden Sie ein Beispiel dafür, wie dieser Fehler in Ihren AWS Toolkit-Protokollen aussieht:

```
        2023-08-08 18:20:45 [ERROR]: _aws.auth.reauthenticate: Error: Unable to
authenticate connection
        -> CredentialsProviderError: Profile vscode-prod-readonly could not be found
in shared credentials file.
```

Lösung

Wenn Ihr Profil bereits in existiert`~/.aws/config`, überprüfen Sie, ob es mit `[profile` beginnt. Im Folgenden finden Sie ein Beispiel für ein Benutzerprofil, das korrekt strukturiert ist:

```
        [profile example]
        region=us-west-2
        credential_process=...
```

Das Folgende ist ein Beispiel für ein Benutzerprofil, das falsch strukturiert ist:

```
        [example]
        region=us-west-2
        credential_process=...
```

SAM-JSON-Schema: Das Schema in der Datei template.yaml kann nicht geändert werden

Problem

Sie können in SAM template.yaml kein anderes JSON-Schema manuell auswählen

Lösung

Nach dem Update auf vscode-yaml Version 1.11+ können Sie eine **`yaml-language-server`** Modeline am Anfang einer YAML-Datei hinzufügen, um die Verwendung eines Schemas per URI zu erzwingen. Weitere Informationen zur Verwendung von Inline-Schemas finden Sie im Thema Yaml-Sprachserver im Redhat-Entwickler-Repository. GitHub Das Folgende ist ein Beispiel für eine Modeline. **`yaml-language-server`**

```
        # yaml-language-server: $schema=https://raw.githubusercontent.com/aws/
serverless-application-model/main/samtranslator/schema/schema.json
```

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, AWS ist verantwortlich für den Schutz der globalen Infrastruktur, auf der die AWS Cloud gesamte Infrastruktur läuft. 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, dass Sie AWS-Konto Anmeldeinformationen schützen und einzelne Benutzer mit AWS IAM Identity Center oder AWS Identity and Access Management (IAM) einrichten. 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, um mit Ressourcen zu kommunizieren. AWS Wir benötigen TLS 1.2 und empfehlen TLS 1.3.
- Richten Sie die API und die Protokollierung von Benutzeraktivitäten mit ein. AWS CloudTrail Informationen zur Verwendung von CloudTrail Pfaden zur Erfassung von AWS Aktivitäten finden Sie unter Arbeiten mit CloudTrail Pfaden im AWS CloudTrail Benutzerhandbuch.
- Verwenden Sie AWS Verschlüsselungslösungen zusammen mit allen darin enthaltenen Standardsicherheitskontrollen 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 AWS über eine Befehlszeilenschnittstelle oder eine API FIPS 140-3-validierte kryptografische Module benötigen, verwenden Sie einen FIPS-Endpunkt. Weitere Informationen über verfügbare FIPS-Endpunkte finden Sie unter Federal Information Processing Standard (FIPS) 140-3.

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 auf andere AWS-Services Weise 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.

Dokumentenverlauf für das AWS Toolkit for Visual Studio Code Benutzerhandbuch

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
Mit AWS Lambda Funktionen arbeiten	Das Thema des Benutzerhandbuchs wird aktualisiert, sodass aktualisierte Lambda-Funktionen in das Toolkit aufgenommen werden.	17. Juli 2025
AWS Lambda Debuggen aus der Ferne	Dem AWS Toolkit for Visual Studio Code Benutzerhandbuch wurde ein neues Thema zum AWS Lambda Remote-Debugging hinzugefügt.	17. Juli 2025
AWS Lambda Von der Konsole zur IDE	Dem AWS Toolkit for Visual Studio Code Benutzerhandbuch wurde ein neues AWS Lambda console IDE-Thema hinzugefügt.	17. Juli 2025
AWS Step Functions Inhaltsaktualisierungen und zusätzliche Unterstützung für Workflow Studio	Es wurden Aktualisierungen für bestehende Inhalte AWS Step Functions und das Thema im Benutzerhandbuch für AWS Step Functions Workflow Studio hinzugefügt, um die Einführung von Funktionen zu unterstützen.	6. März 2025

AWS Serverloses Land	Dem AWS Application Builder-Inhaltsverzeichnis wurde ein neues Thema AWS Serverless Land hinzugefügt.	6. März 2025
Firewalls und Gateways werden aktualisiert, um den Zugriff zu ermöglichen	Listen von Endpunkten und Ressourcen, denen der Zugriff auf alle Services und Funktionen in den Erweiterungen Amazon Q for VS Code AWS Toolkit for Visual Studio Code und Amazon Q for VS Code gestattet werden muss.	28. Februar 2025
Support für Amazon ECR App Runner	Dokumentationsunterstützung für das Starten eines AWS App Runner Dienstes über den Amazon Elastic Container Registry-Knoten im AWS Toolkit hinzugefügt.	6. Februar 2025
Amazon DocumentDB	Dem AWS Toolkit for Visual Studio Code Benutzerhandbuch wurde ein neues Amazon DocumentDB-Thema hinzugefügt.	6. Februar 2025
EC2 Unterstützung	Support für den Amazon Elastic Compute Cloud-Service wurde dem Toolkit hinzugefügt.	31. Januar 2025
AWS Dokumente	Neues Thema im Benutzerhandbuch für AWS Dokumente hinzugefügt.	20. Januar 2025

Amazon CloudWatch protokolliert Live-Tail	Es wurde ein neues Unterthema hinzugefügt, um die Amazon CloudWatch Logs Live Tail-Funktion in der AWS Toolkit for Visual Studio Code zu unterstützen.	15. Dezember 2024
AWS Anwendungsgenerator	Dem AWS Toolkit for Visual Studio Code Benutzerhandbuch wurde ein neues AWS Application Builder-Thema hinzugefügt.	30. Oktober 2024
Infrastructure Composer	AWS Application Composer ist jetzt AWS Infrastructure Composer.	3. Oktober 2024
AWS Updates für Identity and Access Management (IAM) Access Analyzer	Der Inhalt von IAM Access Analyzer wurde aktualisiert und enthält nun neue API-Referenzen.	10. Juli 2024
AWS Access Analyzer für Identity and Access Management (IAM)	Neues Thema im Benutzerhandbuch für IAM Access Analyzer hinzugefügt.	23. Mai 2024
Connect zum AWS Autorisierungsablauf herstellen wurde aktualisiert	Der Autorisierungsablauf wurde aktualisiert, um den Änderungen am Authentifizierungsprozess und der Trennung von Amazon Q vom AWS Toolkit for Visual Studio Code Rechnung zu tragen.	30. April 2024

Amazon Q-Erweiterung für VS Code	Seit dem 30. April 2024 CodeWhisperer ist es jetzt Teil von Amazon Q und Amazon Q ist als Erweiterung für VS Code verfügbar.	30. April 2024
Support für Virtual Private Cloud in Entwicklungsumgebungen	Aktualisierter Inhalt zu Änderungen an der Benutzeroberfläche zur Unterstützung von VPC in Entwicklungsumgebungen.	21. Januar 2024
Komponist der Infrastruktur	Dem AWS Toolkit for Visual Studio Code Benutzerhandbuch wurde ein neues Infrastructure Composer-Thema hinzugefügt.	28. November 2023
SSO-Unterstützung für CodeCatalyst	Der Inhalt wurde aktualisiert, um die Unterstützung von IAM Identity Center CodeCatalyst und Entwicklungsumgebungen zu behandeln.	17. November 2023
Download-Links für VS Code und Toolkit hinzugefügt	Aktualisierter Inhalt mit Download-Links für VS Code und die AWS Toolkit for Visual Studio Code.	1. November 2023
Amazon Redshift Redshift-Thema	Dem AWS Toolkit for Visual Studio Code Benutzerhandbuch wurde ein neues Amazon Redshift Redshift-Thema hinzugefügt.	17. Oktober 2023

Connect zum AWS Autorisierungsablauf herstellen wurde aktualisiert	Der Autorisierungsablauf wurde aktualisiert und konzentriert sich nun auf dienstspezifische Authentifizierungsmethoden.	29. September 2023
Benutzeranleitung erstellt: Erstellen Sie eine Vorlage CloudFormation	Es wurde eine neue Benutzeranleitung erstellt, in der 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 für das Aktivieren experimenteller Funktionen für AWS Dienste wurde hinzugefügt.	14. Oktober 2021

Support 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 Features 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 emulierten Umgebungen als auch in ARM64 x86_64-basierten Umgebungen ausführen.	1. Oktober 2021
AWS-Serverless-Anwendung	Unterstützung für die Ausführung von Anwendungen auf der Plattform hinzugefügt AWS SAM ARM64	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	Unterstützung für to hinzugefügt. AWS App Runner 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-Ressourcen	Unterstützung für das Debuggen lokaler Amazon API Gateway Gateway-Ressourcen 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 Unterstützung	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 Unterstützung	Unterstützung für hinzugefügt AWS Step Functions.	31. März 2020
Inhalt zum Thema Sicherheit	Sicherheitsinhalte hinzugefügt.	6. Februar 2020
Mit EventBridge Amazon-Schemas arbeiten	Unterstützung für Amazon EventBridge Schemas 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
Installation des AWS Toolkit for Visual Studio Code	Es wurden Informationen zur Installation der Sprache SDKs zur Unterstützung verschiedener Toolchains hinzugefügt.	12. Juni 2019
Konfiguriere deine 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.