Guide de l’utilisateur

AWS BOÎTE À OUTILS POUR VS CODE

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

AWS BOÎTE À OUTILS POUR VS CODE: Guide de l’utilisateur

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

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

Table of Contents

AWS Toolkit for Visual Studio Code 1
Qu'est-ce que AWS Toolkit for Visual Studio Code 1
Informations connexes 1
Développeur Amazon Q et Amazon CodeWhisperer 2
Téléchargez le kit d'outils 3
Téléchargement du kit d'outils depuis VS Code Marketplace 3
Kits d'outils IDE supplémentaires de AWS 3
Démarrage 4
Installation du Toolkit for VS Code 4
Prérequis 4
Téléchargement et installation du AWS Toolkit for Visual Studio Code 4
Prérequis facultatifs 5
Connexion à AWS 6
Inscrivez-vous pour un Compte AWS 6
Conditions préalables 6
Ouverture du panneau de connexion 7
Connexion à AWS à partir de la boîte à outils 7
Authentification pour Amazon CodeCatalyst 9
Changements de AWS régions 10
Ajouter une région à l' AWS explorateur 10
Masquer une région dans l' AWS explorateur 10
Configuration de votre chaîne d'outils 11
Configuration d'une chaîne d'outils pour .NET Core 11
Configurer une chaîne d'outils pour Node.js 11
Configuration d'une chaîne d'outils pour Python 11
Configuration d'une chaîne d'outils pour Java 12
Configurer une chaîne d'outils pour Go 12
Utilisation de votre chaîne d'outils 13
Authentification et accès 14
IAM Identity Center 14
Informations d'identification IAM 15
Création d'un utilisateur IAM 15
Création d'un fichier d'informations d'identification partagé à partir du AWS Toolkit for Visual Studio Code 16

Ajouter des profils d'identification supplémentaires 17
AWS ID du constructeur 18
Utilisation d'un processus d'identification externe 18
Mise à jour des pare-feux et des passerelles 18
AWS Toolkit for Visual Studio Code Points de terminaison 19
Points de terminaison du plugin Amazon Q 19
Points de terminaison Amazon Q pour développeurs 20
Points de terminaison Amazon Q Code Transform 20
Points de terminaison d'authentification 20
Points de terminaison d'identité 21
Télémétrie 21
Références 22
Travailler avec AWS 23
Fonctionnalités expérimentales 24
AWS Explorateur 24
AWS Documents 25
Mise en route avec AWS Documents 26
Affichage de la documentation, de l'autocomplétion et de la validation dans VS Code 26
Amazon CodeCatalyst 27
Qu'est-ce qu'Amazon CodeCatalyst ? 27
Mise en route avec CodeCatalyst 28
Utilisation des ressources CodeCatalyst 28
Travailler avec des environnements de développement 32
Résolution des problèmes 35
Amazon API Gateway 37
AWS App Runner 37
Conditions préalables 38
Tarification 41
Création de services App Runner 41
Gestion des services App Runner 45
AWS Générateur d'applications 47
Travailler avec AWS Application Builder 47
AWS Compositeur d'infrastructure 52
Travailler avec AWS Infrastructure Composer 52
AWS CDK 54
AWS CDK applications 54

CloudFormation piles 56
Supprimer une CloudFormation pile 57
Création d'un CloudFormation modèle 58
Amazon CloudWatch Logs 59
Affichage des groupes de CloudWatch journaux et des flux de journaux 60
Utilisation des événements du CloudWatch journal 61
Recherche de groupes de journaux 63
CloudWatch Logs Live Tail 66
Amazon DocumentDB 68
Utilisation avec Amazon DocumentDB 68
Amazon EC2 74
Utilisation avec Amazon EC2 74
Résolution des problèmes liés à Amazon EC2 84
Amazon ECR 86
Travailler avec Amazon ECR 87
Création d'un service App Runner 97
Amazon ECS 99
Utilisation IntelliSense pour les fichiers de définition de tâches 99
Amazon ECS Exec 100
Amazon EventBridge 103
Utilisation d'Amazon EventBridge Schemas 103
AWS Analyseur d'accès IAM 105
Utilisation de l' AWS analyseur d'accès IAM 106
AWS IoT 111
AWS IoT prérequis 111
AWS IoT Choses 111
AWS IoT certificats 113
AWS IoT politiques 116
AWS Lambda Fonctions 119
Travailler avec AWS Lambda Fonctions 120
AWS Lambda console vers IDE 126
AWS Lambda LocalStack soutien 128
Débogage à distance Lambda 133
Amazon Redshift 145
Travailler avec Amazon Redshift 145
Amazon S3 150

Utilisation des ressources S3 150
Utilisation d'objets S3 152
Amazon SageMaker Unified Studio 156
Application sans serveur AWS 157
Démarrage 157
Utilisation de Serverless Land 165
Exécution et débogage de fonctions Lambda directement à partir du code 167
Exécution et débogage des ressources locales Amazon API Gateway 173
Options de configuration pour le débogage d'applications sans serveur 176
Résolution des problèmes 183
AWS Systems Manager 185
Hypothèses et prérequis 186
Autorisations IAM pour les documents d'automatisation de Systems Manager 186
Création d'un nouveau document d'automatisation de Systems Manager 187
Ouverture d'un document Systems Manager Automation existant 188
Modification d'un document d'automatisation de Systems Manager 189
Publication d'un document sur l'automatisation de Systems Manager 189
Supprimer un document d'automatisation de Systems Manager 190
Exécution d'un document d'automatisation de Systems Manager 191
Résolution des problèmes 191
AWS Step Functions 192
Utilisation des fonctions Step 192
Utilisation de Workflow Studio 196
Threat Composer 201
Utilisation de Threat Composer 201
Ressources 202
Autorisations IAM pour l'accès aux ressources 203
Ajout et interaction avec des ressources existantes 204
Création et modification de ressources 206
Résolution des problèmes 208
Bonnes pratiques de résolution des problèmes 208
Le profil... est introuvable dans le fichier de configuration 209
Schéma SAM json : impossible de modifier le schéma dans le fichier template.yaml 210
Sécurité 212
Protection des données 212
Historique de la documentation 214

..... ccxxii

AWS Toolkit for Visual Studio Code

Ceci est le guide de l'utilisateur du AWS Toolkit for VS Code. Si vous recherchez le AWS Toolkit for Visual Studio, consultez le Guide de l'utilisateur pour le AWS Toolkit for Visual Studio.

Qu'est-ce que AWS Toolkit for Visual Studio Code

Le Toolkit for VS Code est une extension open source pour l'éditeur Visual Studio Code (VS Code). Cette extension permet aux développeurs de développer, de déboguer localement et de déployer plus facilement des applications sans serveur qui utilisent Amazon Web Services ()AWS.

Rubriques

- Commencer à utiliser le AWS Toolkit for Visual Studio Code
- Travailler avec AWS services et outils

Informations connexes

Utilisez les ressources suivantes pour accéder au code source de la boîte à outils ou afficher les problèmes en cours.

- Code source
- Dispositif de suivi des problèmes

Pour en savoir plus sur l'éditeur de code Visual Studio, rendez-vous sur https://code.visualstudio.com/.

Développeur Amazon Q et Amazon CodeWhisperer

Depuis le 30 avril 2024, Amazon CodeWhisperer fait désormais partie d'Amazon Q Developer, ce qui inclut des suggestions de code en ligne et des analyses de sécurité Amazon Q Developer. Téléchargez l'extension Amazon Q Developer IDE depuis VS Code Marketplace pour commencer.

Pour plus d'informations sur le service Amazon Q Developer, consultez le Guide de l'utilisateur Amazon Q Developer. Pour obtenir des informations détaillées sur les forfaits et les tarifs d'Amazon Q, consultez le guide de tarification d'Amazon Q.

Téléchargement du Toolkit for VS Code

Vous pouvez le télécharger, l'installer et le configurer AWS Toolkit for Visual Studio Code via VS Code Marketplace dans votre IDE. Pour obtenir des instructions détaillées, consultez la section Téléchargement et installation de la rubrique Mise en route de ce guide de l'utilisateur.

Téléchargement du kit d'outils depuis VS Code Marketplace

Vous pouvez également télécharger les fichiers AWS Toolkit for Visual Studio Code d'installation en accédant au VS Code Marketplace depuis votre navigateur Web.

Kits d'outils IDE supplémentaires de AWS

En plus de cela AWS Toolkit for Visual Studio Code, propose AWS également des boîtes à outils IDE pour JetBrains Visual Studio.

AWS Toolkit for JetBrains liens

- Suivez ce lien pour le télécharger AWS Toolkit for JetBrains depuis le JetBrains Marketplace.
- Pour en savoir plus AWS Toolkit for JetBrains, consultez le guide de l'AWS Toolkit for JetBrainsutilisateur.

Liens vers le Toolkit for Visual Studio

- Suivez ce lien pour télécharger le Toolkit pour Visual Studio depuis le Visual Studio Marketplace.
- Pour en savoir plus sur le Toolkit for Visual Studio, consultez le guide de l'utilisateur du Toolkit for Visual Studio.

Commencer à utiliser le AWS Toolkit for Visual Studio Code

AWS Toolkit for Visual Studio Code Cela rend vos AWS services et ressources disponibles, directement depuis votre environnement de développement intégré (IDE) VS Code.

Pour vous aider à démarrer, les rubriques suivantes décrivent comment configurer, installer et configurer le AWS Toolkit for Visual Studio Code.

Rubriques

- Installation du AWS Toolkit for Visual Studio Code
- Connexion à AWS
- Changements de AWS régions
- Configuration de votre chaîne d'outils

Installation du AWS Toolkit for Visual Studio Code

Prérequis

Pour commencer à travailler avec AWS Toolkit for Visual Studio Code VS Code, les conditions suivantes doivent être remplies. Pour en savoir plus sur l'accès à tous les AWS services et ressources disponibles sur le AWS Toolkit for Visual Studio Code site, consultez la the section called "Prérequis facultatifs" section de ce guide.

- VS Code nécessite un système d'exploitation Windows, macOS ou Linux.
- Vous devez AWS Toolkit for Visual Studio Code travailler à partir de la version 1.73.0 de VS Code ou d'une version ultérieure.

Pour plus d'informations sur VS Code ou pour télécharger la dernière version de VS Code, consultez le site Web de téléchargement de VS Code.

Téléchargement et installation du AWS Toolkit for Visual Studio Code

Vous pouvez le télécharger, l'installer et le configurer AWS Toolkit for Visual Studio Code via VS Code Marketplace dans votre IDE. Vous pouvez également télécharger les fichiers AWS Toolkit for Visual Studio Code d'installation en accédant au VS Code Marketplace depuis votre navigateur Web.

Installation du AWS Toolkit for Visual Studio Code depuis VS Code IDE Marketplace

1. Ouvrez l' AWS Toolkit for Visual Studio Code extension dans votre IDE VS Code à l'aide du lien suivant : Ouvrez le VS Code Marketplace.

   > **Note**
   >
   > Si VS Code n'est pas déjà en cours d'exécution sur votre machine, cette opération peut prendre quelques instants pendant le chargement de VS Code.

2. À partir de l' AWS Toolkit for Visual Studio Code extension dans VS Code Marketplace, choisissez Installer pour commencer le processus d'installation.
3. Lorsque vous y êtes invité, choisissez de redémarrer VS Code pour terminer le processus d'installation.

Prérequis facultatifs

Avant de pouvoir utiliser certaines fonctionnalités du AWS Toolkit for Visual Studio Code, vous devez disposer des éléments suivants :

- Compte Amazon Web Services (AWS) : il n'est pas obligatoire d'avoir un AWS compte pour utiliser le AWS Toolkit for Visual Studio Code, mais les fonctionnalités sont considérablement limitées sans ce compte. Pour obtenir un AWS compte, rendez-vous sur la page d'AWS accueil. Choisissez Créer un AWS compte ou Terminer l'inscription (si vous avez déjà visité le site).
- Développement de code — Le SDK correspondant au langage que vous souhaitez utiliser. Vous pouvez le télécharger à partir des liens suivants, ou utiliser votre gestionnaire de package favori :
  - Kit SDK .NET : https://dotnet.microsoft.com/download
  - Kit SDK Node.js : https://nodejs.org/en/download
  - Kit SDK Python : https://www.python.org/downloads
  - SDK Java : https://aws.amazon.com/corretto/
  - Accédez au SDK : https://golang.org/doc/install
- AWS SAM CLI — Il s'agit d'un outil AWS CLI qui vous aide à développer, tester et analyser vos applications sans serveur localement. Cette interface n'est pas obligatoire pour l'installation du Toolkit. Cependant, nous vous recommandons de l'installer (ainsi que Docker, décrit ci-dessous) car il est nécessaire pour toutes les fonctionnalités AWS Serverless Application Model (AWS SAM), telles queCréation d'une nouvelle application sans serveur (locale).

Pour plus d'informations, consultez la section Installation de la AWS SAM CLI dans le guide du AWS Serverless Application Model développeur.

- Docker — La AWS SAM CLI nécessite cette plate-forme de conteneur logiciel open source. Pour plus d'informations et obtenir les instructions de téléchargement, consultez Docker.
- Gestionnaire de packages : gestionnaire de packages qui vous permet de télécharger et de partager le code de l'application.
  - .NET : NuGet
  - Node.js : npm
  - Python : pip
  - Java : Gradle ou Maven

Connexion à AWS

La plupart des ressources Amazon Web Services (AWS) sont gérées via un AWS compte. Un AWS compte n'est pas nécessaire pour utiliser le AWS Toolkit for Visual Studio Code, mais les fonctions du Toolkit sont limitées sans connexion.

Si vous avez déjà configuré un AWS compte et que vous vous êtes authentifié via un autre AWS service (tel que le AWS Command Line Interface), vos informations d'identification AWS Toolkit for Visual Studio Code sont automatiquement détectées.

Inscrivez-vous pour un Compte AWS

Pour commencer AWS, vous avez besoin d'un Compte AWS. Il existe deux façons de s'inscrire à AWS. Si vous souhaitez contrôler totalement la configuration de votre compte et l'emplacement de stockage de votre contenu, ou si vous avez des charges de travail réglementées telles que HIPAA ou FedRAMP, utilisez Sign up for (advanced). AWS Si vous êtes un nouvel utilisateur et que vous souhaitez utiliser un identifiant que vous possédez déjà pour vous connecter AWS et créer un AWS environnement avec des paramètres par défaut préconfigurés dans Select Régions AWS, utilisez S'inscrire pour. AWS

Conditions préalables

Si vous êtes nouveau AWS ou si vous n'avez pas créé de compte, il y a deux étapes principales pour le connecter AWS Toolkit for Visual Studio Code à votre AWS compte :

1. Configuration de l'authentification : Il existe 3 méthodes principales pour vous authentifier avec votre AWS compte à partir du AWS Toolkit for Visual Studio Code. Pour en savoir plus sur chacune de ces méthodes, consultez la rubrique Authentification et accès de ce guide de l'utilisateur.
2. Authentification AWS depuis le kit d'outils : vous pouvez vous connecter à votre AWS compte depuis le kit d'outils en suivant les procédures décrites dans les sections suivantes de ce guide de l'utilisateur.

Ouverture du panneau de connexion

Effectuez l'une des procédures suivantes pour ouvrir le panneau de connexion AWS à la boîte à outils.

Pour ouvrir le AWS Connectez-vous à la boîte à outils panneau du AWS Explorateur :

1. À partir du AWS Toolkit for Visual Studio Code, développez EXPLORER.
2. Élargissez le nombre d'actions supplémentaires... menu en sélectionnant le... icône.
3. Parmi les autres actions... dans le menu, choisissez Connect pour AWS ouvrir le panneau de connexion du AWS kit d'outils.

Pour ouvrir le AWS Connectez-vous à la boîte à outils panneau utilisant la palette de commandes VS Code :

1. Ouvrez la palette de commandes en appuyant sur **`Shift+Command+P`** (**`Ctrl+Shift+P`**Windows).
2. Entrez **`AWS: Add a New Connection`** dans le champ de recherche.
3. Sélectionnez cette option **`AWS: Add a New Connection`** pour ouvrir le panneau de connexion à la AWS boîte à outils.

Connexion à AWS à partir de la boîte à outils

Authentifiez-vous et connectez-vous avec le SSO

Pour vous authentifier et vous connecter en AWS utilisant AWS IAM Identity Center, procédez comme suit.

> **Note**
>
> L'authentification avec AWS Builder ID ou IAM Identity Center lance le portail AWS d'autorisation dans votre navigateur Web par défaut. Chaque fois que vos informations d'identification expirent, ce processus doit être répété pour renouveler la connexion entre votre AWS compte et le AWS Toolkit for Visual Studio Code.

Authentifiez-vous et connectez-vous avec AWS IAM Identity Center

1. Dans le panneau de connexion à la AWS boîte à outils, choisissez l'onglet Main-d'œuvre, puis cliquez sur le bouton Continuer pour continuer.
2. Dans le panneau Se connecter avec IAM Identity Center, entrez l'URL de démarrage de votre organisation. Cette URL vous est fournie par un administrateur ou un service d'assistance de votre entreprise.
3. Sélectionnez votre AWS région dans le menu déroulant. Il s'agit de la AWS région qui héberge votre répertoire d'identités.
4. Cliquez sur le bouton Continuer et confirmez que vous souhaitez ouvrir le site Web de demande AWS d'autorisation dans votre navigateur Web par défaut.
5. Suivez les instructions de votre navigateur Web par défaut, vous êtes averti lorsque le processus d'autorisation est terminé, vous pouvez fermer votre navigateur en toute sécurité et revenir à VS Code.

Authentifiez-vous et connectez-vous avec les informations d'identification IAM

Pour vous authentifier et vous connecter à AWS l'aide des informations d'identification IAM, procédez comme suit.

Authentifiez-vous et connectez-vous avec les informations d'identification IAM

1. Dans le panneau de connexion au AWS kit d'outils, choisissez IAM Credential, puis cliquez sur le bouton Continuer pour continuer.
2. Entrez le **`Profile NameAccess Key`**, et **`Secret Key`** de votre AWS compte dans les champs fournis, puis cliquez sur le bouton Continuer pour ajouter le profil à votre fichier de configuration et connecter le Toolkit à votre AWS compte.

3. Le Toolkit AWS Explorer est mis à jour pour afficher vos AWS services et ressources lorsque l'authentification est terminée et qu'une connexion a été établie.

Authentification pour Amazon CodeCatalyst

Pour commencer à travailler avec CodeCatalyst le Toolkit, authentifiez-vous et connectez-vous avec votre identifiant AWS Builder ID ou IAM Identity Center.

Les procédures suivantes décrivent comment authentifier et connecter le Toolkit à votre AWS compte.

Authentifiez-vous et connectez-vous avec un AWS ID de créateur

1. Dans le panneau de connexion à la AWS boîte à outils, choisissez l'onglet Main-d'œuvre, puis cliquez sur le bouton Continuer pour continuer.
2. En haut du panneau Se connecter avec SSO, cliquez sur le lien Passer à la connexion.
3. Suivez les instructions de votre navigateur Web par défaut, vous êtes averti lorsque le processus d'autorisation est terminé, vous pouvez fermer votre navigateur en toute sécurité et revenir à VS Code.

Authentifiez-vous et connectez-vous à IAM Identity Center

1. Dans le panneau de connexion à la AWS boîte à outils, choisissez l'onglet Main-d'œuvre, puis cliquez sur le bouton Continuer pour continuer.
2. Dans le panneau Se connecter avec IAM Identity Center, entrez l'URL de démarrage de votre organisation. Cette URL vous est fournie par un administrateur ou un service d'assistance de votre entreprise.
3. Sélectionnez votre AWS région dans le menu déroulant. Il s'agit de la AWS région qui héberge votre répertoire d'identités.
4. Cliquez sur le bouton Continuer et confirmez que vous souhaitez ouvrir le site Web de demande AWS d'autorisation dans votre navigateur Web par défaut.
5. Suivez les instructions de votre navigateur Web par défaut, vous êtes averti lorsque le processus d'autorisation est terminé, vous pouvez fermer votre navigateur en toute sécurité et revenir à VS Code.

Changements de AWS régions

Une AWS région indique où vos AWS ressources sont gérées. Votre AWS région par défaut est détectée lorsque vous vous connectez à votre AWS compte depuis le AWS Toolkit for Visual Studio Code, et elle s'affiche automatiquement dans l'AWS explorateur.

Les sections suivantes décrivent comment ajouter ou masquer une région dans l'AWS explorateur.

Ajouter une région à l' AWS explorateur

Procédez comme suit pour ajouter une région à l' AWS explorateur.

1. Dans VS Code, ouvrez la palette de commandes en développant Afficher dans le menu principal et en choisissant Palette de commandes. Ou utilisez les touches de raccourci suivantes :
   - Windows et Linux — Appuyez sur**`Ctrl+Shift+P`**.
   - macOS — Appuyez sur**`Shift+Command+P`**.
2. Dans la palette de commandes, recherchez **`AWS: Show or Hide Regions`** et choisissez AWS: Afficher ou masquer les régions pour afficher la liste des régions disponibles.
3. Dans la liste, sélectionnez les AWS régions que vous souhaitez ajouter à l'AWS explorateur.
4. Cliquez sur le bouton OK pour confirmer vos choix et mettre à jour l'AWS explorateur.

Masquer une région dans l' AWS explorateur

Pour masquer une région dans la vue AWS Explorateur, procédez comme suit.

1. Dans l'AWS explorateur, localisez la AWS région que vous souhaitez masquer.
2. Ouvrez le menu contextuel correspondant (cliquez avec le bouton droit) à la région que vous souhaitez masquer.
3. Choisissez Afficher ou masquer les régions pour ouvrir les options AWS: Afficher ou masquer les régions dans VS Code.
4. Désélectionnez les régions que vous souhaitez masquer dans la vue AWS Explorateur.

Configuration de votre chaîne d'outils

Il AWS Toolkit for Visual Studio Code prend en charge plusieurs langues dans tous les AWS services. Les sections suivantes décrivent comment configurer votre chaîne d'outils pour différentes langues.

Configuration d'une chaîne d'outils pour .NET Core

1. Assurez-vous que le AWS Toolkit for VS Code est installé.
2. Installez l'extension C#. Cette extension permet à VS Code de déboguer les applications .NET Core.
3. Ouvrez une AWS Serverless Application Model (AWS SAM) application ou créez-en une.
4. Ouvrez le dossier qui contient `template.yaml`.

Configurer une chaîne d'outils pour Node.js

1. Assurez-vous que le AWS Toolkit for VS Code est installé.
2. Ouvrez une AWS SAM application ou créez-en une.
3. Ouvrez le dossier qui contient `template.yaml`.

> **Note**
>
> Lors du débogage d'une fonction TypeScript Lambda directement à partir du code source (configuration `"target": "code"` de lancement), TypeScript le compilateur doit être installé soit globalement, soit dans celui de votre projet. `package.json`

Configuration d'une chaîne d'outils pour Python

1. Assurez-vous que le AWS Toolkit for VS Code est installé.
2. Installez l'extension Python pour Visual Studio Code. Cette extension permet à VS Code de déboguer des applications Python.
3. Ouvrez une AWS SAM application ou créez-en une.
4. Ouvrez le dossier qui contient `template.yaml`.

5. Ouvrez un terminal à la racine de votre application et configurez `virtualenv` en exécutant `python -m venv ./.venv`.

> **Note**
>
> Vous n'avez besoin de configurer `virtualenv` qu'une seule fois par système.

6. Activez `virtualenv` en exécutant l'un des scripts suivants :

   - Linux ou macOS (Bash) : `source ./.venv/bin/activate`
   - Fenêtres (PowerShell) : `./.venv/Scripts/Activate.ps1`
   - Windows (invite de commandes) : `.venv\Scripts\activate.bat`

Configuration d'une chaîne d'outils pour Java

1. Assurez-vous que le AWS Toolkit for VS Code est installé.
2. Installez l'extension Java et Java 11. Cette extension permet à VS Code de reconnaître les fonctions Java.
3. Installez l'extension Java Debugger. Cette extension permet à VS Code de déboguer des applications Java.
4. Ouvrez une AWS SAM application ou créez-en une.
5. Ouvrez le dossier qui contient `template.yaml`.

Configurer une chaîne d'outils pour Go

1. Assurez-vous que le AWS Toolkit for VS Code est installé.
2. Go 1.14 ou supérieur est requis pour le débogage des fonctions Go Lambda.
3. Installez l'extension Go.

> **Note**
>
> La version 0.25.0 ou supérieure est requise pour le débogage Go1.15 et les environnements d'exécution.

4. Installez les outils Go à l'aide de la palette de commandes :

a. Dans la palette de commandes, choisissez`Go: Install/Update Tools`.

b. Dans l'ensemble de cases à cocher, sélectionnez `dlv` et`gopls`.

5. Ouvrez une AWS SAM application ou créez-en une.
6. Ouvrez le dossier qui contient `template.yaml`.

Utilisation de votre chaîne d'outils

Une fois votre chaîne d'outils configurée, vous pouvez l'utiliser pour exécuter ou déboguer l' AWS SAM application.

Authentification et accès pour AWS Toolkit for Visual Studio Code

Vous n'avez pas besoin de vous authentifier AWS pour commencer à utiliser le AWS Toolkit for Visual Studio Code. Cependant, la plupart des AWS ressources sont gérées par le biais d'un AWS compte. Pour accéder à tous les AWS Toolkit for Visual Studio Code services et fonctionnalités, vous devez vous authentifier avec AWS IAM Identity Center des informations d'identification AWS Builder ID ou IAM.

Les rubriques suivantes contiennent des informations supplémentaires sur chaque type d'identifiant.

Pour plus de détails sur la façon de vous connecter à AWS in AWS Toolkit for Visual Studio Code avec vos informations d'identification existantes, consultez la AWS rubrique Connexion à du présent guide de l'utilisateur.

Rubriques

- AWS IAM Identity Center
- AWS Informations d'identification IAM
- AWS Builder ID pour les développeurs
- Utilisation d'un processus d'identification externe
- Mettre à jour les pare-feux et les passerelles pour autoriser l'accès

AWS IAM Identity Center

AWS IAM Identity Center est la meilleure pratique recommandée pour gérer l'authentification de votre AWS compte.

Pour obtenir des instructions détaillées sur la configuration d'IAM Identity Center pour les kits de développement logiciel (SDK), consultez la section Authentification IAM Identity Center du Guide de référence des kits de développement logiciel (AWS SDK) et des outils.

Pour plus d'informations sur la manière d'authentifier et de connecter le AWS kit d'outils à vos informations d'identification IAM Identity Center existantes, consultez la AWS rubrique Se connecter à dans ce guide de l'utilisateur.

AWS Informations d'identification IAM

AWS Authentification des informations d'identification IAM avec votre AWS compte via des clés d'accès stockées localement.

Pour plus d'informations sur la manière d'authentifier et de connecter le AWS kit d'outils à vos informations d'identification AWS IAM existantes, consultez la AWS rubrique Se connecter à dans ce guide de l'utilisateur.

Les sections suivantes décrivent comment configurer les informations d'identification IAM pour vous authentifier auprès de votre AWS compte depuis le. AWS Toolkit for Visual Studio Code

> **Important**
>
> Avant de configurer les informations d'identification IAM pour vous authentifier auprès de votre AWS compte, notez que :
>
> - Si vous avez déjà défini des informations d'identification IAM via un autre AWS service (tel que AWS CLI), il détecte AWS Toolkit for Visual Studio Code automatiquement ces informations d'identification et les rend disponibles dans VS Code.
> - AWS recommande d'utiliser l'authentification IAM Identity Center. Pour plus d'informations sur les meilleures pratiques AWS IAM, consultez la section Meilleures pratiques en matière de sécurité dans IAM du Guide de l'utilisateur de la gestion des AWS identités et des accès.
> - Afin d'éviter les risques de sécurité, n'employez pas les utilisateurs IAM pour l'authentification lorsque vous développez des logiciels spécialisés ou lorsque vous travaillez avec des données réelles. Utilisez plutôt la fédération avec un fournisseur d'identité, comme indiqué dans la section Qu'est-ce que l'IAM Identity Center ? du Guide de l'utilisateur AWS IAM Identity Center .

Création d'un utilisateur IAM

Avant de pouvoir configurer l' AWS Toolkit for Visual Studio Code authentification avec votre AWS compte, vous devez terminer l'étape 1 : créer votre utilisateur IAM et l'étape 2 : obtenir vos clés d'accès dans la rubrique Authentifier à l'aide d'informations d'identification à long terme du Guide de référence des AWS kits de développement logiciel et des outils.

> **Note**
>
> Étape 3 : La mise à jour du fichier d'informations d'identification partagé dans le Guide de référence AWS des kits de développement logiciel et des outils est facultative.
> Si vous terminez l'étape 3, vos informations d'identification seront AWS Toolkit for Visual Studio Code automatiquement détectées au cours de the section called "Création d'un fichier d'informations d'identification partagé à partir du AWS Toolkit for Visual Studio Code" la procédure ci-dessous.
> Si vous n'avez pas terminé l'étape 3, il vous AWS Toolkit for Visual Studio Code guide tout au long du processus de création d'un, `credentials file` comme the section called "Création d'un fichier d'informations d'identification partagé à partir du AWS Toolkit for Visual Studio Code" décrit ci-dessous.

Création d'un fichier d'informations d'identification partagé à partir du AWS Toolkit for Visual Studio Code

Votre fichier de configuration partagé et votre fichier d'informations d'identification partagé stockent les informations de configuration et d'identification de vos AWS comptes. Pour plus d'informations sur la configuration et les informations d'identification partagées, consultez la section Où sont stockés les paramètres de configuration ? du Guide de l'utilisateur AWS Command Line Interface .

Création d'un fichier d'informations d'identification partagé via le AWS Toolkit for Visual Studio Code

1. Ouvrez la palette de commandes en appuyant sur **`Shift+Command+P`** (**`Ctrl+Shift+P`**Windows).
2. Entrez **`AWS: Add a New Connection`** dans le champ de recherche.
3. Sélectionnez cette option **`AWS: Add a New Connection`** pour ouvrir le panneau de connexion à la AWS boîte à outils.
4. Dans le panneau de connexion du AWS Toolkit, choisissez IAM Credential, puis cliquez sur le bouton Continuer pour continuer.
5. Entrez le **`Profile NameAccess Key`**, et **`Secret Key`** de votre AWS compte dans les champs prévus à cet effet, puis cliquez sur le bouton Continuer pour ajouter le profil à votre fichier de configuration et connecter le Toolkit à votre AWS compte.
6. Le Toolkit AWS Explorer est mis à jour pour afficher vos AWS services et ressources lorsque l'authentification est terminée et qu'une connexion a été établie.

> **Note**
>
> Dans cet exemple, supposons qu'il *[Profile_Name]* contient des erreurs de syntaxe et entraîne l'échec de l'authentification.
>
> ```
> [Profile_Name]
> xaws_access_key_id= AKIAI44QH8DHBEXAMPLE
> xaws_secret_access_key= wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
> ```
>
> Voici un exemple de message de journal généré en réponse à une tentative d'authentification échouée.
>
> ```
> 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.
> ```

Ajouter des profils d'identification supplémentaires

Vous pouvez ajouter plusieurs informations d'identification à vos fichiers de configuration. Pour ce faire, ouvrez la palette de commandes et choisissez AWS Toolkit Create Credentials Profile. Cela ouvrira le fichier d'informations d'identification. Sur cette page, vous pouvez ajouter un nouveau profil en dessous de votre premier profil, comme illustré dans l'exemple suivant :

```
# 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 pour les développeurs

Un AWS Builder ID est un AWS compte supplémentaire facultatif ou obligatoire pour certains AWS services. Pour obtenir des informations détaillées sur la méthode d'authentification AWS Builder ID, consultez la rubrique Se connecter avec AWS Builder ID dans le Guide de l'utilisateur de AWS connexion.

Pour plus d'informations sur la façon d'authentifier et de connecter le AWS kit d'outils à votre identifiant AWS Builder existant, consultez la AWS rubrique Connect to de ce guide de l'utilisateur.

Utilisation d'un processus d'identification externe

Vous pouvez configurer les processus AWS Toolkit for Visual Studio Code d'identification qui ne sont pas directement pris en charge par AWS, en modifiant votre`shared config file`.

La modification `shared config file` de vos processus d'identification est la même pour le AWS Toolkit for Visual Studio Code et le AWS Command Line Interface. Pour des informations détaillées sur la configuration des informations d'identification externes, consultez la rubrique Sourcing des informations d'identification avec un processus externe dans le Guide de AWS Command Line Interface l'utilisateur.

Mettre à jour les pare-feux et les passerelles pour autoriser l'accès

Si vous filtrez l'accès à des AWS domaines ou à des points de terminaison d'URL spécifiques à l'aide d'une solution de filtrage de contenu Web, les points de terminaison suivants doivent être autorisés dans la liste afin d'accéder à tous les services et fonctionnalités disponibles via Amazon Q. AWS Toolkit for Visual Studio Code

AWS Toolkit for Visual Studio Code Points de terminaison

Vous trouverez ci-dessous des listes de points de terminaison et de références AWS Toolkit for Visual Studio Code spécifiques qui doivent être autorisés.

Endpoint

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

Fichiers hébergés

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

Support de schéma

```
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
```

cSharpSamScript d'installation de débogage

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

Points de terminaison du plugin Amazon Q

Vous trouverez ci-dessous une liste des points de terminaison et des références spécifiques au plugin Amazon Q qui doivent être autorisés.

```
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)
```

Points de terminaison Amazon Q pour développeurs

Vous trouverez ci-dessous une liste des points de terminaison et des références spécifiques à Amazon Q Developer qui doivent être autorisés.

```
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)
```

Points de terminaison Amazon Q Code Transform

Vous trouverez ci-dessous une liste des points de terminaison et des références spécifiques à Amazon Q Code Transform qui doivent être autorisés.

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

Points de terminaison d'authentification

Vous trouverez ci-dessous une liste des points de terminaison et des références d'authentification qui doivent être autorisés.

```
[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
```

Points de terminaison d'identité

Les listes suivantes contiennent des points de terminaison spécifiques à l'identité, tels que le AWS IAM Identity Center AWS Builder ID.

AWS IAM Identity Center

Pour plus de détails sur les points de terminaison requis pour IAM Identity Center, consultez la rubrique Activer le centre d'identité IAM dans le guide de l'AWS IAM Identity Centerutilisateur.

Centre d'identité IAM d'entreprise

```
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 ID du constructeur

```
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)
```

Télémétrie

Vous trouverez ci-dessous une liste des points de terminaison spécifiques à la télémétrie qui doivent être autorisés.

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

Références

Vous trouverez ci-dessous une liste de références de points de terminaison.

```
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.
```

Travailler avec AWS services et outils

Il AWS Toolkit for Visual Studio Code met à votre disposition des AWS services, des outils et des ressources, directement dans VS Code. Vous trouverez ci-dessous une liste de rubriques du guide couvrant chaque service Toolkit for VS Code et ses fonctionnalités. Choisissez un service ou un outil pour plus d'informations sur son fonctionnement, sa configuration et son utilisation des fonctionnalités de base.

Rubriques

- Travailler avec des fonctionnalités expérimentales
- Utilisation des AWS services dans l' AWS explorateur
- AWS Documents
- Amazon CodeCatalyst pour VS Code
- Utilisation d'Amazon API Gateway
- Utilisation AWS App Runner avec AWS Toolkit for Visual Studio Code
- AWS Générateur d'applications
- AWS Compositeur d'infrastructure
- AWS CDK pour VS Code
- Travailler avec des AWS CloudFormation piles
- Utilisation des CloudWatch journaux à l'aide du AWS Toolkit for Visual Studio Code
- Amazon DocumentDB
- Amazon Elastic Compute Cloud
- Utilisation d'Amazon Elastic Container Registry
- Fonctionnement d'Amazon Elastic Container Service
- Travailler avec Amazon EventBridge
- AWS Analyseur d'accès IAM
- Travailler avec AWS IoT in AWS Toolkit for Visual Studio Code
- AWS Lambda Fonctions
- Amazon Redshift dans le kit d'outils pour VS Code
- Travailler avec Amazon S3
- Amazon SageMaker Unified Studio pour VS Code

- Utilisation d'applications sans serveur
- Utilisation des documents d'automatisation de Systems Manager
- AWS Step Functions
- Utilisation de Threat Composer
- Utilisation des ressources

Travailler avec des fonctionnalités expérimentales

Les fonctionnalités expérimentales offrent un accès anticipé aux fonctionnalités AWS Toolkit for Visual Studio Code avant leur sortie officielle.

> **Warning**
>
> Comme les fonctionnalités expérimentales continuent d'être testées et mises à jour, elles peuvent présenter des problèmes d'utilisation. Et les fonctionnalités expérimentales peuvent être supprimées du AWS Toolkit for Visual Studio Code sans préavis.

Vous pouvez activer des fonctionnalités expérimentales pour des AWS services spécifiques dans la section AWS Toolkit du volet Paramètres de votre IDE VS Code.

1. Pour modifier AWS les paramètres dans VS Code, choisissez Fichier, Préférences, Paramètres.
2. Dans le volet Paramètres, développez Extensions et choisissez AWS Toolkit.
3. Sous AWS: Expériences, cochez les cases correspondant aux fonctionnalités expérimentales auxquelles vous souhaitez accéder avant la publication. Si vous voulez désactiver une fonctionnalité expérimentale, décochez la case correspondante.

Utilisation des AWS services dans l' AWS explorateur

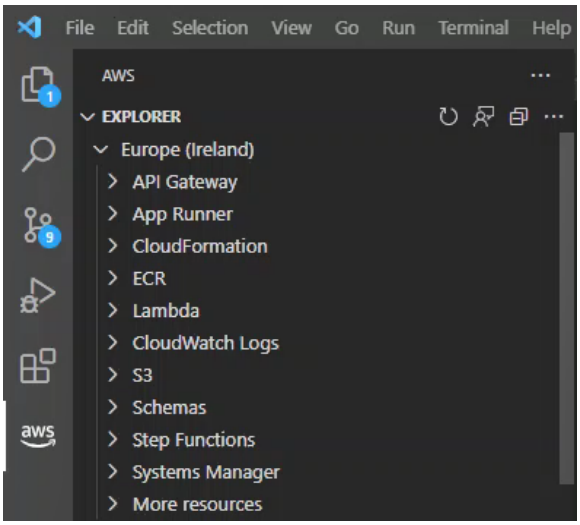
L'AWS explorateur vous donne un aperçu de certains des AWS services avec lesquels vous pouvez travailler lorsque vous utilisez le AWS Toolkit for Visual Studio Code.

Cette section fournit des informations sur l'accès et l'utilisation de l'AWS explorateur dans VS Code. Cela suppose que vous avez déjà installé et configuré le Toolkit for VS Code sur votre système.

Quelques points importants :

- Si le kit d'outils est correctement installé et configuré, vous devriez voir des éléments dans l'explorateur.AWS Pour voir l'AWS explorateur, cliquez AWSsur l'icône dans la barre d'activité.

  Par exemple :



- Certaines fonctionnalités nécessitent certaines AWS autorisations. Par exemple, pour voir les AWS Lambda fonctions de votre AWS compte, les informations d'identification que vous avez configurées Authentification et accès doivent inclure au moins des autorisations Lambda en lecture seule. Consultez les rubriques suivantes pour plus d'informations sur les autorisations dont chaque fonctionnalité a besoin.
- Si vous souhaitez interagir avec AWS des services qui ne sont pas immédiatement visibles dans l'AWS explorateur, vous pouvez accéder à Plus de ressources et choisir parmi des centaines de ressources qui peuvent être ajoutées à l'interface.

  Par exemple, vous pouvez choisir AWS Toolkit : CodeArtifact : :Repository parmi les types de ressources disponibles. Une fois ce type de ressource ajouté à Plus de ressources, vous pouvez développer l'entrée pour afficher une liste de ressources qui créent différents CodeArtifact référentiels dotés de leurs propres propriétés et attributs. En outre, vous pouvez décrire les propriétés et les attributs des ressources dans des modèles au format JSON, qui peuvent être enregistrés pour créer de nouvelles ressources dans le Cloud. AWS

AWS Documents

Il AWS Toolkit for Visual Studio Code prend en charge le AWS Serverless Application Model `JSON Schema` for`AWS SAM templates`, améliorant l'expérience de création de modèles en activant les définitions, l'autocomplétion et la validation directement dans VS Code. AWS Documents prend

en charge toutes AWS SAM les CloudFormation ressources. Pour plus de détails, consultez les ressources suivantes :

- Pour des informations spécifiques sur le schéma JSON, consultez le JSON-Schema.org site Web du schéma JSON.
- Pour plus d'informations sur les AWS SAM modèles, consultez la rubrique relative à l'anatomie des AWS SAM modèles dans le Guide du AWS Serverless Application Modeldéveloppeur.
- Pour plus d'informations sur les AWS ressources et les types de propriétés, consultez la rubrique de référence des types de AWS ressources et de propriétés dans le Guide de CloudFormationl'utilisateur.
- Pour des informations détaillées sur le AWS SAM schéma utilisé par le AWS Toolkit, consultez le AWS Serverless Application Modelschéma dans le AWS GitHub référentiel.

Mise en route avec AWS Documents

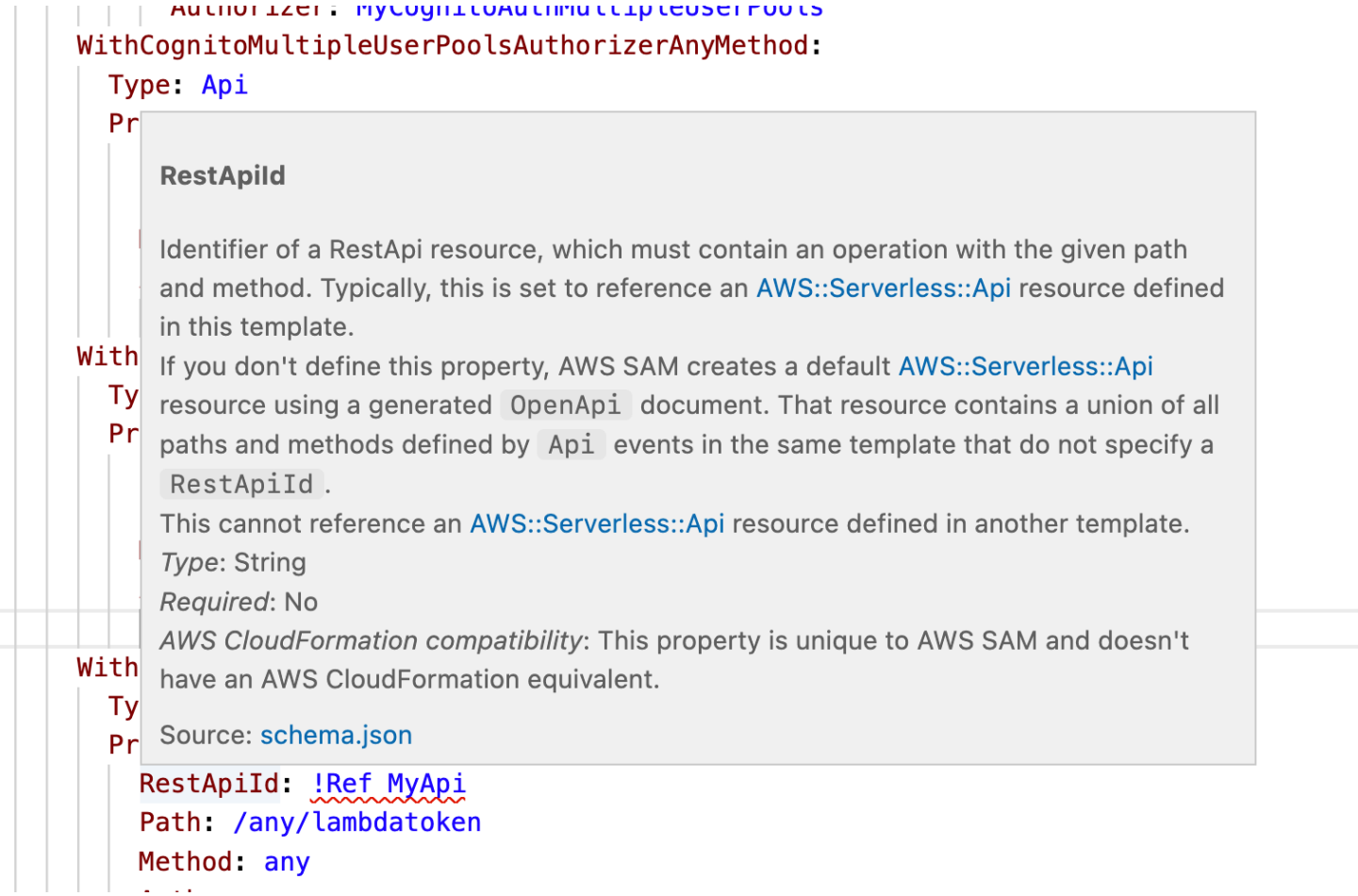
Pour commencer à travailler avec AWS des documents dans VS Code, installez l' AWS Toolkit for Visual Studio Code extension depuis votre IDE ou VS Code Marketplace, et ouvrez n'importe quel AWS SAM modèle.

Affichage de la documentation, de l'autocomplétion et de la validation dans VS Code

L'affichage de la documentation, l'autocomplétion et la validation sont des fonctionnalités incluses dans le AWS kit d'outils. Voir l'image ci-dessous pour un exemple de ce à quoi ressemblent ces fonctionnalités dans VS Code.

- Pour afficher la documentation à partir de votre AWS SAM modèle ouvert, placez le pointeur sur une entrée de ligne du document.
- Pour l'autocomplétion, commencez à taper dans votre AWS SAM modèle pour activer une fenêtre contextuelle contenant des suggestions basées sur vos entrées.
- Votre AWS SAM modèle est automatiquement scanné pour validation et les erreurs sont mises en évidence par une icône représentant une ampoule que vous pouvez sélectionner pour obtenir des suggestions supplémentaires.

Voir l'image ci-dessous pour un exemple de ce à quoi ressemblent ces fonctionnalités dans VS Code.



Amazon CodeCatalyst pour VS Code

Qu'est-ce qu'Amazon CodeCatalyst ?

Amazon CodeCatalyst est un espace de collaboration basé sur le cloud pour les équipes de développement de logiciels. Grâce au AWS Toolkit for Visual Studio Code, vous pouvez consulter et gérer vos CodeCatalyst ressources directement depuis VS Code. Vous pouvez également travailler directement dans le cloud en utilisant la AWS boîte à outils pour lancer des environnements informatiques virtuels Dev Environments exécutant VS Code. Pour plus d'informations sur le CodeCatalyst service, consultez le guide de CodeCatalyst l'utilisateur Amazon.

Les rubriques suivantes décrivent comment connecter VS Code à CodeCatalyst et comment utiliser le Toolkit for VS Code. CodeCatalyst

Rubriques

- Démarrage CodeCatalyst et boîte à outils pour VS Code

- Utilisation des CodeCatalyst ressources Amazon dans VS Code
- Utilisation de la boîte à outils dans un environnement de développement
- Résolution des problèmes liés à Amazon CodeCatalyst et à VS Code

Démarrage CodeCatalyst et boîte à outils pour VS Code

Pour commencer à travailler avec CodeCatalyst VS Code, suivez ces procédures.

Rubriques

- Création d'un CodeCatalyst compte
- Connecter le AWS Boîte à outils avec CodeCatalyst

Création d'un CodeCatalyst compte

Vous devez disposer d'un ID AWS Builder ou AWS IAM Identity Center d'informations d'identification actifs pour vous connecter à CodeCatalyst partir du Toolkit for VS Code. Pour en savoir plus sur AWS Builder ID, IAM Identity Center et les CodeCatalyst informations d'identification, consultez la CodeCatalyst section Configuration avec du Guide de l'CodeCatalystutilisateur.

Connecter le AWS Boîte à outils avec CodeCatalyst

Pour connecter le AWS Toolkit à votre CodeCatalyst compte, consultez la CodeCatalyst section Authentification pour Amazon dans la AWS rubrique Connexion à de ce guide de l'utilisateur.

Utilisation des CodeCatalyst ressources Amazon dans VS Code

Les sections suivantes fournissent une vue d'ensemble des fonctionnalités de gestion des CodeCatalyst ressources Amazon disponibles dans le Toolkit for VS Code.

Pour plus d'informations sur les environnements de développement et sur la manière d'y accéder CodeCatalyst, consultez la section Environnements de développement du guide de CodeCatalyst l'utilisateur Amazon.

Les sections suivantes décrivent comment créer, ouvrir et utiliser des environnements de développement à partir de VS Code.

Rubriques

- Clonage d'un référentiel
- Ouverture d'un environnement de développement
- Création d'un CodeCatalyst environnement de développement
- Création d'un environnement de développement à partir d'un référentiel tiers
- CodeCatalyst commandes dans VS Code

Clonage d'un référentiel

CodeCatalyst est un service basé sur le cloud qui nécessite que vous soyez connecté au cloud pour travailler sur CodeCatalyst des projets. Si vous préférez travailler sur un projet localement, vous pouvez cloner vos CodeCatalyst référentiels sur votre machine locale et les synchroniser avec votre CodeCatalyst projet en ligne lors de votre prochaine connexion au cloud.

Pour cloner un référentiel depuis votre CodeCatalyst compte vers VS Code à l'aide du AWS Toolkit, procédez comme suit :

> **Note**
>
> Si vous clonez un référentiel à partir d'un service tiers, vous pouvez être invité à vous authentifier à l'aide des informations d'identification de ce service.
> Pendant le clonage du référentiel, VS Code affiche la progression dans la fenêtre d'état du référentiel de clonage. Une fois le référentiel cloné, le champ Voulez-vous ouvrir le référentiel cloné ? un message apparaît.

1. À partir de la boîte à outils pour VS Code, développez l'explorateur DEVELOPER TOOLS.
2. Développez CodeCatalyst, choisissez Clone Repository.
3. Dans la boîte de dialogue Sélectionner un CodeCatalyst référentiel, recherchez le référentiel que vous souhaitez cloner, puis sélectionnez-le pour ouvrir la boîte de dialogue Choisir un dossier à cloner.
4. Choisissez Sélectionner l'emplacement du référentiel pour fermer l'invite et commencer à cloner le référentiel.
5. Dans la fenêtre de dialogue, choisissez l'une des options suivantes pour terminer le processus de clonage :

   - Pour ouvrir votre référentiel dans votre fenêtre VS Code actuelle, choisissez Ouvrir.

- Pour ouvrir votre référentiel dans une nouvelle fenêtre VS Code, choisissez Ouvrir dans une nouvelle fenêtre.
- Pour terminer le processus de clonage sans ouvrir votre référentiel, fermez la fenêtre de dialogue.

Ouverture d'un environnement de développement

Pour ouvrir un environnement de développement existant dans VS Code, procédez comme suit.

> **Note**
>
> La sélection de l'environnement de développement lance le processus de connexion à VS Code CodeCatalyst en ouvrant votre environnement de développement. Au cours de ce processus, VS Code affiche les mises à jour de progression dans une fenêtre d' CodeCatalyst état. La fenêtre d'état est mise à jour lorsque le processus est terminé.
>
> - Si l'environnement de développement ne s'ouvre pas, l'état est mis à jour avec des informations sur les raisons de l'échec du processus et un lien permettant d'ouvrir les journaux de processus.
> - Si le processus aboutit, votre environnement de développement s'ouvre dans une nouvelle fenêtre, depuis VS Code.

1. À partir de la boîte à outils pour VS Code, développez l'explorateur DEVELOPER TOOLS.
2. Développez CodeCatalyst et choisissez Open Dev Environment pour ouvrir la boîte de dialogue Sélectionner un environnement de CodeCatalyst développement dans VS Code.
3. Dans la boîte de dialogue CodeCatalyst Sélectionnez un environnement de développement, choisissez l'environnement de développement que vous souhaitez ouvrir.

Création d'un CodeCatalyst environnement de développement

Pour créer un nouvel environnement de développement, procédez comme suit :

> **Note**
>
> Lorsque vous créez un nouvel environnement de développement, observez les points suivants :

- AWS vous recommande de spécifier un alias car cela simplifie l'organisation et améliore les capacités de recherche pour les environnements de développement.
- Dev Environments enregistre votre travail en permanence. Cela signifie que votre environnement de développement peut être arrêté sans perdre votre travail. L'arrêt de votre environnement de développement réduit les coûts nécessaires pour maintenir votre environnement de développement opérationnel.
- Le stockage est le seul paramètre qui ne peut pas être modifié une fois que votre environnement de développement a été créé.
- VS Code affiche la progression de la création de votre environnement de développement dans une fenêtre d'état. Une fois l'environnement de développement créé, VS Code ouvre l'environnement de développement dans une nouvelle fenêtre et le bouton Faites-vous confiance aux auteurs des fichiers de ce dossier ? une invite apparaît également. Acceptez les termes et conditions pour continuer à travailler dans votre environnement de développement.

1. À partir de la boîte à outils pour VS Code, développez l'explorateur DEVELOPER TOOLS.
2. Développez CodeCatalyst et choisissez l'option Créer un environnement de développement pour ouvrir le menu Créer CodeCatalyst un environnement de développement dans VS Code.
3. Dans la section Code source, choisissez l'une des options suivantes :

   - Utiliser un CodeCatalyst référentiel existant : crée un environnement de développement à partir d'un CodeCatalyst référentiel existant. Vous devez sélectionner le CodeCatalyst projet et la branche.
   - Créer un environnement de développement vide : crée un environnement de développement vide.
4. (Facultatif) Dans la section Alias, entrez un autre nom pour votre environnement de développement.
5. (Facultatif) Dans la section Configuration des environnements de développement, modifiez les paramètres suivants pour répondre à vos besoins spécifiques.

   - Calcul : choisissez Modifier le calcul pour modifier la quantité de puissance de traitement et de RAM attribuée à votre système.
   - Délai d'attente : choisissez Modifier le délai d'inactivité pour modifier la durée d'inactivité du système autorisée avant l'arrêt de votre environnement de développement.

- Stockage : choisissez Modifier la taille de stockage pour modifier la quantité d'espace de stockage attribuée à votre système.

6. Choisissez Créer un environnement de développement pour créer votre nouvel environnement de développement cloud.

Création d'un environnement de développement à partir d'un référentiel tiers

Vous pouvez créer des environnements de développement à partir d'un référentiel tiers en établissant un lien vers le référentiel en tant que source.

Les liens vers un référentiel tiers en tant que source sont gérés au niveau du projet dans CodeCatalyst. Pour obtenir des instructions et des détails supplémentaires sur la façon de connecter un référentiel tiers à votre environnement de développement, consultez la rubrique Associer un référentiel source dans le Guide de CodeCatalyst l'utilisateur Amazon.

CodeCatalyst commandes dans VS Code

D'autres commandes VS Code sont attribuées à des CodeCatalyst-related fonctionnalités qui ne sont pas affichées directement dans le AWS Toolkit.

Pour afficher la liste des commandes attribuées à CodeCatalyst partir de la palette de commandes, procédez comme suit :

1. À partir de la boîte à outils pour VS Code, développez l'explorateur DEVELOPER TOOLS.
2. Choisissez Afficher CodeCatalyst les commandes pour ouvrir la palette de commandes avec une recherche `CodeCatalyst` préremplie.
3. Sélectionnez une CodeCatalyst commande dans la liste pour l'activer.

Utilisation de la boîte à outils dans un environnement de développement

Les environnements de développement sont des environnements informatiques virtuels pour Amazon CodeCatalyst. Les sections suivantes décrivent comment créer, lancer et utiliser des environnements de développement à l'aide du AWS Toolkit for Visual Studio Code.

Pour obtenir des informations détaillées sur les environnements de développement, consultez la https://docs.aws.amazon.com/codecatalyst/latest/userguide/devenvironment.html rubrique Environnements de développement du Guide de CodeCatalyst l'utilisateur Amazon.

Configuration de votre environnement de développement avec devfiles

La `devfile` spécification est un format standard ouvert pour YAML qui peut être utilisé pour définir des configurations pour les environnements de développement. Chaque environnement de développement possède un devfile. Si vous créez un environnement de développement sans référentiel ou à partir d'un référentiel ne contenant pas de fichier de développement, une valeur par défaut est automatiquement appliquée à la source. Les fichiers de développement peuvent être mis à jour depuis CodeCatalyst ou depuis votre IDE. Les processus de mise à jour d'un fichier de développement dans une instance locale ou distante de VS Code sont identiques, mais si vous mettez à jour un fichier de développement localement, vous devez envoyer les mises à jour vers votre référentiel source avant qu'elles ne prennent effet.

Pour plus d'informations sur la configuration des environnements de développement à l'aide de devfiles, consultez la rubrique Configuration de votre environnement de développement dans le guide de CodeCatalyst l'utilisateur Amazon.

La procédure suivante décrit comment modifier votre fichier de développement à partir d'une instance distante du Toolkit lorsqu'il s'exécute dans un environnement de développement.

> **Important**
>
> Si vous modifiez le code `Devfile` depuis VS Code, tenez compte des points suivants :
>
> - La modification du nom du fichier devfile ou du nom du composant devfile remplace le contenu de votre répertoire racine. Tout le contenu précédent est perdu et irrécupérable.
> - Si vous créez un environnement de développement sans fichier de développement dans le dossier racine ou un environnement de développement qui n'est associé à aucun référentiel source, un fichier de développement avec les paramètres de configuration par défaut est généré pour votre environnement de développement lorsque vous le créez.
> - Pour obtenir des instructions sur la façon de définir et de configurer votre`Devfile`, consultez la documentation relative à l'ajout de commandes sur le site Web devfile.io.

1. À partir de la boîte à outils pour VS Code, développez l'explorateur DEVELOPER TOOLS.
2. Développez CodeCatalyst et choisissez Open Devfile pour l'ouvrir `devfile.yaml` dans une nouvelle fenêtre d'éditeur, dans votre environnement de développement actuel.
3. Depuis l'éditeur VS Code, mettez à jour votre devfile, puis enregistrez vos modifications.

4. La prochaine fois que vous lancerez votre environnement de développement, la configuration sera mise à jour pour correspondre aux spécifications définies dans votre`Devfile`.

Authentification et connexion à AWS depuis votre environnement de développement

Pour accéder à toutes vos AWS ressources depuis votre environnement de développement, vous devez vous authentifier et connecter votre instance distante du Toolkit à votre AWS compte. L'instance distante du Toolkit s'authentifie automatiquement à l'aide des informations d'identification héritées de votre instance locale du Toolkit lorsque votre environnement de développement est lancé.

Les procédures de mise à jour de vos informations d'identification pour une instance distante du Toolkit sont identiques à l'expérience d'authentification dans votre instance locale du Toolkit. Pour obtenir des instructions détaillées sur la façon de mettre à jour les informations d'identification, de s'authentifier et de se connecter à AWS partir du Toolkit, consultez la AWS section Connexion à de la rubrique Démarrage de ce guide de l'utilisateur.

Pour plus d'informations sur chacune des méthodes AWS d'authentification compatibles avec le AWS Toolkit for Visual Studio Code, consultez la rubrique Authentification et accès de ce guide de l'utilisateur.

Utilisation de la boîte à outils pour VS Code dans les environnements de développement

Après avoir ouvert ou créé un environnement de développement dans VS Code, vous pouvez utiliser le Toolkit pour VS Code, de la même manière que vous pouvez le faire à partir d'une instance locale de VS Code. Les environnements de développement exécutant VS Code sont configurés pour installer automatiquement le AWS Toolkit et se connecter à votre AWS Builder ID.

Arrêt d'un environnement de développement

Pour arrêter votre environnement de développement actuel :

1. À partir de la boîte à outils pour VS Code, développez l'explorateur DEVELOPER TOOLS.
2. Développez CodeCatalyst et choisissez Stop Dev Environment.
3. Lorsque VS Code vous y invite, confirmez que vous souhaitez arrêter votre environnement de développement.
4. Votre environnement de développement s'est arrêté avec succès lorsque VS Code ferme la connexion à distance et revient à une instance de développement locale.

Ouverture des paramètres d'un environnement de développement

Pour ouvrir les paramètres de votre environnement de développement actuel, procédez comme suit :

> **Note**
>
> Vous ne pouvez pas modifier la quantité d'espace de stockage attribuée à votre environnement de développement après sa création.

1. À partir de la boîte à outils pour VS Code, développez l'explorateur DEVELOPER TOOLS.
2. Développez CodeCatalyst et choisissez Ouvrir les paramètres pour ouvrir la vue Paramètres de l'environnement de développement, pour votre environnement de développement actuel.
3. Dans la vue Dev Environment Settings (Paramètres de l'environnement de développement), les sections suivantes contiennent des options pour votre environnement de développement :
   - Alias : visualisez et modifiez l'alias attribué à votre environnement de développement.
   - État : consultez l'état actuel de votre environnement de développement, le projet auquel il est affecté et arrêtez votre environnement.
   - Devfile : affiche le nom et l'emplacement du fichier correspondant `Devfile` à votre environnement de développement. Ouvrez le vôtre `Devfile` en cliquant sur le bouton Ouvrir dans l'éditeur.
   - Compute Settings (Paramètres de calcul) : modifiez la taille et la valeur Timeout Length (Délai d'attente) par défaut pour votre environnement de développement.

Résolution des problèmes liés à Amazon CodeCatalyst et à VS Code

Les rubriques suivantes abordent les problèmes techniques potentiels liés à l'utilisation d'Amazon CodeCatalyst et de VS Code.

Rubriques

- Version VS Code
- Autorisations pour Amazon CodeCatalyst
- Connexion à un environnement de développement à partir de la boîte à outils pour VS Code

Version VS Code

Votre version de VS Code est censée configurer un gestionnaire pour les `vscode://` URI sur votre système. Sans ce gestionnaire, vous ne pouvez pas accéder à toutes les CodeCatalyst fonctionnalités du AWS Toolkit. Par exemple, vous rencontrez une erreur lors du lancement d'un environnement de développement depuis VS Code Insiders. Cela est dû au fait que VS Code Insiders gère les `vscode-insiders://` URI et ne gère `vscode://` pas les URI.

Autorisations pour Amazon CodeCatalyst

Les autorisations requises pour travailler avec les fichiers suivants sont CodeCatalyst les AWS Toolkit for Visual Studio Code suivantes :

- Définissez vos propres autorisations d'accès pour votre `~/.ssh/config` fichier sur `read` et`write`. Restreignez les `write` autorisations pour tous les autres utilisateurs.
- Définissez vos autorisations d'accès pour les `~/.ssh/id_rsa` fichiers `~/.ssh/id_dsa` et `read` uniquement. Restreindre `read` `write` `execute` et autoriser tous les autres utilisateurs.
- Votre `globals.context.globalStorageUri.fsPath` fichier doit se trouver dans un emplacement accessible en écriture.

Connexion à un environnement de développement à partir de la boîte à outils pour VS Code

Si le message d'erreur suivant s'affiche lorsque vous tentez de vous connecter à un environnement de développement depuis AWS Toolkit for Visual Studio Code :

Votre *`~/.ssh/config` `aws-devenv-*`* section est peut-être obsolète.

- Choisissez la configuration Open... bouton pour ouvrir votre `~/.ssh/config` fichier dans l'éditeur de code VS.
- Dans l'éditeur, sélectionnez et supprimez le contenu de la `Host aws-devenv-*` section.
- Enregistrez les modifications que vous avez apportées au `Host aws-devenv-*` de`~/.ssh/config`. Fermez ensuite le fichier.
- Réessayez de vous connecter à un environnement de développement à partir du Toolkit for VS Code.

Utilisation d'Amazon API Gateway

Vous pouvez parcourir et exécuter des ressources API Gateway distantes sur votre AWS compte connecté à l'aide du AWS Toolkit for Visual Studio Code.

> **Note**
>
> Cette fonctionnalité ne prend pas en charge le débogage.

Pour parcourir et exécuter des ressources API Gateway distantes

1. Dans l'AWS explorateur, choisissez API Gateway pour développer le menu. Les ressources API Gateway distantes sont répertoriées.
2. Localisez la ressource API Gateway que vous souhaitez appeler, ouvrez son menu contextuel (clic droit), puis choisissez Invoke on AWS.
3. Dans le formulaire de paramètres, spécifiez les paramètres d'appel.
4. Pour exécuter la ressource API Gateway distante, choisissez Invoke. Les résultats sont affichés dans la vue VS Code Output.

Utilisation AWS App Runner avec AWS Toolkit for Visual Studio Code

AWS App Runnerfournit un moyen rapide, simple et rentable de déployer à partir du code source ou d'une image de conteneur directement vers une application Web évolutive et sécurisée dans le AWS cloud. Pour l'utiliser, vous n'avez pas besoin d'apprendre de nouvelles technologies, de choisir le service de calcul à utiliser ou de savoir comment provisionner et configurer les AWS ressources.

Vous pouvez l'utiliser AWS App Runner pour créer et gérer des services en fonction d'une image source ou d'un code source. Si vous utilisez une image source, vous pouvez choisir une image de conteneur publique ou privée qui est stockée dans un référentiel d'images. App Runner prend en charge les fournisseurs de référentiels d'images suivants :

- Amazon Elastic Container Registry (Amazon ECR) : stocke les images privées dans votre AWS compte.
- Amazon Elastic Container Registry Public (Amazon ECR Public) : stocke les images qui peuvent être lues publiquement.

Si vous choisissez l'option de code source, vous pouvez déployer à partir d'un référentiel de code source maintenu par un fournisseur de référentiel pris en charge. Actuellement, App Runner est compatible en GitHub tant que fournisseur de référentiel de code source.

Conditions préalables

Pour interagir avec App Runner à l'aide des éléments AWS Toolkit for Visual Studio Code suivants, vous devez :

- Un AWS compte
- Une version de AWS Toolkit for Visual Studio Code ces fonctionnalités AWS App Runner

En plus de ces exigences de base, assurez-vous que tous les utilisateurs IAM concernés ont l'autorisation d'interagir avec le service App Runner. Vous devez également obtenir des informations spécifiques sur votre source de service, telles que l'URI de l'image du conteneur ou la connexion au GitHub référentiel. Vous avez besoin de ces informations lors de la création de votre service App Runner.

Configuration des autorisations IAM pour App Runner

Le moyen le plus simple d'accorder les autorisations requises pour App Runner consiste à associer une politique AWS gérée existante à l'entité Gestion des identités et des accès AWS (IAM) concernée, en particulier à un utilisateur ou à un groupe. App Runner fournit deux politiques gérées que vous pouvez attacher à vos utilisateurs IAM :

- `AWSAppRunnerFullAccess` : permet aux utilisateurs d'effectuer toutes les actions d'App Runner.
- `AWSAppRunnerReadOnlyAccess` : permet aux utilisateurs de répertorier et d'afficher les détails des ressources d'App Runner.

En outre, si vous choisissez un référentiel privé dans Amazon Elastic Container Registry (Amazon ECR) comme source de service, vous devez créer le rôle d'accès suivant pour votre service App Runner :

- `AWSAppRunnerServicePolicyForECRAccess` : permet à App Runner d'accéder aux images Amazon Elastic Container Registry (Amazon ECR) dans votre compte.

Vous pouvez créer ce rôle automatiquement lors de la configuration de votre instance de service à l'aide de la palette de commandes de VS Code.

> **Note**
>
> Le rôle AWSServiceRoleForAppRunner lié à un service permet d' AWS App Runner effectuer les tâches suivantes :
>
> - Transférez les journaux vers les groupes de CloudWatch journaux Amazon Logs.
> - Créez des règles Amazon CloudWatch Events pour vous abonner à la transmission d'images Amazon Elastic Container Registry (Amazon ECR).
>
> Vous n'avez pas besoin de créer manuellement un rôle lié au service . Lorsque vous créez un rôle AWS App Runner dans Console de gestion AWS ou à l'aide d'opérations d'API appelées par AWS Toolkit for Visual Studio Code, AWS App Runner crée ce rôle lié à un service pour vous.

Pour plus d'informations, consultez Gestion des identités et des accès pour App Runner dans le Guide du développeur AWS App Runner .

Obtention de sources de service pour App Runner

Vous pouvez utiliser AWS App Runner pour déployer des services à partir d'une image source ou d'un code source.

Source image

Si vous effectuez un déploiement à partir d'une image source, vous pouvez obtenir un lien vers le référentiel de cette image à partir d'un registre d' AWS images privé ou public.

- Registre privé Amazon ECR : copiez l'URI d'un référentiel privé qui utilise la console Amazon ECR à l'adresse. https://console.aws.amazon.com/ecr/repositories
- Registre public Amazon ECR : copiez l'URI d'un référentiel public qui utilise la galerie publique Amazon ECR à l'adresse. https://gallery.ecr.aws/

> **Note**
>
> Vous pouvez également obtenir l'URI d'un référentiel Amazon ECR privé directement depuis AWS Explorer dans Toolkit for VS Code :

- Ouvrez AWS Explorer et développez le nœud ECR pour afficher la liste des référentiels de cette AWS région.
- Right-click un référentiel et choisissez Copier l'URI du référentiel pour copier le lien dans votre presse-papiers.

Vous spécifiez l'URI du référentiel d'images lors de la configuration de votre instance de service à l'aide de la palette de commandes de VS Code

Pour plus d'informations, consultez Service App Runner basé sur une image source dans le Guide du développeur AWS App Runner .

Source code

Pour que votre code source soit déployé sur un AWS App Runner service, il doit être stocké dans un dépôt Git géré par un fournisseur de référentiel pris en charge. App Runner prend en charge un fournisseur de référentiel de code source : GitHub.

Pour plus d'informations sur la configuration d'un GitHub référentiel, consultez la documentation de démarrage sur GitHub.

Pour déployer votre code source sur un service App Runner à partir d'un GitHub référentiel, App Runner établit une connexion à GitHub. Si votre dépôt est privé (c'est-à-dire qu'il n'est pas accessible au public le GitHub), vous devez fournir les informations de connexion à App Runner.

> **Important**
>
> Pour créer GitHub des connexions, vous devez utiliser la console App Runner (https://console.aws.amazon.com/apprunner) pour créer une connexion GitHub vers AWS. Vous pouvez sélectionner les connexions disponibles sur la page des GitHub connexions lors de la configuration de votre instance de service à l'aide de la palette de commandes de VS Code.
> Pour plus d'informations, consultez Gestion des connexions App Runner dans le Guide du développeur AWS App Runner .

L'instance de service App Runner fournit un environnement d'exécution géré qui permet de créer et d'exécuter votre code. AWS App Runner prend actuellement en charge les environnements d'exécution suivants :

- Exécution gérée par Python
- Node.js exécution gérée

Dans le cadre de la configuration de votre service, vous fournissez des informations sur la façon dont le service App Runner crée et démarre votre service. Vous pouvez saisir ces informations à l'aide de la palette de commandes ou spécifier un fichier de configuration YAML-formatted App Runner. Les valeurs de ce fichier indiquent à App Runner comment créer et démarrer votre service et fournissent le contexte d'exécution. Cela inclut les paramètres réseau et les variables d'environnement pertinents. Le fichier de configuration a pour nom `apprunner.yaml`. Il est automatiquement ajouté au répertoire racine du référentiel de votre application.

Tarification

Les ressources de calcul et de mémoire utilisées par votre application vous sont facturées. De plus, si vous automatisez vos déploiements, vous payez également des frais mensuels fixes pour chaque application, qui couvrent tous les déploiements automatisés pour ce mois. Si vous optez pour un déploiement à partir du code source, vous payez en outre des frais de création correspondant au temps nécessaire à App Runner pour créer un conteneur à partir de votre code source.

Pour plus d'informations, consultez AWS App Runner Tarification.

Rubriques

- Création de services App Runner
- Gestion des services App Runner

Création de services App Runner

Vous pouvez créer un service App Runner dans Toolkit for VS Code à l'aide de l'AWS Explorateur et de la palette de commandes de VS Code. Une fois que vous avez choisi de créer un service dans une AWS région spécifique, les étapes numérotées fournies par la palette de commandes vous guident tout au long du processus de configuration de l'instance de service sur laquelle votre application s'exécute.

Avant de créer un service App Runner, assurez-vous que vous avez rempli les prérequis. Cela inclut la fourniture des autorisations IAM pertinentes et la confirmation du référentiel source spécifique que vous souhaitez déployer.

Pour créer un service App Runner

1. Ouvrez AWS Explorer, s'il n'est pas déjà ouvert.
2. Right-click le nœud App Runner et choisissez Create Service.

   La palette de commandes s'affiche.
3. Pour Select a source code location type (Sélectionner un type d'emplacement de code source), choisissez ECR ou Repository (Référentiel).

   Si vous choisissez ECR, vous spécifiez une image de conteneur dans un référentiel maintenu par Amazon Elastic Container Registry. Si vous choisissez Repository (Référentiel), vous spécifiez un référentiel de code source maintenu par un fournisseur de référentiel pris en charge. Actuellement, App Runner est compatible en GitHub tant que fournisseur de référentiel de code source.

Déploiement à partir d'ECR

1. Pour Select or enter an image repository (Sélectionner ou entrer un référentiel d'images), choisissez ou saisissez l'URL du référentiel d'images qui est maintenu par votre registre privé Amazon ECR ou la galerie publique Amazon ECR.

   > **Note**
   >
   > Si vous spécifiez un référentiel de la galerie publique Amazon ECR, assurez-vous que les déploiements automatiques sont désactivés car App Runner ne prend pas en charge les déploiements automatiques pour une image dans un référentiel public ECR.
   > Les déploiements automatiques sont désactivés par défaut, et cela est indiqué lorsque l'icône de l'en-tête de la Palette de commandes est traversée par une diagonale. Si vous choisissez d'activer les déploiements automatiques, un message vous informe que cette option peut entraîner des coûts supplémentaires.
2. Si l'étape Palette de commandes indique qu'aucune balise n'a été trouvée, vous devez revenir en arrière pour sélectionner un référentiel contenant une image de conteneur balisée.
3. Si vous utilisez un registre privé Amazon ECR, vous avez besoin du rôle d'accès ECR AppRunnerECRAccessRole, qui permet à App Runner d'accéder aux images Amazon Elastic Container Registry (Amazon ECR) de votre compte. Choisissez l'icône « + » dans l'en-tête de la Palette de commandes pour créer automatiquement ce rôle. (Un rôle d'accès n'est pas

nécessaire si votre image est stockée dans Amazon ECR public, où les images sont accessibles au public.)

4. Pour Port, saisissez le port IP utilisé par le service (Port `8000`, par exemple).
5. Pour Configure environment variables (Configurer les variables d'environnement), vous pouvez spécifier un fichier qui contient des variables d'environnement utilisées pour personnaliser le comportement de votre instance de service. Vous pouvez également ignorer cette étape.
6. Pour Name your service (Nommer votre service), saisissez un nom unique sans espaces et appuyez sur Entrée.
7. Pour Select instance configuration (Sélectionner la configuration de l'instance), choisissez une combinaison d'unités CPU et de mémoire en Go pour votre instance de service.

   Lorsque votre service est en cours de création, son statut passe de Creating (Création) à Running (Exécution).
8. Une fois que votre service commence à s'exécuter, cliquez dessus du bouton droit de la souris et choisissez Copy Service URL (Copier l'URL du service).
9. Pour accéder à votre application déployée, collez l'URL copiée dans la barre d'adresse de votre navigateur Web.

Déploiement à partir d'un référentiel distant

1. Pour Sélectionnez une connexion, choisissez une connexion qui est liée GitHub à AWS. Les connexions disponibles pour la sélection sont répertoriées sur la page GitHub des connexions de la console App Runner.
2. Pour Sélectionner un GitHub dépôt distant, choisissez ou entrez une URL pour le dépôt distant.

   Les référentiels distants déjà configurés avec la gestion du contrôle de source (SCM) de Visual Studio Code peuvent être sélectionnés. Vous pouvez également coller un lien vers le référentiel s'il n'est pas répertorié.
3. Pour Select a branch (Sélectionner une branche), choisissez la branche Git de votre code source que vous souhaitez déployer.
4. Pour Choose configuration source (Choisir la source de la configuration), indiquez comment vous souhaitez définir votre configuration d'exécution.

   Si vous choisissez Use configuration file (Utiliser le fichier de configuration), votre instance de service est configurée par les paramètres qui sont définis par le fichier de configuration

`apprunner.yaml`. Ce fichier se trouve dans le répertoire racine du référentiel de votre application.

Si vous choisissez Configurer tous les paramètres ici, utilisez la palette de commandes pour spécifier les éléments suivants :

- Runtime (Exécution) : choisissez Python 3 ou Nodejs 12.
- Build command (Commande de création) : saisissez la commande permettant de créer votre application dans l'environnement d'exécution de votre instance de service.
- Start command (Commande de démarrage) : saisissez la commande permettant de démarrer votre application dans l'environnement d'exécution de votre instance de service.

5. Pour Port, saisissez le port IP utilisé par le service (Port `8000`, par exemple).
6. Pour Configure environment variables (Configurer les variables d'environnement), vous pouvez spécifier un fichier qui contient des variables d'environnement utilisées pour personnaliser le comportement de votre instance de service. Vous pouvez également ignorer cette étape.
7. Pour Name your service (Nommer votre service), saisissez un nom unique sans espaces et appuyez sur Entrée.
8. Pour Select instance configuration (Sélectionner la configuration de l'instance), choisissez une combinaison d'unités CPU et de mémoire en Go pour votre instance de service.

   Lorsque votre service est en cours de création, son statut passe de Creating (Création) à Running (Exécution).
9. Une fois que votre service commence à s'exécuter, cliquez dessus du bouton droit de la souris et choisissez Copy Service URL (Copier l'URL du service).
10. Pour accéder à votre application déployée, collez l'URL copiée dans la barre d'adresse de votre navigateur Web.

> **Note**
>
> Si votre tentative de création d'un service App Runner échoue, le service affiche le statut Create failed (Création échouée) dans AWS Explorer. Pour des conseils de dépannage, consultez Lorsque la création du service échoue dans le Guide du développeur App Runner.

Gestion des services App Runner

Après avoir créé un service App Runner, vous pouvez le gérer à l'aide du volet AWS Explorer pour effectuer les activités suivantes :

- Mettre en pause et reprendre les services App Runner
- Déploiement des services App Runner
- Affichage des flux de journaux pour App Runner
- Suppression de services App Runner

Mettre en pause et reprendre les services App Runner

Si vous devez désactiver temporairement votre application Web et arrêter l'exécution du code, vous pouvez suspendre votre service AWS App Runner. App Runner réduit la capacité de calcul du service à zéro. Lorsque vous êtes prêt à exécuter à nouveau votre application, réactivez votre service App Runner. App Runner fournit une nouvelle capacité de calcul, y déploie votre application et l'exécute.

> **Important**
>
> Vous êtes facturé pour App Runner uniquement lorsqu'il est en cours d'exécution. Par conséquent, vous pouvez interrompre et reprendre votre application selon vos besoins pour gérer les coûts. Ceci est particulièrement utile dans les scénarios de développement et de test.

Pour mettre en pause votre service App Runner

1. Ouvrez AWS Explorer, s'il n'est pas déjà ouvert.
2. Développez App Runner pour afficher la liste des services.
3. Right-click votre service et choisissez Pause.
4. Dans la boîte de dialogue qui s'affiche, cliquez sur Confirm (Confirmer).

   Pendant que le service est en pause, son statut passe de Running (En cours d'exécution) à Pausing (Mise en pause), puis à Paused (En pause).

Pour reprendre votre service App Runner

1. Ouvrez AWS Explorer, s'il n'est pas déjà ouvert.
2. Développez App Runner pour afficher la liste des services.
3. Right-click votre service et choisissez Resume.

   Pendant la reprise du service, le statut du service passe de Resuming (Reprise) à Running (Exécution).

Déploiement des services App Runner

Si vous choisissez l'option de déploiement manuel pour votre service, vous devez initier explicitement chaque déploiement vers votre service.

1. Ouvrez AWS Explorer, s'il n'est pas déjà ouvert.
2. Développez App Runner pour afficher la liste des services.
3. Right-click votre service et choisissez Démarrer le déploiement.
4. Pendant le déploiement de votre application, le statut du service passe de Deploying (Déploiement) à Running (Exécution).
5. Pour confirmer que votre application a été déployée avec succès, cliquez avec le bouton droit de la souris sur le même service et choisissez Copy Service URL (Copier l'URL du service).
6. Pour accéder à votre application web déployée, collez l'URL copiée dans la barre d'adresse de votre navigateur web.

Affichage des flux de journaux pour App Runner

Utilisez CloudWatch les journaux pour surveiller, stocker et accéder à vos flux de journaux pour des services tels que App Runner. Un flux de journal est une séquence d'événements du journaux qui partagent la même source.

1. Développez App Runner pour afficher la liste des instances de service.
2. Développez une instance de service spécifique pour afficher la liste des groupes de journaux. (Un groupe de journaux est un groupe de flux de journaux qui partagent les mêmes paramètres de conservation, de surveillance et de contrôle d'accès.)
3. Right-click un groupe de journaux et choisissez Afficher les flux de journaux.
4. Dans la palette de commandes, choisissez un flux de journaux dans le groupe.

L'éditeur VS Code affiche la liste des événements de journal qui constituent le flux. Vous pouvez choisir de charger des événements plus anciens ou plus récents dans l'éditeur.

Suppression de services App Runner

> **Important**
>
> Si vous supprimez votre service App Runner, il est définitivement supprimé et vos données stockées sont effacées. Si vous devez recréer le service, App Runner doit à nouveau récupérer votre source et la compiler s'il s'agit d'un référentiel de code. Votre application web obtient un nouveau domaine App Runner.

1. Ouvrez AWS Explorer, s'il n'est pas déjà ouvert.
2. Développez App Runner pour afficher la liste des services.
3. Right-click un service et choisissez Supprimer le service.
4. Dans la palette de commandes, saisissez Supprimer, puis appuyez sur Entrée pour confirmer.

   Le service supprimé affiche le statut Deleting Suppression, puis le service disparaît de la liste.

AWS Générateur d'applications

AWS Application Builder for the AWS Toolkit for Visual Studio Code est votre guide pour créer des projets visuellement, les itérer localement et déployer vos applications sur. AWS

Les rubriques suivantes décrivent comment utiliser AWS Application Builder à partir du AWS Toolkit for Visual Studio Code.

Rubriques

- Travailler avec AWS Générateur d'applications

Travailler avec AWS Générateur d'applications

Les sections suivantes décrivent comment accéder à AWS Application Builder dans le AWS Toolkit for Visual Studio Code. Avec Application Builder, vous pouvez créer des projets visuellement, les

itérer localement et les déployer AWS sur. Pour un aperçu des fonctionnalités et des cas d'utilisation potentiels d'Application Builder et de votre AWS Lambda expérience locale, regardez la YouTube vidéo pour AWS développeurs *New* AWS Lambda Local IDE Experience ! .

En collaboration avec AWS Explorateur Application Builder

Pour accéder à Application Builder dans le AWS Toolkit, ouvrez le AWS Toolkit dans VS Code, puis développez l'explorateur AWS Application Builder. L'explorateur AWS Application Builder contient un lien permettant d'ouvrir la procédure pas à pas d'Application Builder dans un onglet de l'éditeur VS Code et affiche les dossiers de votre espace de travail VS Code actuel contenant des ressources relatives à AWS Application Builder.

Depuis l'explorateur Application Builder de la AWS boîte à outils, quatre actions au niveau du dossier de projet sont accessibles à partir des icônes de bouton situées à côté de votre dossier de projet ou en ouvrant le menu contextuel du dossier de projet (en cliquant avec le bouton droit de la souris) sur le dossier de projet :

- Ouvrir un fichier modèle : ouvre votre fichier modèle dans l'explorateur VS Code.
- Ouvrir avec Infrastructure Composer : ouvre votre fichier modèle avec AWS Infrastructure Composer dans l'éditeur VS Code. Pour plus d'informations sur l'utilisation d' AWS Infrastructure Composer, consultez la rubrique Qu'est-ce qu' AWS Infrastructure Composer dans le Guide du développeur AWS d'Infrastructure Composer.
- Créer un modèle SAM : ouvre la boîte de dialogue Spécifier les paramètres pour la génération dans la AWS boîte à outils. Vous pouvez choisir de spécifier des indicateurs de compilation pour la génération ou d'utiliser les valeurs par défaut de samconfig. Pour des informations détaillées sur les AWS SAM modèles, consultez la rubrique Anatomie des modèles dans le Guide du AWS Serverless Application Model développeur.
- Déployer l'application SAM : ouvre la boîte de dialogue de commande de sélection du déploiement dans VS Code, dans laquelle vous pouvez choisir de déployer votre application ou de la synchroniser pour mettre à jour une application que vous avez déjà déployée. Pour obtenir des informations détaillées sur le déploiement AWS SAM d'applications, consultez la rubrique Déployez votre application et vos ressources dans le Guide du AWS Serverless Application Model développeur.

Deux actions sont accessibles à partir des icônes situées à côté de la AWS Lambda fonction dans le dossier de votre projet ou en cliquant avec le bouton droit sur la AWS Lambda fonction :

- Configuration d'appel et de débogage locaux : ouvre le formulaire de configuration d'appel et de débogage locaux dans votre éditeur VS Code. Avec ce formulaire, vous pouvez créer, modifier et exécuter des configurations de lancement de type :. `aws-sam` Pour plus d'informations sur les configurations de débogage SAM, consultez la rubrique Options de configuration pour le débogage des applications sans serveur dans ce guide de l'utilisateur.

  > **Note**
  >
  > À l'heure actuelle, le débogage d'une application .NET Core sur une architecture ARM64 n'est pas pris en charge par VS Code. Si vous tentez de déboguer une application .NET Core, l'erreur suivante s'affiche :
  > `The vsdbg debugger does not currently support the arm64 architecture. Function will run locally without debug.`
  > Pour plus de détails sur ce problème, consultez ce VSCode-csharp problème dans le DotNet GitHub référentiel.

- Ouvrir le gestionnaire de fonctions : ouvre votre fichier de projet contenant le gestionnaire de fonctions.

Deux actions supplémentaires sont disponibles pour les AWS Lambda fonctions déployées.

- Invocation à distance : ouvre le menu de configuration de l'appel à distance dans l'éditeur VS Code.
- Journaux de recherche : ouvre la boîte de dialogue des journaux de recherche dans VS Code.

Présentation d'Application Builder

La procédure pas à pas d'Application Builder est un guide interactif étape par étape qui vous explique le processus de création d'une nouvelle application avec AWS Application Builder. Vous pouvez accéder à la présentation d'Application Builder à partir de deux endroits : l'explorateur Application Builder dans l'onglet VS Code Welcome AWS Toolkit for Visual Studio Code et l'onglet VS Code Welcome. Lorsque vous sélectionnez Procédure pas à pas d'Application Builder dans l'explorateur Application Builder du AWS Toolkit, la procédure pas à pas d'Application Builder s'ouvre dans l'onglet Bienvenue de VS Code de la fenêtre VS Code Editor.

La présentation d'Application Builder comprend 5 sections principales :

1. Installation

La section Installation vérifie si vous avez installé les AWS CLI outils requis par Application Builder et d'autres outils facultatifs. Si vous ne disposez pas des outils nécessaires ou si vos outils ne sont plus à jour, vous êtes guidé tout au long du processus d'installation des versions appropriées.

Pour vérifier si les outils appropriés AWS CLI et facultatifs sont installés, sélectionnez le bouton correspondant à l'outil AWS CLI ou à un autre outil que vous souhaitez tester. Après avoir sélectionné un bouton, les journaux de votre AWS boîte à outils sont mis à jour et VS Code affiche un message d'alerte indiquant l'état de vos outils. Si vous devez installer ou mettre à jour vos outils, la procédure pas à pas d'Application Builder est mise à jour avec les instructions et les ressources dont vous avez besoin pour continuer.

Pour obtenir des informations détaillées sur l'installation de AWS CLI, consultez la rubrique Installer ou mettre à jour la dernière version de la AWS CLI rubrique dans le Guide du AWS CLI développeur. Pour obtenir des informations détaillées sur l'installation de la AWS SAM CLI, consultez la rubrique Installer la AWS SAM CLI dans le Guide du développeur de la AWS SAM CLI.

2. Choisissez votre modèle de candidature

La section Choisissez votre modèle de candidature vous guide tout au long du processus de création d'une nouvelle application à partir d'un modèle.

Pour choisir un modèle et initialiser votre application, procédez comme suit.

1. Dans la procédure pas à pas d'Application Builder, sélectionnez la section Choisissez votre modèle d'application pour afficher une liste des options de modèle sur votre écran.
2. Choisissez un modèle dans la liste, puis cliquez sur le bouton Initialiser votre projet pour ouvrir une boîte de dialogue VS Code.
3. Suivez les étapes de la boîte de dialogue VS Code pour initialiser votre nouvelle application.
4. Les journaux de la AWS boîte à outils sont mis à jour en fonction de l'état de votre application pendant le processus d'initialisation.
5. Pour afficher votre application dans l'explorateur Application Builder, cliquez sur l'icône Actualiser l'explorateur Application Builder pour mettre à jour l'explorateur avec vos modifications.

3. Répéter localement

La section Itérer localement contient des exemples d'images qui montrent comment vous pouvez itérer à l'aide des fonctionnalités de l'Application Builder disponibles dans les explorateurs VS Code et AWS Toolkit.

Pour plus d'informations sur toutes les fonctionnalités d'Application Builder disponibles dans les explorateurs VS Code et AWS Toolkit, consultez la section Utilisation de l'explorateur Application Builder, située dans cette rubrique du Guide de l'utilisateur.

4. Déployez vers AWS

La AWS section Déployer vers contient des informations sur la façon de configurer vos informations d'identification AWS pour vous connecter aux fins du déploiement de votre application et des exemples de déploiement de votre application avec Application Builder.

Pour vous connecter à l' AWS aide de vos informations d'identification existantes à partir de la procédure pas à pas d'Application Builder, effectuez l'une des procédures suivantes.

Main-d'œuvre : connectez-vous à AWS avec authentification unique.

1. Dans la AWS section Déployer vers de la procédure pas à pas d'Application Builder, cliquez sur le bouton Configurer les informations d'identification pour ouvrir le menu AWS : CONNEXION dans l'explorateur AWS Toolkit.
2. Dans le menu AWS : CONNEXION, choisissez Workforce, puis cliquez sur le bouton Continuer pour continuer.
3. Entrez votre URL de démarrage dans le champ prévu à cet effet, choisissez votre AWS région dans le menu déroulant, puis cliquez sur le bouton Continuer pour continuer.
4. Dans la fenêtre contextuelle VS Code, confirmez que vous souhaitez ouvrir le site AWS d'authentification dans votre navigateur par défaut.
5. À partir de votre navigateur par défaut, effectuez les étapes d'authentification, vous êtes averti lorsque l'authentification est terminée et vous pouvez fermer la fenêtre de votre navigateur en toute sécurité.

Informations d'identification IAM : stockez les clés à utiliser avec AWS CLI outils.

1. Dans la AWS section Déployer vers de la procédure pas à pas d'Application Builder, cliquez sur le bouton Configurer les informations d'identification pour ouvrir le menu AWS : CONNEXION dans l'explorateur AWS Toolkit.

2. Dans le menu AWS : CONNEXION, choisissez IAM Credentials, puis cliquez sur le bouton Continuer pour continuer.
3. Entrez un nom de profil dans le champ prévu à cet effet, puis saisissez votre **`Access Key`** et**`Secret Key`**, puis cliquez sur le bouton Continuer pour continuer.
4. VS Code affiche l'état de votre authentification et vous avertit si l'authentification est terminée ou si vos informations d'identification ne sont pas valides.

Pour plus d'informations sur la configuration de vos informations d'identification pour le déploiement avec le AWS CLI, consultez la AWS CLI rubrique Configurer le dans le Guide du AWS CLI développeur. Pour plus d'informations sur la connexion à AWS depuis le AWS Toolkit à l'aide de vos informations d'identification existantes, consultez la AWS rubrique Connexion à de ce guide de l'utilisateur.

AWS Compositeur d'infrastructure

Vous pouvez utiliser le AWS Toolkit for Visual Studio Code pour travailler avec le compositeur AWS d'infrastructure. AWS Infrastructure Composer est un constructeur visuel pour les AWS applications qui vous aide à concevoir l'architecture de votre application et à visualiser votre CloudFormation infrastructure.

Pour des informations détaillées sur AWS Infrastructure Composer, consultez le guide de l'utilisateur d'AWS Infrastructure Composer.

Les rubriques suivantes décrivent comment utiliser AWS Infrastructure Composer à partir du AWS Toolkit for Visual Studio Code.

Rubriques

- Utilisation du compositeur d' AWS infrastructure dans le kit d'outils

Utilisation du compositeur d' AWS infrastructure dans le kit d'outils

AWS Infrastructure Composer for the vous AWS Toolkit for Visual Studio Code permet de concevoir visuellement des applications via un canevas interactif. Vous pouvez également utiliser Infrastructure Composer pour visualiser CloudFormation et modifier des modèles AWS Serverless Application Model (AWS SAM). Lorsque vous travaillez avec Infrastructure Composer, vos modifications sont stockées de manière permanente, ce qui vous permet de basculer facilement entre les fichiers d'édition directement dans l'éditeur VS Code ou à l'aide du canevas interactif.

Pour obtenir des informations détaillées sur AWS Infrastructure Composer, des informations de démarrage et des didacticiels, consultez le guide de l'utilisateur d'AWS Infrastructure Composer.

Les sections suivantes décrivent comment accéder à AWS Infrastructure Composer depuis le AWS Toolkit for Visual Studio Code.

Accès à AWS Infrastructure Composer à partir de la boîte à outils

Il existe 3 manières principales d'accéder à AWS Infrastructure Composer à partir du kit d'outils.

Accès à AWS Infrastructure Composer à partir d'un modèle existant

1. Dans VS Code, ouvrez un fichier modèle existant dans l'éditeur VS Code.
2. Dans la fenêtre de l'éditeur, cliquez sur le bouton AWS Infrastructure Composer situé dans le coin supérieur droit de la fenêtre de l'éditeur.
3. AWS Infrastructure Composer ouvre et visualise votre fichier de modèle dans la fenêtre de l'éditeur de code VS.

Accès à AWS Infrastructure Composer depuis le menu contextuel (clic droit)

1. Dans VS Code, cliquez avec le bouton droit sur le fichier modèle que vous souhaitez ouvrir avec AWS Infrastructure Composer.
2. Dans le menu contextuel, choisissez l'option Ouvrir avec App Composer.
3. AWS Infrastructure Composer ouvre et visualise votre fichier de modèle dans une nouvelle fenêtre de l'éditeur VS Code.

Accès à AWS Infrastructure Composer depuis la palette de commandes

1. Depuis VS Code, ouvrez la palette de commandes en appuyant sur **`Cmd + Shift + P`** ou **`Ctrl + Shift + P`** (Windows)
2. Dans le champ de recherche, entrez **`AWS Infrastructure Composer`** et choisissez AWS Infrastructure Composer lorsqu'il renseigne les résultats.
3. Choisissez le fichier modèle que vous souhaitez ouvrir, AWS Infrastructure Composer ouvre et visualise votre fichier modèle dans une nouvelle fenêtre de l'éditeur VS Code.

AWS CDK pour VS Code

Il s'agit de la documentation d'avant-première d'une fonctionnalité en version préliminaire. Elle est susceptible d'être modifiée.

AWS CDK Explorer vous permet d'utiliser des applications Cloud Development Kit AWS (AWS CDK) ou applis. Vous trouverez des informations détaillées à ce sujet AWS CDK dans le guide du Cloud Development Kit AWS (AWS CDK) développeur.

AWS CDK les applications sont composées d'éléments de base appelés constructions, qui incluent les définitions de vos CloudFormation piles et des AWS ressources qu'elles contiennent. À l'aide de l'AWS CDK explorateur, vous pouvez visualiser les piles et les ressources définies dans les AWS CDK constructions. Cette visualisation est fournie sous forme d'arborescence dans le volet Outils de développement de l'éditeur Visual Studio Code (VS Code).

Cette section fournit des informations sur l'accès et l'utilisation AWS CDK de l'éditeur VS Code. Cela suppose que vous avez déjà installé et configuré le Toolkit for VS Code pour votre IDE local.

Rubriques

- Travailler avec AWS CDK applications

Travailler avec AWS CDK applications

Il s'agit de la documentation d'avant-première d'une fonctionnalité en version préliminaire. Elle est susceptible d'être modifiée.

Utilisez l'AWS CDK explorateur du AWS Toolkit for VS Code for VS Code pour visualiser et utiliser des AWS CDK applications.

Conditions préalables

- Assurez-vous que votre système répond aux conditions requises spécifiées dans Installation du Toolkit for VS Code.

- Installez l'interface de ligne de AWS CDK commande, comme décrit dans les premières sections de Getting Started with the AWS CDK du Guide du Cloud Development Kit AWS (AWS CDK) développeur.

> **Important**
>
> La AWS CDK version doit être 1.17.0 ou ultérieure. Utilisez **`cdk --version`** sur la ligne de commande pour voir quelle version vous exécutez.

Visualisez un AWS CDK vigie

À l'aide du AWS Toolkit for VS Code AWS CDK Explorer, vous pouvez gérer les piles et les ressources stockées dans les constructions CDK de vos applications. L' AWS CDK explorateur affiche vos ressources sous forme d'arborescence à l'aide des informations définies dans le `tree.json` fichier créé lorsque vous exécutez la **`cdk synth`**commande. Le `tree.json` fichier se trouve par défaut dans le `cdk.out` répertoire de l'application.

Pour commencer à utiliser le Toolkit AWS CDK Explorer, vous devez créer une application CDK.

1. Effectuez les premières étapes du didacticiel Hello World qui se trouve dans le guide du AWS CDK développeur.

   > **Important**
   >
   > Lorsque vous arrivez à l'étape du didacticiel Déploiement de la pile, arrêtez-vous et revenez à ce guide.

   > **Note**
   >
   > Vous pouvez exécuter les commandes fournies dans le didacticiel, par exemple **`cdk init`**, **`mkdir`**et sur la ligne de commande d'un système d'exploitation ou dans une fenêtre de terminal dans l'éditeur VS Code.

2. Après avoir terminé les étapes requises du didacticiel CDK, ouvrez le contenu du CDK que vous avez créé dans l'éditeur VS Code.

3. Dans le volet de AWS navigation, développez le titre CDK (Preview). Vos applications CDK et leurs ressources associées sont désormais affichées dans l'arborescence de CDK Explorer.

Remarques importantes

- Lorsque vous chargez des applications CDK dans l'éditeur VS Code, vous pouvez charger plusieurs dossiers à la fois. Chaque dossier peut contenir plusieurs applications CDK, comme indiqué dans l'image précédente. L' AWS CDK explorateur trouve les applications dans le répertoire racine du projet et dans ses sous-répertoires directs.

- Lorsque vous effectuez les premières étapes du didacticiel, vous pouvez remarquer que la dernière commande que vous exécutez est **`cdk synth`**, qui génère le fichier `tree.json`. Si vous modifiez certains aspects d'une application CDK, par exemple si vous ajoutez des ressources supplémentaires, vous devez exécuter à nouveau cette commande pour que les modifications soient reflétées dans l'arborescence.

Effectuez d'autres opérations sur un AWS CDK app

Vous pouvez utiliser l'éditeur VS Code pour effectuer d'autres opérations sur une application CDK, comme vous le feriez en utilisant la ligne de commande de votre système d'exploitation ou d'autres outils. Par exemple, vous pouvez mettre à jour les fichiers de code dans l'éditeur et déployer l'application à l'aide d'une fenêtre VS Code Terminal.

Pour essayer ce type d'actions, utilisez l'éditeur VS Code pour poursuivre le didacticiel Hello World dans le guide du AWS CDK développeur. Assurez-vous d'effectuer la dernière étape, la destruction des ressources de l'application, afin de ne pas encourir de frais imprévus sur votre AWS compte.

Travailler avec des AWS CloudFormation piles

AWS Toolkit for Visual Studio Code Il fournit un support pour les AWS CloudFormationpiles. À l'aide du Toolkit for VS Code, vous pouvez effectuer certaines tâches avec des AWS CloudFormation piles, telles que leur suppression.

Rubriques

- Supprimer une CloudFormation pile
- Créez un AWS CloudFormation modèle utilisant le AWS Toolkit for Visual Studio Code

Supprimer une CloudFormation pile

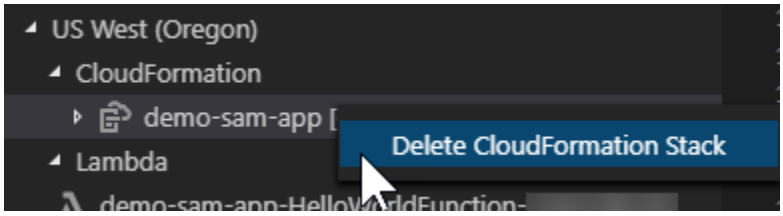
Vous pouvez utiliser le AWS Toolkit for Visual Studio Code pour supprimer des CloudFormation piles.

Prérequis

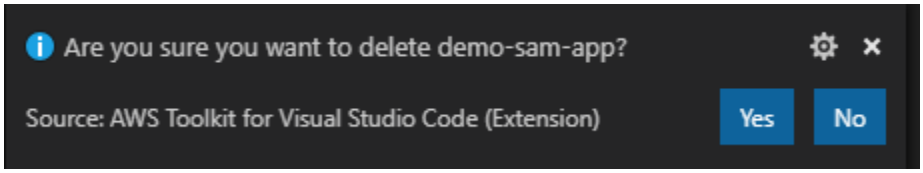
- Assurez-vous que votre système répond aux conditions requises spécifiées dans Installation du Toolkit for VS Code.
- Assurez-vous que les informations d'identification que vous avez configurées Authentification et accès incluent un accès en lecture/écriture approprié au CloudFormation service. Si dans l'AWS explorateur, sous CloudFormation, vous voyez un message similaire à « Erreur lors du chargement CloudFormation des ressources », vérifiez les autorisations associées à ces informations d'identification. Les modifications que vous apportez aux autorisations mettront quelques minutes à affecter l'AWS explorateur dans VS Code.

Supprimer une CloudFormation pile

1. Dans AWS Explorer, ouvrez le menu contextuel de la pile CloudFormation à supprimer.



2. Choisissez Delete CloudFormation Stack.
3. Dans le message qui s'affiche, choisissez Oui pour confirmer la suppression.



Une fois la pile supprimée, elle disparaît d'AWS Explorer.

Créez un AWS CloudFormation modèle utilisant le AWS Toolkit for Visual Studio Code

Ils AWS Toolkit for Visual Studio Code peuvent vous aider à rédiger AWS CloudFormation et à créer des modèles SAM.

Conditions préalables

Toolkit pour VS Code et prérequis d'identification

- Avant de pouvoir accéder au CloudFormation service depuis le Toolkit for VS Code, vous devez satisfaire aux exigences décrites dans le guide d'utilisation relatif à l'installation du Toolkit for VS Code.
- Les informations d'identification que vous avez créées Authentification et accès doivent inclure read/write un accès approprié au AWS CloudFormation service.

> **Note**
>
> Si le CloudFormationservice affiche un message d'erreur lors du chargement CloudFormation des ressources, vérifiez les autorisations que vous avez associées à ces informations d'identification. Notez également que les modifications apportées aux autorisations peuvent prendre quelques minutes pour être mises à jour dans l'AWS explorateur.

CloudFormation prérequis du modèle

- Installez et activez l'extension Redhat Developer YAML VS Code.
- Vous devez être connecté à Internet lorsque vous utilisez l'extension Redhat Developer YAML VS Code, car elle est utilisée pour télécharger et mettre en cache des schémas JSON sur votre machine.

Rédaction d'un CloudFormation modèle avec le support du schéma YAML

La boîte à outils utilise le support du langage YAML et des schémas JSON pour rationaliser le processus d'écriture CloudFormation et les modèles SAM. Des fonctionnalités telles que la validation syntaxique et l'autocomplétion accélèrent non seulement le processus, mais contribuent également

à améliorer la qualité de votre modèle. Lorsque vous sélectionnez un schéma pour votre modèle, les meilleures pratiques recommandées sont les suivantes.

CloudFormation modèle

- Le fichier possède une extension .yaml ou .yml.
- Le fichier possède un nœud de niveau supérieur `AWSTemplateFormatVersion` ou un nœud de ressources.

Modèle SAM

- Tous les critères déjà décrits pour CloudFormation
- Le fichier possède un nœud Transform de haut niveau, contenant une valeur commençant `AWS::Serverless` par.

Le schéma sera appliqué lors de la modification du fichier. Par exemple, un schéma de modèle SAM sera appliqué après avoir ajouté une transformation sans serveur à un CloudFormation modèle et enregistré le fichier.

Validation de syntaxe

L'extension YAML appliquera automatiquement la validation de type à votre modèle. Cela met en évidence les entrées dont le type n'est pas valide pour une propriété donnée. Si vous survolez une entrée surlignée, les extensions affichent des actions correctives.

Autocomplétion

Lorsque vous ajoutez de nouveaux champs, des valeurs énumérées ou d'autres types de ressources, vous pouvez lancer la fonction d'autocomplétion de l'extension YAML en tapant Ctrl + espace.

Utilisation des CloudWatch journaux à l'aide du AWS Toolkit for Visual Studio Code

Amazon CloudWatch Logs vous permet de centraliser les journaux de tous vos systèmes, applications et AWS services que vous utilisez, au sein d'un seul service hautement évolutif. Vous pouvez ensuite les afficher facilement, y effectuer des recherches de codes d'erreur ou des modèles, les filtrer en fonction de champs spécifiques ou les archiver en toute sécurité pour procéder à une

analyse ultérieure. Pour plus d'informations, consultez Qu'est-ce qu'Amazon CloudWatch Logs ? dans le guide de CloudWatch l'utilisateur Amazon.

Les rubriques suivantes décrivent comment utiliser le AWS Toolkit for Visual Studio Code pour utiliser les CloudWatch journaux dans un AWS compte.

Rubriques

- Afficher les groupes de CloudWatch journaux et les flux de journaux à l'aide du AWS Toolkit for Visual Studio Code
- Utilisation des événements de CloudWatch journalisation dans les flux de journaux à l'aide du AWS Toolkit for Visual Studio Code
- Recherche de groupes de CloudWatch journaux
- Amazon CloudWatch Logs Live Tail

Afficher les groupes de CloudWatch journaux et les flux de journaux à l'aide du AWS Toolkit for Visual Studio Code

Un flux de journaux est une séquence d'événements de journaux qui partagent la même source. Chaque source distincte de CloudWatch journaux dans Logs constitue un flux de journaux distinct.

Un groupe de journaux est un groupe de flux de journaux qui partagent les mêmes paramètres de conservation, de surveillance et de contrôle d'accès. Vous pouvez définir des groupes de journaux et spécifier les flux à placer dans chaque groupe. Le nombre de flux de journaux pouvant appartenir à un groupe de journaux est illimité.

Pour plus d'informations, consultez la section Travailler avec des groupes de journaux et des flux de journaux dans le guide de CloudWatch l'utilisateur Amazon.

Rubriques

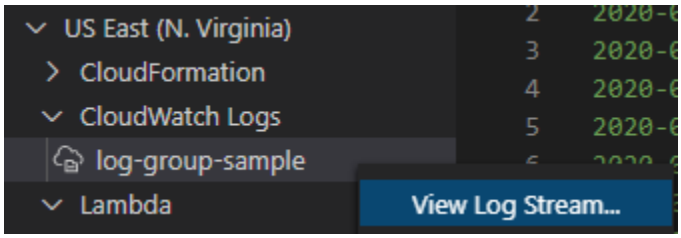
- Affichage des groupes de journaux et des flux de journaux avec le nœud CloudWatch Logs

Affichage des groupes de journaux et des flux de journaux avec le nœud CloudWatch Logs

1. Dans VS Code, choisissez Afficher, Explorateur pour ouvrir l' AWS Explorateur.
2. Cliquez sur le nœud CloudWatch Logs pour développer la liste des groupes de journaux.

Les groupes de journaux de la AWS région actuelle sont affichés sous le nœud CloudWatch Logs.

3. Pour afficher les flux de journaux d'un groupe de journaux, cliquez avec le bouton droit sur le nom du groupe de journaux, puis choisissez Afficher le flux de journaux.



4. Dans la palette de commandes, sélectionnez un flux de journal dans le groupe à afficher.

> **Note**
>
> La palette de commandes affiche l'horodatage du dernier événement de chaque flux.

L'éditeur Log Stream démarre pour afficher les événements du journal du flux.

Utilisation des événements de CloudWatch journalisation dans les flux de journaux à l'aide du AWS Toolkit for Visual Studio Code

Après avoir ouvert l'éditeur Log Steam, vous pouvez accéder aux événements du journal de chaque stream. Les événements de journaux sont des enregistrements de l'activité enregistrée par l'application ou la ressource contrôlée.

Rubriques

- Affichage et copie des informations des flux de journaux
- Enregistrement du contenu de l'éditeur de flux de journaux dans un fichier local

Affichage et copie des informations des flux de journaux

Lorsque vous ouvrez un flux de journal, l'éditeur de flux de journal affiche la séquence des événements de journal de ce flux.

1. Pour trouver un flux de journal à consulter, ouvrez l'éditeur de flux de journal (voirAffichage des groupes de CloudWatch journaux et des flux de journaux).

   Chaque ligne répertoriant un événement est horodaté pour indiquer quand il a été enregistré.

2. Vous pouvez afficher et copier des informations sur les événements du flux à l'aide des options suivantes :

   - Afficher les événements par heure : affichez les tout derniers et les anciens événements de journaux en choisissant Charger les événements les plus récentsou Charger les anciens événements.

     > **Note**
     >
     > L'éditeur Flux de journaux charge initialement un lot de 10 000 lignes d'événements de journaux les plus récents, soit 1 Mo de données de journaux (la plus petite des deux valeurs utilisée). Si vous choisissez Charger des événements les récents, l'éditeur affiche les événements qui ont été journalisés après le chargement du dernier lot. Si vous choisissez Charger les anciens événements, l'éditeur affiche un lot d'événements antérieur aux événements affichés.

   - Copier les événements de journaux : sélectionnez les événements à copier, puis cliquez avec le bouton droit et sélectionnez Copier dans le menu.
   - Copiez le nom du flux de journal : cliquez avec le bouton droit sur l'onglet de l'éditeur de flux de journal et choisissez Copier le nom du flux de journal.

     > **Note**
     >
     > Vous pouvez également utiliser la palette de commandes pour exécuter AWS Toolkit Copy Log Stream Name.

Enregistrement du contenu de l'éditeur de flux de journaux dans un fichier local

Vous pouvez télécharger le contenu de l'éditeur de flux de CloudWatch log dans un `log` fichier sur votre machine locale.

> **Note**
>
> Avec cette option, vous n'enregistrez dans un fichier que les événements du journal actuellement affichés dans l'éditeur de flux de journaux. Par exemple, si la taille totale d'un flux de journaux est de 5 Mo et que 2 Mo seulement sont chargés dans l'éditeur, votre fichier enregistré contiendra également 2 Mo de données de journaux. Pour afficher plus de données à enregistrer, choisissez Charger les événements les plus récents ou Charger les anciens événements dans l'éditeur.

1. Pour trouver un flux de journal à copier, ouvrez l'éditeur Log Streams (voirAffichage des groupes de CloudWatch journaux et des flux de journaux).
2. Cliquez sur l'icône Enregistrer à côté de l'onglet affichant le nom du flux de journal.

   > **Note**
   >
   > Vous pouvez également utiliser la palette de commandes pour exécuter AWS Toolkit Save Current Log Stream Content.

3. Utilisez la boîte de dialogue pour sélectionner ou créer un dossier de téléchargement pour le fichier journal, puis cliquez sur Enregistrer.

Recherche de groupes de CloudWatch journaux

Vous pouvez utiliser Search Log Group pour rechercher tous les flux de journaux d'un groupe de journaux.

Pour obtenir des informations détaillées sur le service Amazon CloudWatch Logs, consultez la rubrique Travailler avec les groupes de journaux et les flux de journaux dans le guide de CloudWatch l'utilisateur Amazon.

Recherche de groupes de journaux à partir de la palette de commandes VS Code

Pour rechercher des groupes de journaux à partir de la palette de commandes VS Code, procédez comme suit.

Pour obtenir des informations détaillées sur les filtres et modèles Amazon CloudWatch Logs, consultez la section Syntaxe des filtres et modèles du guide de CloudWatch l'utilisateur Amazon.

1. Dans VS Code, ouvrez la palette de commandes en appuyant sur **cmd+shift+p** (windows :**ctrl+shift+p**).
2. Dans la palette de commandes, entrez la commande**AWS: Search Log Group**, puis sélectionnez-la pour ouvrir la boîte de dialogue du groupe de journaux de recherche dans VS Code et suivez les instructions pour continuer.

   > **Note**
   >
   > Dès le premier message, vous avez la possibilité de changer de AWS région avant de passer aux étapes suivantes.

3. À partir de l'invite Select Log Group (1/3), sélectionnez le groupe de journaux que vous souhaitez rechercher.
4. À l'invite Select Time Filter (2/3), choisissez le filtre temporel à appliquer à votre recherche.
5. À partir de l'invite Search Log Group... (3/3), saisissez votre modèle de recherche dans le champ prévu à cet effet, puis appuyez sur la **Enter** touche pour continuer ou sur la **ESC** touche pour annuler la recherche.
6. Les résultats de votre recherche s'ouvrent dans l'éditeur VS Code, une fois la recherche terminée.

Recherche de groupes de journaux à partir de l' AWS explorateur

Pour rechercher des groupes de journaux à partir de l' AWS Toolkit for Visual Studio Code explorateur, procédez comme suit.

1. À partir de l' AWS Toolkit for Visual Studio Code explorateur, agrandissez CloudWatch.
2. Ouvrez le menu contextuel correspondant (avec le bouton droit de la souris) au groupe de journaux de recherche que vous souhaitez rechercher, puis choisissez Search Log Group pour ouvrir l'invite de recherche.
3. Suivez les instructions en sélectionnant une période pour continuer.
4. Lorsque vous y êtes invité, entrez votre modèle de recherche dans le champ prévu à cet effet, puis appuyez sur la **Enter** touche pour continuer ou sur la **ESC** touche pour annuler la recherche.
5. Les résultats de votre recherche s'ouvrent dans l'éditeur VS Code une fois la recherche terminée.

Utilisation des résultats du journal de recherche

Après avoir effectué une recherche de groupe de CloudWatch journaux réussie, les résultats de votre recherche s'ouvrent dans l'éditeur VS Code. Les procédures suivantes décrivent comment utiliser les résultats des journaux de recherche.

> **Note**
>
> Lorsque vous consultez un seul flux de journal, les fonctionnalités suivantes sont limitées aux résultats de votre flux de journal actuellement actif.

Sauvegarde des résultats de votre groupe de journaux de recherche

Pour enregistrer les résultats de votre groupe de journaux de recherche localement, procédez comme suit.

1. Dans les résultats de votre groupe de journaux de recherche, cliquez sur le bouton icône Enregistrer le journal dans un fichier, situé dans le coin supérieur droit de l'éditeur VS Code.
2. À l'invite Enregistrer sous, spécifiez le nom et l'emplacement dans lesquels vous souhaitez enregistrer le fichier.
3. Cliquez sur OK pour enregistrer le fichier sur votre ordinateur local.

Modification de l'intervalle de temps, de l'intervalle de temps

Pour modifier la plage de temps active dans les résultats de votre groupe de journaux de recherche, procédez comme suit.

1. Dans les résultats de votre groupe de journaux de recherche, choisissez la recherche par date... bouton d'icône, situé dans le coin supérieur droit de l'éditeur VS Code.
2. À l'invite de sélection du filtre temporel, choisissez une nouvelle plage de temps pour les résultats de votre journal de recherche.
3. Vos résultats sont mis à jour lorsque l'invite de sélection du filtre temporel est fermée.

Modification du modèle de recherche

Pour modifier le modèle de recherche actif dans les résultats de votre groupe de journaux de recherche, procédez comme suit.

1. Dans les résultats de votre groupe de journaux de recherche, choisissez la recherche par modèle... bouton d'icône, situé dans le coin supérieur droit de l'éditeur VS Code.
2. À partir de l'invite Search Log Group, entrez le nouveau modèle de recherche dans le champ prévu à cet effet.
3. Appuyez sur la **`Enter`** touche pour fermer l'invite et mettre à jour vos résultats avec le nouveau modèle de recherche.

Amazon CloudWatch Logs Live Tail

Diffusez en direct les événements de votre CloudWatch journal au fur et à mesure qu'ils sont ingérés dans un groupe de CloudWatch journaux particulier avec Amazon Logs Live Tail.

Pour obtenir des informations détaillées sur la fonctionnalité Live Tail, consultez la rubrique Résoudre les problèmes liés à CloudWatch Logs Live Tail dans le guide de l'utilisateur Amazon CloudWatch Logs.

Le coût des sessions Live Tail est calculé en fonction du temps d'utilisation de la session, par minute. Pour plus d'informations sur les tarifs, consultez l'onglet Logs dans la section Paid Tier du guide de CloudWatch tarification Amazon.

Démarrer une session Live Tail à partir de la palette de commandes VS Code

Pour démarrer une session Live Tail à partir de la palette de commandes VS Code, procédez comme suit.

Pour obtenir des informations détaillées sur les filtres et modèles Amazon CloudWatch Logs, consultez la section Syntaxe des filtres et modèles du guide de CloudWatch l'utilisateur Amazon.

Démarrer une session de suivi à partir de la palette de commandes

1. Dans VS Code, ouvrez la palette de commandes en appuyant sur **`cmd+shift+p`** (Windows :**`ctrl+shift+p`**).
2. Dans la palette de commandes, entrez la commande`AWS: Tail Log Group`, puis sélectionnez-la pour ouvrir la boîte de dialogue du groupe Tail log dans VS Code et suivez les instructions pour continuer.

> **Note**
>
> À la première invite, vous avez la possibilité de changer de AWS région avant de passer aux étapes suivantes.

3. À l'invite Tail Log Group (1/3), choisissez le groupe de journaux que vous souhaitez suivre.
4. À partir de l'invite Inclure les événements du journal depuis... (2/3), choisissez le filtre de flux de journal à appliquer à votre session de suivi.
5. À partir de l'invite Provide log event filter pattern... (3/3), saisissez votre modèle de filtre dans le champ prévu à cet effet, puis appuyez sur la **`Enter`** touche pour continuer ou sur la **`ESC`** touche pour annuler la recherche.
6. Une fois terminé, vos résultats sont diffusés dans l'éditeur VS Code

> **Note**
>
> Si une session Live Tail exécutée dans la fenêtre VS Code correspond à la configuration d'une commande Tail Log Group récemment soumise, aucune nouvelle session ne démarre. Au lieu de cela, votre session existante devient l'éditeur de texte actif.

Démarrer une session Live Tail depuis l' AWS explorateur

Pour démarrer une session Live Tail à partir de l'explorateur de boîtes à AWS outils, procédez comme suit.

Démarrer une session de suivi depuis l'explorateur AWS

1. À partir de l'explorateur de boîtes AWS à outils, développez CloudWatch.
2. Ouvrez le menu contextuel correspondant (avec le bouton droit de la souris) au groupe de journaux que vous souhaitez suivre, puis choisissez Tail Log Group pour ouvrir l'invite de suivi.
3. Suivez les instructions pour continuer.
4. Vos résultats seront diffusés dans l'éditeur VS Code.

Arrêter une session Live Tail

Il existe deux manières d'arrêter une session de Tailing en cours.

Arrêter une session de suivi

1. Cliquez sur le bouton Arrêter le suivi CodeLens au bas du document texte de la session de suivi.
2. Fermez tous les éditeurs contenant le document texte de la session de suivi.

Amazon DocumentDB

Vous pouvez gérer vos clusters et instances Amazon DocumentDB directement dans VS Code à l'aide du AWS Toolkit for Visual Studio Code. Amazon DocumentDB (compatible avec MongoDB) est un service de base de données rapide, fiable et entièrement géré qui simplifie la configuration, le fonctionnement et la mise à l'échelle des MongoDB-compatible bases de données dans le cloud. Pour obtenir des informations détaillées sur le service Amazon DocumentDB, consultez le manuel Amazon DocumentDB Developer Guide.

Les rubriques suivantes décrivent comment utiliser Amazon DocumentDB avec le AWS Toolkit for Visual Studio Code.

Rubriques

- Utilisation d'Amazon DocumentDB dans la boîte à outils

Utilisation d'Amazon DocumentDB dans la boîte à outils

Amazon DocumentDB (compatible avec MongoDB) est un service de base de données rapide, fiable et entièrement géré qui simplifie la configuration, le fonctionnement et la mise à l'échelle des MongoDB-compatible bases de données dans le cloud.

Pour obtenir des informations détaillées sur Amazon DocumentDB, des informations de démarrage et des didacticiels, consultez le guide du développeur Amazon DocumentDB.

Les sections suivantes décrivent comment utiliser Amazon DocumentDB avec. AWS Toolkit for Visual Studio Code

Accès à Amazon DocumentDB depuis le AWS Boîte à outils

Pour accéder à Amazon DocumentDB avec le AWS Toolkit, suivez la procédure suivante.

Accès à Amazon DocumentDB dans AWS Boîte à outils

1. Depuis VS Code, ouvrez AWS Toolkit for Visual Studio Code.

2. À partir de la AWS boîte à outils, développez l'explorateur.
3. Depuis l'Explorateur, développez Amazon DocumentDB pour afficher vos ressources Amazon DocumentDB existantes.

Création d'un cluster basé sur une instance.

Pour commencer à travailler avec Amazon DocumentDB, créez un cluster en suivant la procédure suivante.

Création d'un cluster basé sur une instance

1. Dans le AWS Toolkit for Visual Studio Code menu contextuel d'Amazon DocumentDB (cliquez avec le bouton droit de la souris), puis sélectionnez Créer un cluster pour ouvrir la boîte de dialogue Créer un cluster Amazon DocumentDB dans VS Code.
2. Dans l'écran Type de cluster, choisissez Instance Based Cluster.
3. Dans l'écran Nom du cluster, indiquez le nom de votre nouveau cluster.
4. Sur l'écran Sélectionner la version du moteur, choisissez votre version préférée du moteur Amazon DocumentDB.
5. Dans les écrans Nom d'utilisateur et mot de passe d'administrateur, spécifiez un nom d'utilisateur et un mot de passe d'administrateur pour protéger votre cluster.
6. Dans l'écran Spécifier le chiffrement du stockage, choisissez de chiffrer ou non votre cluster.
7. Dans l'écran Nombre d'instances, configurez le nombre d'instances que vous préférez.
8. Dans l'écran Sélectionner une classe d'instance, choisissez votre classe d'instance préférée, puis créez votre nouveau cluster.

> **Note**
>
> Le démarrage de votre cluster peut prendre plusieurs minutes.

Copier un point de terminaison de cluster

Pour copier le point de terminaison de votre cluster Amazon DocumentDB, effectuez la procédure suivante.

Copier un point de terminaison de cluster

1. À partir de AWS Toolkit for Visual Studio Code, développez Amazon DocumentDB pour afficher vos clusters Amazon DocumentDB.
2. Right-click le cluster dont vous souhaitez copier les détails de connexion, puis choisissez Copier le point de terminaison pour copier les informations de point de terminaison du cluster dans votre presse-papiers.
3. Le point de terminaison de votre cluster peut désormais être collé dans vos documents.

Ouvrir dans le navigateur

Ouvrez vos clusters Amazon DocumentDB dans la AWS console pour bénéficier de fonctionnalités supplémentaires de gestion des clusters. Pour ouvrir la AWS console à votre cluster Amazon DocumentDB dans votre navigateur Web par défaut, effectuez la procédure suivante.

Ouverture de votre cluster dans AWS Console

1. À partir de AWS Toolkit for Visual Studio Code, développez Amazon DocumentDB pour afficher vos clusters Amazon DocumentDB.
2. Right-click le cluster que vous souhaitez afficher dans la AWS console, puis choisissez Ouvrir dans le navigateur.
3. La AWS console s'ouvre sur le cluster Amazon DocumentDB dans votre navigateur Web par défaut.

Extension d'un cluster existant

Pour dimensionner vos clusters Amazon DocumentDB en ajoutant des instances, suivez la procédure suivante.

Ajouter une instance pour étendre votre cluster

1. À partir de AWS Toolkit for Visual Studio Code, développez Amazon DocumentDB pour afficher vos clusters Amazon DocumentDB.
2. Right-click le cluster que vous souhaitez développer et choisissez Ajouter une instance pour ouvrir la boîte de dialogue Ajouter une instance dans VS Code.
3. Lorsque vous y êtes invité, saisissez le nom de votre nouvelle instance dans le champ de texte, puis appuyez sur la **`Enter`** touche pour continuer.

4. Lorsque vous y êtes invité, sélectionnez une classe d'instance dans la liste pour continuer.
5. L'AWS explorateur affiche l'état de création et se met à jour lorsque la nouvelle instance est prête.

Arrêt d'un cluster

Arrêtez votre cluster Amazon DocumentDB en suivant la procédure suivante.

> **Note**
>
> Lorsque votre cluster est arrêté, la plupart des fonctionnalités de gestion de cluster ne seront pas disponibles.

Arrêter votre cluster Amazon DocumentDB

1. À partir de AWS Toolkit for Visual Studio Code, développez Amazon DocumentDB pour afficher vos clusters Amazon DocumentDB.
2. Cliquez sur le bouton Arrêter le cluster situé à côté du cluster que vous souhaitez arrêter ou cliquez avec le bouton droit sur le cluster et choisissez Arrêter le cluster.
3. Lorsque vous y êtes invité, choisissez Oui pour arrêter votre cluster ou Annuler pour annuler le processus d'arrêt et laisser votre cluster fonctionner.
4. L'AWS explorateur affiche l'état de votre cluster et se met à jour lorsque celui-ci est arrêté.

Redémarrer une instance

Le redémarrage d'une instance est utile pour résoudre les problèmes et apporter des modifications mineures sans affecter l'ensemble de votre cluster. Pour redémarrer une instance Amazon DocumentDB, effectuez la procédure suivante.

Redémarrage d'une instance de cluster

1. À partir de AWS Toolkit for Visual Studio Code, développez Amazon DocumentDB pour afficher vos clusters Amazon DocumentDB.
2. Right-click l'instance de cluster que vous souhaitez redémarrer, puis choisissez Redémarrer l'instance.

3. Lorsque vous y êtes invité, choisissez Oui pour redémarrer votre instance ou Annuler pour annuler le processus de redémarrage et laisser votre instance arrêtée.
4. L'AWS explorateur affiche l'état de votre cluster et se met à jour lorsque votre instance redémarre.

Suppression d'une instance

Pour supprimer une instance de cluster Amazon DocumentDB, procédez comme suit.

> **Note**
>
> La suppression d'une instance n'a aucun impact sur les données de votre cluster. Si vous supprimez l'instance principale, l'une des instances de réplication prend la relève en tant qu'instance inscriptible.

Supprimer une instance de cluster

1. À partir de AWS Toolkit for Visual Studio Code, développez Amazon DocumentDB pour afficher vos clusters Amazon DocumentDB.
2. Right-click l'instance de cluster que vous souhaitez supprimer, puis choisissez Supprimer pour ouvrir la boîte de dialogue de confirmation supprime-cluster-instance dans VS Code.
3. Suivez l'invite de confirmation, puis appuyez sur la **`Enter`** touche pour supprimer votre instance de cluster.
4. L'AWS explorateur affiche l'état de votre instance de cluster et se met à jour lorsque votre instance a été supprimée.

Afficher, ajouter et supprimer des balises

Les balises sont utilisées pour organiser et suivre les ressources de votre environnement. Pour afficher ou modifier les balises associées à votre cluster Amazon DocumentDB, effectuez l'une des procédures suivantes.

Affichage des balises de cluster

1. À partir de AWS Toolkit for Visual Studio Code, développez Amazon DocumentDB pour afficher vos clusters Amazon DocumentDB.

2. Right-click le cluster pour lequel vous souhaitez afficher les balises, puis choisissez Tags... pour ouvrir la boîte de **`your cluster name`** dialogue Tags pour.
3. Vos balises sont affichées dans la fenêtre de dialogue. Si aucune balise n'est associée à votre cluster, le message [Aucune étiquette attribuée] s'affiche.

Ajouter des balises à votre cluster

1. À partir de AWS Toolkit for Visual Studio Code, développez Amazon DocumentDB pour afficher vos clusters Amazon DocumentDB.
2. Right-click le cluster pour lequel vous souhaitez ajouter des balises, puis choisissez Tags... pour ouvrir la boîte de **`your cluster name`** dialogue Tags pour.
3. Choisissez le bouton Ajouter une étiquette... bouton pour ouvrir la boîte de dialogue Ajouter une étiquette dans VS Code.
4. Entrez une nouvelle étiquette dans le champ de texte, puis appuyez sur la touche Entrée pour continuer.
5. Entrez une valeur dans le champ de texte, puis appuyez sur Entrée pour ajouter la key/value paire à votre cluster.

Supprimer des balises de votre cluster

1. À partir de AWS Toolkit for Visual Studio Code, développez Amazon DocumentDB pour afficher vos clusters Amazon DocumentDB.
2. Right-click le cluster dont vous souhaitez supprimer les balises, puis choisissez Tags... pour ouvrir la boîte de **`your cluster name`** dialogue Tags pour.
3. Choisissez la balise Supprimer... bouton pour ouvrir la **`your cluster name`** boîte de dialogue Supprimer une balise de VS Code.
4. Choisissez la balise que vous souhaitez supprimer de la liste fournie pour la supprimer de votre cluster.

Modifier une classe d'instance

Pour modifier la classe d'une instance de cluster Amazon DocumentDB, effectuez la procédure suivante.

Modifier une classe d'instance

1. À partir de AWS Toolkit for Visual Studio Code, développez Amazon DocumentDB pour afficher vos clusters Amazon DocumentDB.
2. Right-click l'instance de cluster que vous souhaitez modifier, puis choisissez Modifier la classe... pour ouvrir la boîte de dialogue Sélectionner la classe d'instance dans VS Code.
3. Choisissez une nouvelle classe pour votre instance dans la liste afin de la mettre à jour.
4. L'AWS explorateur affiche l'état de votre instance de cluster et se met à jour lorsque la classe de votre instance est mise à jour.

Amazon Elastic Compute Cloud

Amazon Elastic Compute Cloud vous AWS Toolkit for Visual Studio Code permet de lancer et de vous connecter à vos EC2 instances Amazon depuis VS Code. Pour obtenir des informations détaillées sur Amazon EC2, consultez le document Qu'est-ce qu'Amazon EC2 ? rubrique du guide de l'utilisateur d'Amazon Elastic Compute Cloud.

Les rubriques suivantes décrivent comment utiliser AWS Application Builder à partir du AWS Toolkit for Visual Studio Code.

Rubriques

- Travailler avec Amazon Elastic Compute Cloud
- Résolution des problèmes liés à Amazon Elastic Compute Cloud

Travailler avec Amazon Elastic Compute Cloud

Les sections suivantes décrivent comment utiliser Amazon Elastic Compute Cloud dans le AWS Toolkit for Visual Studio Code.

Conditions préalables

Les fonctionnalités décrites dans cette rubrique du guide de l'utilisateur ont été testées sur des instances Amazon EC2 avec les systèmes d'exploitation suivants :

- Windows 2016+

> **Note**
>
> Ce système d'exploitation ne fonctionne que lors de la connexion d'un terminal VS Code. Cela ne fonctionne pas lors de la connexion d'une instance distante VS Code complète. Pour plus d'informations sur les terminaux VS Code et les instances distantes, consultez les rubriques Démarrage avec le terminal et Développement à distance VS Code dans la documentation VS Code.

- Amazon Linux 2023
- Ubuntu, version 22.04

Un SSH installé localement est requis pour ouvrir une connexion à distance à une instance Amazon EC2, mais pas pour ouvrir un terminal à une instance Amazon EC2.

Votre profil d'instance Amazon EC2 doit inclure les autorisations Gestion des identités et des accès AWS (IAM) suivantes.

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

> **Note**
>
> Les autorisations requises sont incluses dans la politique AWS gérée suivante.
>
> - `AmazonSSMManagedInstanceCore`
> - `AmazonSSMManagedEC2InstanceDefaultPolicy`

Affichage des instances Amazon EC2 existantes

Pour consulter vos instances Amazon EC2 existantes à partir de la AWS boîte à outils, procédez comme suit.

1. À partir de la AWS boîte à outils, développez l'explorateur de la AWS boîte à outils.
2. Développez la région qui contient les instances Amazon EC2 que vous souhaitez afficher.
3. Développez la rubrique EC2 pour afficher vos instances Amazon EC2 existantes.

Lancement d'une nouvelle instance Amazon EC2

Il existe trois méthodes pour créer une nouvelle instance Amazon EC2 à l'aide du AWS Toolkit.

Chaque flux de travail ouvre l'assistant Lancer une instance dans la AWS console. Pour obtenir des informations détaillées sur le lancement d'une nouvelle instance Amazon EC2 à partir de l'assistant de lancement d'une instance, consultez la section Lancer une instance EC2 à l'aide de l'assistant de lancement d'instance dans la rubrique console du guide de l'utilisateur Amazon Elastic Compute Cloud. Pour lancer une nouvelle instance Amazon EC2, effectuez l'une des procédures suivantes.

Lancement d'une nouvelle instance Amazon EC2 à partir de la palette de commandes VS Code

1. Depuis VS Code, ouvrez la palette de commandes VS Code en appuyant sur **`command + shift + P (Windows: ctrl + shift + P)`**
2. Dans la palette de commandes VS Code, recherchez la **`AWS: Launch EC2`** commande et sélectionnez-la lorsqu'elle apparaît dans la liste pour ouvrir l'invite Launch EC2 instance Select Region dans VS Code.
3. Dans l'invite Lancer l'instance EC2 Sélectionnez une région, choisissez la région dans laquelle vous souhaitez lancer votre nouvelle instance, puis confirmez que vous souhaitez ouvrir la AWS console dans votre navigateur Web par défaut.
4. À partir de la AWS console de votre navigateur Web par défaut, terminez le processus d'authentification pour accéder à l'assistant de lancement d'une instance.
5. Dans l'assistant de lancement d'une instance, complétez les sections requises, puis cliquez sur le bouton Lancer une instance pour lancer votre nouvelle instance Amazon EC2.
6. L' AWS explorateur est mis à jour pour afficher votre nouvelle instance Amazon EC2.

Lancement d'une nouvelle instance Amazon EC2 à partir du AWS Explorer

1. Développez le AWS Toolkit Explorer, puis développez la région dans laquelle vous souhaitez créer la nouvelle instance Amazon EC2.
2. Développez ou passez la souris sur l'en-tête EC2, puis choisissez l'icône + (Lancer l'instance EC2).
3. Lorsque vous y êtes invité, confirmez que vous souhaitez ouvrir la AWS console dans votre navigateur Web par défaut.
4. À partir de la AWS console de votre navigateur Web, terminez le processus d'authentification pour accéder à l'assistant de lancement d'une instance.
5. Dans l'assistant de lancement d'une instance, complétez les sections requises, puis cliquez sur le bouton Lancer une instance pour lancer votre nouvelle instance Amazon EC2.
6. L' AWS explorateur est mis à jour pour afficher votre nouvelle instance Amazon EC2.

Lancer une nouvelle instance Amazon EC2 depuis le menu contextuel (clic droit)

1. Développez le AWS Toolkit Explorer, puis développez la région dans laquelle vous souhaitez créer la nouvelle instance Amazon EC2.
2. Right-click l'en-tête EC2, puis choisissez Lancer l'instance EC2.
3. Lorsque vous y êtes invité, confirmez que vous souhaitez ouvrir la AWS console dans votre navigateur Web par défaut.
4. À partir de la AWS console de votre navigateur Web, terminez le processus d'authentification pour accéder à l'assistant de lancement d'une instance.
5. Dans l'assistant de lancement d'une instance, complétez les sections requises, puis cliquez sur le bouton Lancer une instance pour lancer votre nouvelle instance Amazon EC2.
6. L' AWS explorateur est mis à jour pour afficher votre nouvelle instance Amazon EC2.

Connexion de VS Code à une instance Amazon EC2

Il existe trois manières de se connecter à une instance Amazon EC2 depuis VS Code. Pour connecter VS Code à votre instance EC2, effectuez l'une des procédures suivantes.

Connexion de VS Code à une instance Amazon EC2 depuis la palette de commandes

1. Depuis VS Code, ouvrez la palette de commandes VS Code en appuyant sur **`command + shift + P (Windows: ctrl + shift + P)`**

2. Dans la palette de commandes VS Code, recherchez la **`AWS: Connect VS Code to EC2 instance...`** commande et sélectionnez-la lorsqu'elle apparaît dans la liste pour ouvrir l'invite Select EC2 Instance dans VS Code.
3. Dans l'invite Sélectionner une instance EC2, choisissez la région qui contient l'instance à laquelle vous souhaitez vous connecter, puis choisissez l'instance à laquelle vous souhaitez vous connecter.
4. VS Code affiche l'état pendant l'établissement de la connexion.
5. Une nouvelle fenêtre s'ouvre pour afficher votre instance Amazon EC2 lorsque la connexion est terminée.

Connexion de VS Code à une instance Amazon EC2 depuis AWS Explorateur.

1. Développez le AWS Toolkit Explorer, puis développez la région qui contient l'instance Amazon EC2 à laquelle vous souhaitez vous connecter.
2. Passez la souris sur l'instance Amazon EC2, puis choisissez l'icône (Connecter VS Code à l'instance EC2).

   > **Note**
   >
   > Vous pouvez également choisir l'icône (Connecter VS Code à l'instance EC2) dans l'en-tête du service EC2 dans l' AWS Explorateur.

3. VS Code affiche l'état pendant l'établissement de la connexion.
4. Une nouvelle fenêtre s'ouvre pour afficher votre instance Amazon EC2 lorsque la connexion est terminée.

Connexion de VS Code à une instance Amazon EC2 depuis le menu contextuel

1. Développez le AWS Toolkit Explorer, puis développez la région qui contient l'instance Amazon EC2 à laquelle vous souhaitez vous connecter.
2. Right-click l'instance Amazon EC2 à laquelle vous souhaitez vous connecter, puis choisissez Connect VS Code to EC2 instance.

> **Note**
>
> Vous pouvez également cliquer avec le bouton droit sur l'en-tête du service EC2 dans l'AWS Explorateur et choisir l'instance Connect VS Code to EC2.

3. VS Code affiche l'état pendant l'établissement de la connexion.
4. Une nouvelle fenêtre s'ouvre pour afficher votre instance Amazon EC2 lorsque la connexion est terminée.

Ouverture d'un terminal sur une instance Amazon EC2.

Il existe trois manières de se connecter à une instance Amazon EC2 depuis le terminal VS Code.

Connexion de VS Code à une instance Amazon EC2 depuis la palette de commandes

1. Depuis VS Code, ouvrez la palette de commandes VS Code en appuyant sur **`command + shift + P (Windows: ctrl + shift + P)`**
2. Dans la palette de commandes VS Code, recherchez la **`AWS:Open terminal to EC2 instance...`** commande et sélectionnez-la lorsqu'elle apparaît dans la liste pour ouvrir l'invite Select EC2 Instance dans VS Code.
3. Dans l'invite Sélectionner une instance EC2, choisissez la région contenant l'instance que vous souhaitez ouvrir dans le terminal, puis choisissez l'instance.
4. VS Code affiche l'état pendant l'établissement de la connexion.
5. Le terminal VS Code s'ouvre pour afficher votre nouvelle session lorsque la connexion est terminée.

Ouverture d'une instance Amazon EC2 dans le terminal VS Code à partir du AWS Explorateur.

1. Développez le AWS Toolkit Explorer, puis développez la région qui contient l'instance Amazon EC2 à laquelle vous souhaitez vous connecter.
2. Passez la souris sur l'instance Amazon EC2, puis choisissez le (Ouvrir le terminal sur l'instance EC2...) icône.

> **Note**
>
> Vous pouvez également choisir le (Ouvrir le terminal à l'instance EC2...) icône de l'en-tête du service EC2 dans l' AWS Explorateur.

3. VS Code affiche l'état pendant l'établissement de la connexion.
4. Le terminal VS Code s'ouvre pour afficher votre nouvelle session lorsque la connexion est terminée.

Ouverture d'une instance Amazon EC2 dans le terminal VS Code à partir du menu contextuel

1. Développez le AWS Toolkit Explorer, puis développez la région qui contient l'instance Amazon EC2 que vous souhaitez ouvrir dans le terminal VS Code.
2. Right-click l'instance Amazon EC2 que vous souhaitez ouvrir dans le terminal, puis choisissez Open terminal to EC2 instance... .

> **Note**
>
> Vous pouvez également cliquer avec le bouton droit sur l'en-tête du service EC2 dans l' AWS Explorateur et choisir l'option Ouvrir le terminal dans l'instance EC2... .

3. VS Code affiche l'état pendant l'établissement de la connexion.
4. Le terminal VS Code s'ouvre pour afficher votre nouvelle session lorsque la connexion est terminée.

Démarrage ou redémarrage d'une instance Amazon EC2

Il existe trois méthodes pour démarrer ou redémarrer une instance Amazon EC2.

Redémarrage d'une instance Amazon EC2 à partir de la palette de commandes

1. Depuis VS Code, ouvrez la palette de commandes VS Code en appuyant sur **`command + shift + P (Windows: ctrl + shift + P)`**
2. Dans la palette de commandes VS Code, recherchez la **`AWS: Reboot EC2 instance`** commande et sélectionnez-la lorsqu'elle apparaît dans la liste pour ouvrir l'invite Select EC2 Instance dans VS Code.

> **Note**
>
> Pour démarrer une instance qui n'est pas en cours d'exécution, vous devez choisir la **`AWS: Start EC2 instance`** commande. La **`AWS: Reboot EC2 instance`** commande ne redémarre que les instances en cours d'exécution.

3. À l'invite Sélectionnez une instance EC2, choisissez la région qui contient l'instance que vous souhaitez démarrer ou redémarrer.
4. VS Code affiche l'état pendant le redémarrage de l'instance.
5. L' AWS Explorateur est mis à jour pour indiquer que votre instance est en cours d'exécution une fois le redémarrage terminé.

Démarrage ou redémarrage d'une instance Amazon EC2 à partir du AWS Explorer

1. Développez le AWS Toolkit Explorer, puis développez la région qui contient l'instance Amazon EC2 que vous souhaitez démarrer ou redémarrer.
2. Passez la souris sur l'instance Amazon EC2, puis choisissez l'icône (Redémarrer l'instance EC2).

> **Note**
>
> Si l'instance est arrêtée, les seules options sont l'icône (Démarrer l'instance EC2)

3. VS Code affiche l'état pendant le redémarrage de l'instance.
4. L' AWS Explorateur est mis à jour pour indiquer que votre instance est en cours d'exécution une fois le redémarrage terminé.

Démarrage ou redémarrage d'une instance Amazon EC2 depuis le menu contextuel

1. Développez le AWS Toolkit Explorer, puis développez la région qui contient l'instance Amazon EC2 que vous souhaitez démarrer ou redémarrer.
2. Right-click l'instance Amazon EC2 à laquelle vous souhaitez vous connecter, puis choisissez Redémarrer l'instance EC2.

> **Note**
>
> Si l'instance est arrêtée, les seules options sont l'option Démarrer l'instance EC2.

3. VS Code affiche l'état pendant le redémarrage de l'instance.
4. L' AWS Explorateur est mis à jour pour indiquer que votre instance est en cours d'exécution une fois le redémarrage terminé.

Arrêt d'une instance Amazon EC2

Il existe trois méthodes pour arrêter une instance Amazon EC2.

Arrêter une instance Amazon EC2 depuis la palette de commandes

1. Depuis VS Code, ouvrez la palette de commandes VS Code en appuyant sur **`command + shift + P (Windows: ctrl + shift + P)`**
2. Dans la palette de commandes VS Code, recherchez la **`AWS: Stop EC2 instance`** commande et sélectionnez-la lorsqu'elle apparaît dans la liste pour ouvrir l'invite Select EC2 Instance dans VS Code.
3. À l'invite de sélection d'une instance EC2, choisissez la région qui contient l'instance que vous souhaitez arrêter.
4. VS Code affiche l'état pendant l'arrêt de l'instance.
5. L' AWS Explorateur est mis à jour pour indiquer que votre instance est arrêtée.

Arrêter une instance Amazon EC2 à partir du AWS Explorer

1. Développez le AWS Toolkit Explorer, puis développez la région qui contient l'instance Amazon EC2 que vous souhaitez arrêter.
2. Passez la souris sur l'instance Amazon EC2, puis choisissez l'icône (Arrêter l'instance EC2).
3. VS Code affiche l'état pendant l'arrêt de l'instance.
4. L' AWS Explorateur est mis à jour pour indiquer que votre instance s'est arrêtée.

Arrêter une instance Amazon EC2 depuis le menu contextuel

1. Développez le AWS Toolkit Explorer, puis développez la région qui contient l'instance Amazon EC2 que vous souhaitez arrêter.
2. Right-click l'instance Amazon EC2 à laquelle vous souhaitez vous connecter, puis choisissez Redémarrer l'instance EC2.
3. VS Code affiche l'état pendant l'arrêt de l'instance.
4. L' AWS Explorateur est mis à jour pour indiquer que votre instance s'est arrêtée.

Copier l'ID d'instance

Pour copier un ID d'instance, procédez comme suit.

1. Right-click l'instance à partir de laquelle vous souhaitez copier l'identifiant.
2. Choisissez Copier l'ID d'instance.
3. L'ID d'instance est copié dans votre presse-papiers local.

Copier le nom

Pour copier le nom d'une instance, procédez comme suit.

1. Right-click l'instance dont vous souhaitez copier le nom.
2. Choisissez Copier le nom de l'instance.
3. Le nom de l'instance est copié dans votre presse-papiers local.

Copier l'ARN

Pour copier l'ARN d'une instance, procédez comme suit.

1. Right-click l'instance à partir de laquelle vous souhaitez copier l'ARN.
2. Choisissez Copier l'ARN de l'instance.
3. L'ARN de l'instance est copié dans votre presse-papiers local.

Résolution des problèmes liés à Amazon Elastic Compute Cloud

Les sections suivantes décrivent comment résoudre les problèmes connus qui peuvent survenir lors de l'utilisation d'Amazon Elastic Compute Cloud dans le AWS Toolkit for Visual Studio Code. Pour obtenir des informations détaillées sur la résolution des problèmes spécifiques au service Amazon EC2, consultez la rubrique Résoudre les problèmes liés aux instances Amazon EC2 dans le guide de l'utilisateur d'Amazon Elastic Compute Cloud.

Débogage général

Si vous rencontrez un problème de connexion à distance pour quelque raison que ce soit, commencez par vérifier si une AWS Systems Manager connexion peut être établie à partir de la AWS console.

Pour vous connecter à une instance Amazon EC2 via Systems Manager depuis la AWS console, procédez comme suit.

1. Dans votre navigateur Web, accédez à la AWS console.
2. Terminez l'authentification pour passer au lancement de la AWS console EC2.
3. Dans le volet de navigation Amazon EC2, sélectionnez Instances.
4. Cochez la case située à côté de l'instance à laquelle vous souhaitez vous connecter.
5. Cliquez sur le bouton Connect pour ouvrir l'écran Connect to instance dans un nouvel onglet du navigateur.

   > **Note**
   >
   > Vous ne pouvez vous connecter à une instance que si elle est en cours d'exécution. Si vous ne parvenez pas à sélectionner le bouton Connect, assurez-vous que votre instance est en cours d'exécution.

6. Sur l'écran Connect to instance, choisissez l'onglet Session Manager, puis cliquez sur le bouton Connect pour ouvrir la connexion Systems Manager dans l'onglet actuel de votre navigateur.

   > **Note**
   >
   > Si vous avez récemment démarré votre instance et que l'option de connexion au Systems Manager n'est pas disponible, vous devrez peut-être attendre quelques minutes supplémentaires avant que l'option ne soit disponible.

L'instance cible n'est pas en cours d'exécution

Pour se connecter à une instance Amazon EC2 depuis le terminal ou une connexion à distance, l'instance doit être en cours d'exécution. Avant de tenter de vous connecter à votre instance depuis le AWS Toolkit, démarrez-la depuis l' AWS explorateur Console de gestion AWS, ou AWS Command Line Interface.

L'instance cible n'a pas de rôle IAM ou possède un rôle IAM avec des autorisations inappropriées

Pour vous connecter à votre instance Amazon EC2, celle-ci doit disposer d'un rôle IAM associé aux autorisations appropriées. Si vous tentez de vous connecter à une instance à laquelle aucun rôle IAM n'est attaché, VS Code vous en informe.

Si vous tentez de vous connecter à une instance dotée d'un rôle IAM mais ne disposant pas des autorisations nécessaires, vous êtes invité à ajouter le minimum d'actions nécessaires sous forme de politique intégrée au rôle IAM existant. Après avoir mis à jour la politique intégrée, vous êtes connecté à votre instance. Pour obtenir des informations détaillées sur les rôles IAM, les autorisations et l'attachement d'un rôle à une instance, consultez la rubrique sur les rôles IAM pour Amazon EC2 dans le guide de l'utilisateur d'Amazon Elastic Compute Cloud et la rubrique Étape 2 : vérifier ou ajouter des autorisations d'instance pour Session Manager dans le guide de l'utilisateur de Systems AWS Manager.

L'exemple suivant contient les actions minimales nécessaires.

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

> **Note**
>
> Les autorisations requises sont incluses dans la politique AWS gérée suivante.
>
> - `AmazonSSMManagedEC2InstanceDefaultPolicy`

- `AmazonSSMManagedInstanceCore`

Aucun agent Systems Manager n'est en cours d'exécution sur l'instance cible

Vous pouvez rencontrer ce problème pour différentes raisons. Pour résoudre le problème, commencez par redémarrer l'instance et effectuez une nouvelle tentative de connexion. Vous pouvez également démarrer manuellement une connexion initiale via une méthode de connexion autre que SSM. Pour des informations plus détaillées sur Systems Manager, consultez la rubrique Working with Systems Manager Agent dans le AWS Systems Manager.

Au démarrage, le statut d'Amazon EC2 indique qu'il est en cours d'exécution, mais que les connexions ne passent pas

Si vous avez récemment démarré ou créé un nouveau rôle IAM pour une instance et que vous ne parvenez pas à établir de connexion, attendez quelques minutes supplémentaires avant de tenter à nouveau d'établir une connexion.

Utilisation d'Amazon Elastic Container Registry

Amazon Elastic Container Registry (Amazon ECR) est AWS un service géré de registre de conteneurs sécurisé et évolutif. Plusieurs fonctions du service Amazon ECR sont accessibles depuis le Toolkit for VS Code Explorer.

- Création d'un référentiel.
- Création d'un AWS App Runner service pour votre référentiel ou votre image balisée.
- Accès à la balise d'image et au référentiel URIs ou ARNs.
- Suppression des balises d'image et des référentiels.

Vous pouvez également accéder à la gamme complète des fonctions Amazon ECR via la console VS Code en intégrant la AWS CLI et d'autres plateformes à VS Code.

Pour plus d'informations sur Amazon ECR, veuillez consulter Qu'est-ce qu'Amazon ECR ? dans le guide de l'utilisateur Amazon Elastic Container Registry.

Rubriques

- Utilisation d'Amazon Elastic Container Registry
- Création d'un service App Runner via Amazon ECR

Utilisation d'Amazon Elastic Container Registry

Vous pouvez accéder au service Amazon Elastic Container Registry (Amazon ECR) directement depuis l' AWS Explorateur dans VS Code et l'utiliser pour envoyer une image de programme vers un référentiel Amazon ECR. Pour commencer, vous devez suivre les étapes suivantes :

1. Créez un Dockerfile contenant les informations nécessaires à la création d'une image.
2. Créez une image à partir de ce Dockerfile et balisez l'image pour le traitement.
3. Créez un référentiel dans votre instance Amazon ECR.
4. Transmettez l'image balisée à votre référentiel.

Conditions préalables

Vous devez suivre ces étapes pour accéder au service Amazon ECR depuis VS Code Explorer.

Créer un utilisateur IAM

Avant de pouvoir accéder à un AWS service, tel qu'Amazon ECR, vous devez fournir des informations d'identification. Cela permet au service de déterminer si vous êtes autorisé à accéder à ses ressources. Nous vous déconseillons d'y accéder AWS directement via les informations d'identification de votre AWS compte root. Utilisez plutôt Gestion des identités et des accès AWS (IAM) pour créer un utilisateur IAM, puis ajoutez-le à un groupe IAM doté d'autorisations administratives. Vous pouvez ensuite accéder à AWS l'aide d'une URL spéciale et des informations d'identification de l'utilisateur IAM.

Si vous vous êtes inscrit AWS mais que vous n'avez pas créé d'utilisateur IAM pour vous-même, vous pouvez en créer un à l'aide de la console IAM.

Afin de créer un utilisateur administrateur, choisissez l'une des options suivantes :

Choisissez un moyen de gérer votre administrateur	À	En	Vous pouvez également
Dans IAM Identity Center (Recommandé)	Utiliser des informations d'identification à court terme pour accéder à AWS. C'est conforme aux bonnes pratiques en matière de sécurité. Pour plus d'informations sur les bonnes pratiques, consultez Bonnes pratiques de sécurité dans IAM dans le Guide de l'utilisateur IAM.	Suivant les instructions fournies dans Mise en route dans le Guide de l'utilisateur AWS IAM Identity Center .	Configurez l'accès par programmation en configurant les éléments AWS CLI à utiliser AWS IAM Identity Center dans le Guide de l'AWS Command Line Interface utilisateur.
Dans IAM (Non recommandé)	Utiliser les informations d'identification à long terme pour accéder à AWS.	Suivant les instructions fournies dans Création d'un utilisateur IAM pour l'accès d'urgence dans le Guide de l'utilisateur IAM.	Configurer l'accès par programmation en suivant les instructions fournies dans Gestion des clés d'accès pour les utilisateurs IAM dans le Guide de l'utilisateur IAM.

Pour vous connecter en tant que nouvel utilisateur IAM, déconnectez-vous de la AWS console, puis utilisez l'URL suivante. Dans l'URL suivante, où your_aws_account_id est votre numéro de AWS compte sans les tirets (par exemple, si c'est votre numéro de compte, votre identifiant de AWS compte est) : `1234-5678-9012` AWS `123456789012`

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

Saisissez le nom utilisateur et le mot de passe IAM que vous venez de créer. Lorsque vous êtes connecté, la barre de navigation affiche « votre_nom_utilisateur @ votre_id_de_compte_aws ».

Si vous ne souhaitez pas que l'URL de votre page de connexion contienne l'identifiant de votre AWS compte, vous pouvez créer un alias de compte. Dans le tableau de bord IAM, choisissez Personnaliser et saisissez un alias de compte. Il peut s'agir du nom de votre entreprise. Pour plus d'informations, consultez l'ID de votre AWS compte et son alias dans le Guide de l'utilisateur IAM.

Pour vous connecter après avoir créé un alias de compte, utilisez l'URL suivante :

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

Pour vérifier le lien de connexion des utilisateurs IAM pour votre compte, ouvrez la console IAM et vérifiez le lien sous Lien de connexion des utilisateurs IAM dans le tableau de bord.

Pour en savoir plus sur IAM, consultez le guide de l'utilisateur Gestion des identités et des accès AWS.

Installation et configuration de Docker

Vous pouvez installer et configurer Docker en sélectionnant votre système d'exploitation préféré dans le guide de l'utilisateur d'installation du moteur Docker et en suivant les instructions.

Installation et configuration AWS CLI version 2

Installez et configurez la version 2 de la AWS CLI en sélectionnant votre système d'exploitation préféré dans le guide de l'utilisateur d'installation, de mise à jour et de désinstallation de la AWS CLI version 2.

1. Création d'un fichier Dockerfile

Docker utilise un fichier appelé Dockerfile pour définir une image qui peut être transmise et stockée sur un référentiel distant. Avant de pouvoir télécharger une image dans un référentiel ECR, vous devez créer un Dockerfile, puis créer une image à partir de ce Dockerfile.

Création d'un fichier Dockerfile

1. Utilisez l'explorateur Toolkit for VS Code pour accéder au répertoire dans lequel vous souhaitez stocker votre Dockerfile.

2. Créez un nouveau fichier appelé Dockerfile.

> **Note**
>
> VS Code peut vous demander de sélectionner un type de fichier ou une extension de fichier. Dans ce cas, sélectionnez le texte en clair. Vs Code possède une extension « dockerfile ». Cependant, nous vous déconseillons de l'utiliser. Cela est dû au fait que l'extension peut provoquer des conflits avec certaines versions de Docker ou d'autres applications associées.

Modifiez votre Dockerfile à l'aide de VS Code

Si votre fichier Dockerfile possède une extension de fichier, ouvrez le menu contextuel (clic droit) du fichier et supprimez l'extension de fichier.

Une fois l'extension de fichier supprimée de votre Dockerfile :

1. Ouvrez le Dockerfile vide directement dans VS Code.
2. Copiez le contenu de l'exemple suivant dans votre Dockerfile :

   Example Modèle d'image de fichier Dockerfile

```
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
```

Il s'agit d'un fichier Dockerfile qui utilise une image Ubuntu 18.04. Les instructions RUN mettent à jour les caches des packages. Installez les packages logiciels pour le serveur web et écrivent ensuite le contenu « Hello World ! » à la racine du document du serveur web. L'instruction EXPOSE expose le port 80 sur le conteneur et l'instruction CMD démarre le serveur web.

3. Enregistrez votre fichier Dockerfile.

> **Important**
>
> Assurez-vous que le nom de votre Dockerfile n'est pas associé à une extension. Un Dockerfile avec des extensions peut provoquer des conflits avec certaines versions de Docker ou d'autres applications associées.

2. Créez votre image à partir de votre Dockerfile

Le Dockerfile que vous avez créé contient les informations nécessaires pour créer une image pour un programme. Avant de pouvoir transférer cette image vers votre instance Amazon ECR, vous devez d'abord créer l'image.

Créez une image à partir de votre Dockerfile

1. Utilisez l'interface de ligne de commande Docker ou une interface de ligne de commande intégrée à votre instance de Docker pour accéder au répertoire contenant votre Dockerfile.
2. Exécutez la commande Docker build pour créer l'image définie dans votre Dockerfile.

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

3. Exécutez la commande Docker images pour vérifier que l'image a été créée correctement.

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

Example exemple de sortie :

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

4.

> **Note**
>
> Cette étape n'est pas nécessaire pour créer ou diffuser votre image, mais vous pouvez voir comment fonctionne l'image du programme lorsqu'elle est exécutée.

Pour exécuter l'image nouvellement créée, utilisez la commande Docker run.

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

L'option -p spécifiée dans l'exemple précédent mappe le port exposé 80 du conteneur au port 80 du système hôte. Si vous exécutez Docker localement, accédez à:80 à http://localhost l'aide de votre navigateur Web. Si le programme s'est correctement exécuté, un message « Hello World ! » la déclaration s'affiche.

Pour de plus amples informations sur la commande Docker run, consultez la Référence Docker Run dans le manuel de référence Docker.

3. Création d'un nouveau référentiel

Pour charger votre image dans votre instance Amazon ECR, créez un nouveau référentiel dans lequel elle peut être stockée.

Création d'un nouveau référentiel Amazon ECR

1. Dans la barre d'activité de VS Code, choisissez l'icône de la AWS boîte à outils.
2. Développez le menu de l' AWS Explorateur.
3. Localisez la AWS région par défaut associée à votre AWS compte. Ensuite, sélectionnez-le pour afficher la liste des services qui utilisent le Toolkit for VS Code.

4. Choisissez l'option ECR + pour démarrer le processus de création d'un nouveau référentiel.
5. Suivez les instructions pour terminer le processus.
6. Une fois l'opération terminée, vous pouvez accéder à votre nouveau référentiel depuis la section ECR du menu AWS Explorer.

4. Poussez, tirez et supprimez des images

Après avoir créé une image à partir de votre Dockerfile et créé un référentiel, vous pouvez transférer votre image dans votre référentiel Amazon ECR. En outre, à l'aide de l' AWS Explorateur avec Docker et de l' AWS interface de ligne de commande, vous pouvez effectuer les opérations suivantes :

- Extraire une image d'un référentiel.
- Supprimer une image stockée dans votre référentiel.
- Supprimer votre référentiel.

Authentifiez Docker avec votre registre par défaut

L'authentification est requise pour échanger des données entre les instances Amazon ECR et Docker. Pour authentifier Docker avec votre registre :

1. Ouvrez un système d'exploitation en ligne de commande connecté à votre instance de AWS CLI.
2. Utilisez la méthode get-login-password pour vous authentifier auprès de votre registre ECR privé.

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

> **Important**
>
> Dans la commande précédente, vous devez mettre à jour **`AWS_account_id`** à la fois les informations **`region`** et les informations spécifiques à votre AWS compte.

Marquez et transférez une image vers votre référentiel

Après avoir authentifié Docker avec votre instance de AWS, transférez une image vers votre référentiel.

1. Utilisez la commande Docker images pour afficher les images que vous avez stockées localement et identifier celle que vous souhaitez baliser.

```
docker images
```

Example exemple de sortie :

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

2. Créez votre image Docker à l'aide de la commande Docker tag.

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

3. Transférez l'image balisée vers votre référentiel à l'aide de la commande Docker tag.

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

Example exemple de sortie :

```
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
```

Une fois que votre image balisée a été chargée avec succès dans votre référentiel, elle est visible dans le menu de l' AWS Explorateur.

Extraire une image depuis Amazon ECR

- Vous pouvez extraire une image vers votre instance locale de la commande Docker tag.

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

Example exemple de sortie :

```
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
```

Supprimer une image de votre référentiel Amazon ECR

Il existe deux méthodes pour supprimer une image de VS Code. La première méthode consiste à utiliser l' AWS explorateur.

1. Depuis l' AWS Explorateur, développez le menu ECR
2. Développez le référentiel dont vous souhaitez supprimer une image
3. Choisissez la balise d'image associée à l'image que vous souhaitez supprimer, en ouvrant le menu contextuel (clic droit)
4. Choisissez la balise Supprimer... option pour supprimer toutes les images stockées associées à cette balise

Supprimez une image à l'aide du AWS INTERFACE DE LIGNE DE COMMANDE (CLI)

- Vous pouvez également supprimer une image de votre dépôt à l'aide de la commande AWS ecr batch-delete-image.

```
AWS ecr batch-delete-image \
      --repository-name hello-world \
```

```
      --image-ids imageTag=latest
```

Example exemple de sortie :

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

Supprimer un référentiel de votre instance Amazon ECR

Il existe deux méthodes pour supprimer un référentiel de VS Code. La première méthode consiste à utiliser l' AWS explorateur.

1. Depuis l' AWS Explorateur, développez le menu ECR
2. Choisissez le référentiel que vous souhaitez supprimer en ouvrant le menu contextuel (clic droit)
3. Choisissez l'option Supprimer le référentiel... option vers le référentiel choisi

Supprimer un référentiel Amazon ECR du AWS INTERFACE DE LIGNE DE COMMANDE (CLI)

- Vous pouvez supprimer un dépôt à l'aide de la commande AWS ecr delete-repository.

> **Note**
>
> Par défaut, vous ne pouvez pas supprimer un référentiel contenant des images. Cependant, le drapeau --force le permet.

```
AWS ecr delete-repository \
```

```
      --repository-name hello-world \
      --force
```

Example exemple de sortie :

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

Création d'un service App Runner via Amazon ECR

La rubrique suivante décrit comment créer et lancer un AWS App Runner service à partir du nœud Amazon Elastic Container Registry (Amazon ECR), dans le AWS Toolkit for Visual Studio Code. Pour obtenir des informations détaillées sur les services Amazon ECR AWS App Runner et Amazon ECR, consultez les guides AWS App Runner de l'utilisateur Amazon ECR.

Conditions préalables

Avant de pouvoir créer et lancer un AWS App Runner depuis Amazon ECR dans le AWS Toolkit, vous devez effectuer les opérations suivantes. Pour obtenir un guide détaillé sur la manière d'effectuer ces procédures, consultez la rubrique Utilisation d'Amazon Elastic Container Registry dans ce guide de l'utilisateur.

1. Créez un `dockerfile`.
2. Créez une image à partir de votre`dockerfile`.
3. Créez un nouveau référentiel.
4. Étiquetez et transférez une image vers votre référentiel.

Création d'un AWS App Runner service à partir d'un référentiel Amazon ECR existant

La procédure suivante décrit comment créer un AWS App Runner service à partir d'un référentiel Amazon ECR existant, dans le AWS Toolkit.

1. Dans l' AWS Explorateur, développez la région qui contient le référentiel Amazon ECR à partir duquel vous souhaitez créer un AWS App Runner service.
2. Développez le nœud de service Amazon ECR pour afficher vos référentiels Amazon ECR.
3. Ouvrez le menu contextuel (cliquez avec le bouton droit) du référentiel Amazon ECR ou de l'image du référentiel à partir duquel vous souhaitez créer un AWS App Runner service.
4. Dans le menu contextuel, choisissez Create App Runner Service pour ouvrir l'assistant de AWS App Runner création dans VS Code
5. Dans Entrez un port pour le nouveau service (1/5), saisissez le numéro de port que vous souhaitez utiliser, puis appuyez **`Enter`** pour continuer.
6. Dans Configurer les variables d'environnement (2/5), choisissez Utiliser le fichier... pour naviguer, sélectionnez Parcourir vos fichiers locaux ou choisissez Ignorer pour ignorer cette étape.
7. Dans Sélectionnez un rôle à extraire de ECR (3/5), choisissez un rôle IAM existant dans la liste.

   > **Note**
   >
   > Le rôle AppRunnerECRAccessRole d'accès est requis pour créer un AWS App Runner service à partir d'un registre privé Amazon ECR. Si aucun rôle valide n'est disponible dans la liste, choisissez le + (Créer un rôle...) icône pour créer et attribuer automatiquement AppRunnerECRAccessRole à votre registre.

8. Dans Nommez votre service (4/5), saisissez le nom de votre nouveau service, puis appuyez **`Enter`** pour continuer.
9. Dans Sélectionnez la configuration de l'instance (5/5), choisissez la **`Memory`** configuration **`vCPU`** et dans la liste pour créer votre nouveau service.
10. Dans l' AWS Explorateur, développez le nœud de service App Runner pour afficher vos AWS App Runner ressources. Lorsque votre nouveau service a été créé avec succès, le statut passe automatiquement à Running.

Fonctionnement d'Amazon Elastic Container Service

AWS Toolkit for Visual Studio Code Il fournit un certain support pour Amazon Elastic Container Service (Amazon ECS). Le Toolkit for VS Code vous aide dans certaines tâches liées à Amazon ECS, telles que la création de définitions de tâches.

Rubriques

- Utilisation de IntelliSense fichiers de définition de tâches pour Amazon ECS
- Amazon Elastic Container Service Exec dans AWS Toolkit for Visual Studio Code

Utilisation de IntelliSense fichiers de définition de tâches pour Amazon ECS

Lorsque vous travaillez avec Amazon Elastic Container Service (Amazon ECS), vous pouvez notamment créer des définitions de tâches, comme décrit dans la section Création d'une définition de tâche du guide du développeur Amazon Elastic Container Service. Lorsque vous installez le AWS Toolkit for Visual Studio Code, l'installation inclut des IntelliSense fonctionnalités pour les fichiers de définition de tâches Amazon ECS.

Conditions préalables

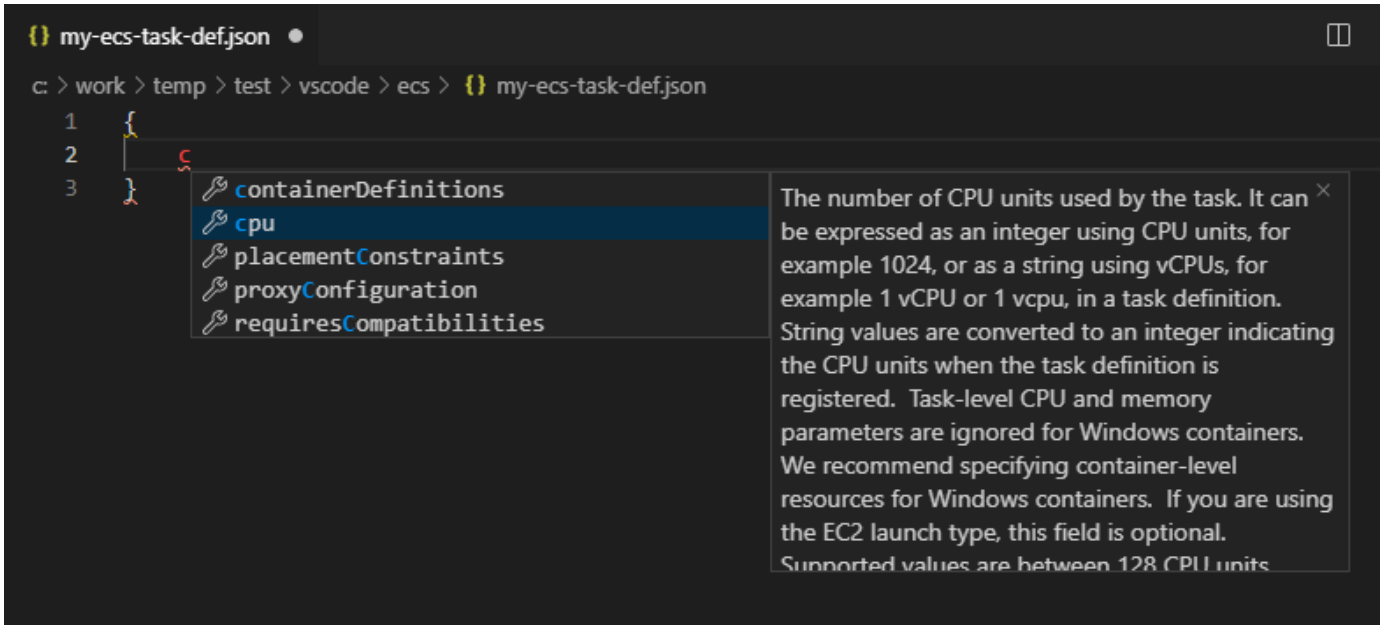
- Assurez-vous que votre système répond aux prérequis spécifiés dans Installation du Toolkit for VS Code.

Utilisation IntelliSense dans les fichiers de définition de tâches Amazon ECS

L'exemple suivant vous montre comment tirer parti des fichiers de définition IntelliSense de tâches Amazon ECS.

1. Créez un fichier JSON pour la définition de votre tâche Amazon ECS. Le nom de ce fichier doit contenir `ecs-task-def.json` à la fin, mais peut comporter des caractères supplémentaires au début.

   Pour cet exemple, créez un fichier nommé `my-ecs-task-def.json`
2. Ouvrez le fichier dans un éditeur VS Code et entrez les accolades initiales.
3. Entrez la lettre « c » comme si vous vouliez ajouter `cpu` à la définition. Observez la IntelliSense boîte de dialogue qui s'ouvre, similaire à la suivante.



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

Vous pouvez émettre des commandes uniques dans un conteneur Amazon Elastic Container Service (Amazon ECS) avec AWS Toolkit for Visual Studio Code le, en utilisant la fonctionnalité Amazon ECS Exec.

> **Important**
>
> L'activation et la désactivation d'Amazon ECS Exec modifient l'état des ressources de votre AWS compte. Cela inclut l'arrêt et le redémarrage du service. La modification de l'état des ressources alors qu'Amazon ECS Exec est activé peut entraîner des résultats imprévisibles. Pour plus d'informations sur Amazon ECS, consultez le guide du développeur Utilisation d'Amazon ECS Exec pour le débogage.

Prérequis Amazon ECS Exec

Avant de pouvoir utiliser la fonctionnalité Amazon ECS Exec, certaines conditions préalables doivent être remplies.

Exigences Amazon ECS

Selon que vos tâches sont hébergées sur Amazon EC2 ou AWS Fargate Amazon ECS Exec, les exigences en matière de version diffèrent.

- Si vous utilisez Amazon EC2, vous devez utiliser une AMI optimisée pour Amazon ECS publiée après le 20 janvier 2021 avec une version 1.50.2 ou supérieure de l'agent. Des informations supplémentaires sont disponibles pour vous dans le guide du développeur Amazon ECS optimized AMIs.
- Si vous utilisez AWS Fargate, vous devez utiliser la version 1.4.0 ou supérieure de la plateforme. Des informations supplémentaires sur les exigences de Fargate sont à votre disposition dans le guide du développeur Versions de plateforme AWS Fargate.

AWS configuration du compte et autorisations IAM

Pour utiliser la fonctionnalité Amazon ECS Exec, vous devez disposer d'un cluster Amazon ECS existant associé à votre AWS compte. Amazon ECS Exec utilise Systems Manager pour établir une connexion avec les conteneurs de votre cluster et nécessite des autorisations de rôle IAM spécifiques aux tâches pour communiquer avec le service SSM.

Vous trouverez des informations sur les rôles et les politiques IAM, spécifiques à Amazon ECS Exec, dans le guide du développeur des autorisations IAM requises pour ECS Exec.

Utiliser Amazon ECS Exec

Vous pouvez activer ou désactiver Amazon ECS Exec directement depuis l' AWS explorateur du Toolkit for VS Code. Lorsque vous avez activé Amazon ECS Exec, vous pouvez choisir des conteneurs dans le menu Amazon ECS, puis exécuter des commandes sur ceux-ci.

Activation d'Amazon ECS Exec

1. Dans l' AWS explorateur, recherchez et développez le menu Amazon ECS.
2. Développez le cluster à partir du service à modifier.
3. Ouvrez le menu contextuel (clic droit) à partir d'un service et choisissez Enable Command Exec.

> **Important**
>
> Cette opération démarre un nouveau déploiement de votre service et peut prendre quelques minutes. Pour de plus amples informations, veuillez consulter la note au début de cette section.

Désactivation Amazon ECS Exec

1. Dans l' AWS explorateur, recherchez et développez le menu Amazon ECS.
2. Développez le cluster qui héberge le service souhaité.
3. Ouvrez le menu contextuel (clic droit) à partir d'un service et choisissez Désactiver l'exécution de commande.

> **Important**
>
> Cette opération démarre un nouveau déploiement de votre service et peut prendre quelques minutes. Pour plus d'informations, consultez la note au début de cette section.)

Exécution de commandes sur un conteneur

Pour exécuter des commandes sur un conteneur à l'aide de l' AWS explorateur, Amazon ECS Exec doit être activé. S'il n'est pas activé, consultez la procédure d'activation d'ECS Exec dans cette section.

1. Dans l' AWS explorateur, recherchez et développez le menu Amazon ECS.
2. Développez le cluster qui héberge le service souhaité.
3. Développez le service pour répertorier les conteneurs associés.
4. Ouvrez le menu contextuel du conteneur (cliquez avec le bouton droit de la souris) et choisissez Exécuter la commande dans le conteneur.
5. Une invite s'ouvre avec une liste des tâches en cours. Choisissez l'ARN des tâches que vous souhaitez.

> **Note**
>
> Si une seule tâche est en cours d'exécution pour ce service, elle sera sélectionnée automatiquement et cette étape sera ignorée.

6. Lorsque vous y êtes invité, tapez la commande que vous souhaitez exécuter et appuyez sur Entrée pour la traiter.

Travailler avec Amazon EventBridge

Le AWS Toolkit for Visual Studio Code (VS Code) fournit un support pour Amazon EventBridge. À l'aide du Toolkit for VS Code, vous pouvez travailler sur certains aspects EventBridge, tels que les schémas.

Rubriques

- Utilisation d'Amazon EventBridge Schemas

Utilisation d'Amazon EventBridge Schemas

Vous pouvez utiliser le AWS Toolkit for Visual Studio Code (VS Code) pour effectuer diverses opérations sur les EventBridge schémas Amazon.

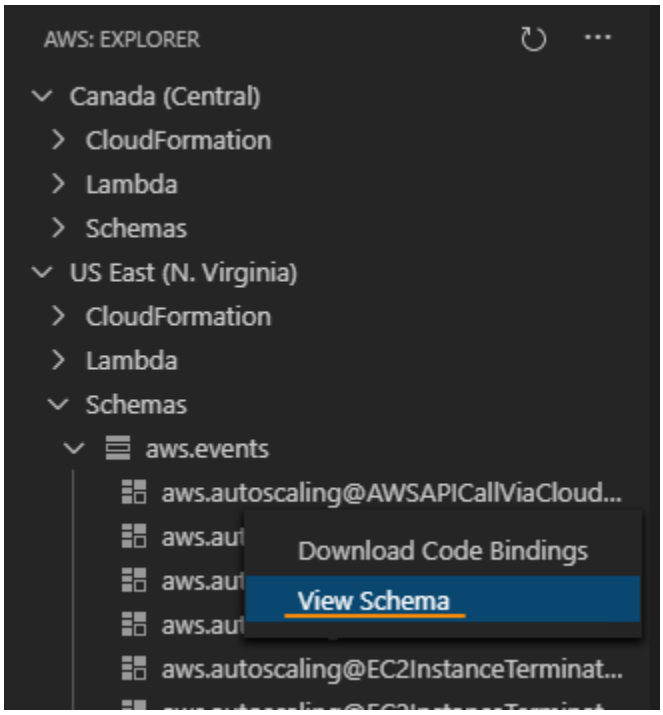
Conditions préalables

- Assurez-vous que votre système répond aux prérequis spécifiés dans Installation du Toolkit for VS Code.
- Le EventBridge schéma que vous souhaitez utiliser doit être disponible dans votre AWS compte. Si ce n'est pas le cas, créez-le ou chargez-le. Consultez Amazon EventBridge Schemas dans le guide de l' EventBridge utilisateur Amazon.

Afficher un schéma disponible

1. Dans AWS Explorer, développez Schemas (Schémas).
2. Développez le nom du registre qui contient le schéma que vous souhaitez afficher. Par exemple, de nombreux schémas fournis se AWS trouvent dans le registre aws.events.

3. Pour afficher un schéma dans l'éditeur, ouvrez le menu contextuel du schéma, puis choisissez View Schema (Afficher le schéma).



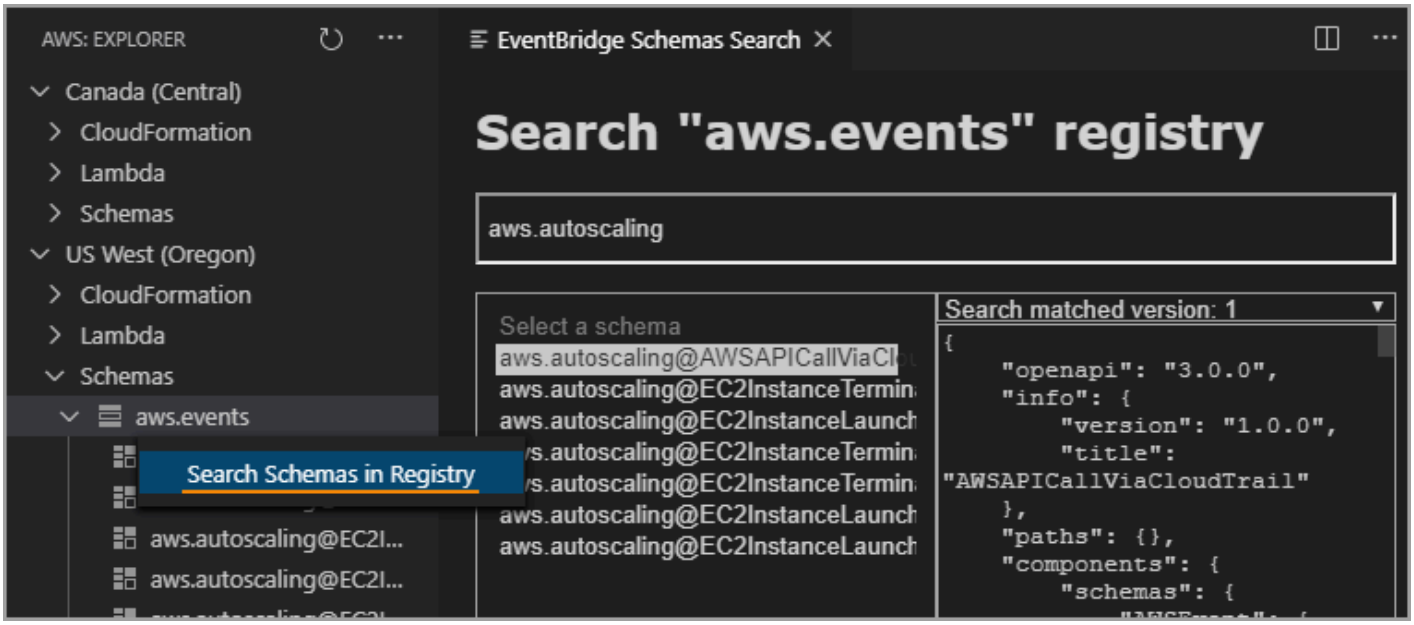
Trouver un schéma disponible

Dans AWS Explorer, effectuez une ou plusieurs des opérations suivantes :

- Commencez à saisir le titre du schéma que vous recherchez. AWS Explorer met en surbrillance les titres de schéma qui contiennent une correspondance. (Un registre doit être développé pour que vous puissiez voir les titres en surbrillance.)

- Ouvrez le menu contextuel de Schemas (Schémas), puis choisissez Search Schemas (Rechercher des schémas). Sinon, développez Schemas (Schémas), ouvrez le menu contextuel du registre qui contient le schéma recherché, puis choisissez Search Schemas in Registry (Rechercher des schémas dans le registre). Dans la boîte EventBridge de dialogue de recherche de schémas, commencez à saisir le titre du schéma que vous souhaitez rechercher. La boîte de dialogue affiche les titres de schéma qui contiennent une correspondance.

  Pour afficher le schéma dans la boîte de dialogue, sélectionnez son titre.



Générer du code pour un schéma disponible

1. Dans AWS Explorer, développez Schemas (Schémas).
2. Développez le nom du registre qui contient le schéma pour lequel vous souhaitez générer du code.
3. Cliquez avec le bouton droit sur le titre du schéma, puis choisissez Download code bindings (Télécharger les liaisons de code).
4. Dans les pages de l'Assistant qui s'affichent, choisissez les options suivantes :
   - La version du schéma.
   - Le langage de liaison de code
   - Le dossier de l'espace de travail dans lequel vous souhaitez stocker le code généré sur votre machine de développement locale

AWS Analyseur d'accès IAM

Vous pouvez exécuter des vérifications des politiques de l'analyseur d'accès IAM (AWS Identity and Access Management) sur vos politiques IAM créées dans des CloudFormation modèles, des plans Terraform et des documents de politique JSON, à l'aide de l'analyseur d'accès IAM dans le. AWS Toolkit for Visual Studio Code

Les contrôles de politique d'IAM Access Analyzer incluent la validation des politiques et des contrôles de stratégie personnalisés. La validation des politiques permet de valider vos politiques IAM conformément aux normes détaillées dans la Grammaire du langage de stratégie JSON IAM et les meilleures pratiques de AWS sécurité dans les rubriques IAM, disponibles dans le Guide de l'Gestion des identités et des accès AWSutilisateur. Les résultats de la validation de votre politique incluent des avertissements de sécurité, des erreurs, des avertissements généraux et des suggestions de politiques.

Vous pouvez également exécuter des contrôles de politique personnalisés pour les nouveaux accès, en fonction de vos normes de sécurité. Des frais sont associés à chaque vérification de politique personnalisée pour un nouvel accès. Pour obtenir des informations détaillées sur la tarification, consultez le site de tarification d'AWS IAM Access Analyzer. Pour plus d'informations sur les contrôles des politiques d'IAM Access Analyzer, consultez la rubrique Contrôles de validation des politiques du Guide de l'Gestion des identités et des accès AWSutilisateur.

Les rubriques suivantes décrivent comment utiliser les contrôles de politique d'IAM Access Analyzer dans le. AWS Toolkit for Visual Studio Code

Rubriques

- Travailler avec AWS Analyseur d'accès IAM

Travailler avec AWS Analyseur d'accès IAM

Les sections suivantes décrivent comment effectuer la validation des politiques IAM et des vérifications de politiques personnalisées dans le AWS Toolkit for Visual Studio Code. Pour plus de détails, consultez les rubriques suivantes du Guide de l' Gestion des identités et des accès AWS utilisateur : Validation des politiques d'IAM Access Analyzer et vérifications des politiques personnalisées d'IAM Access Analyzer.

Conditions préalables

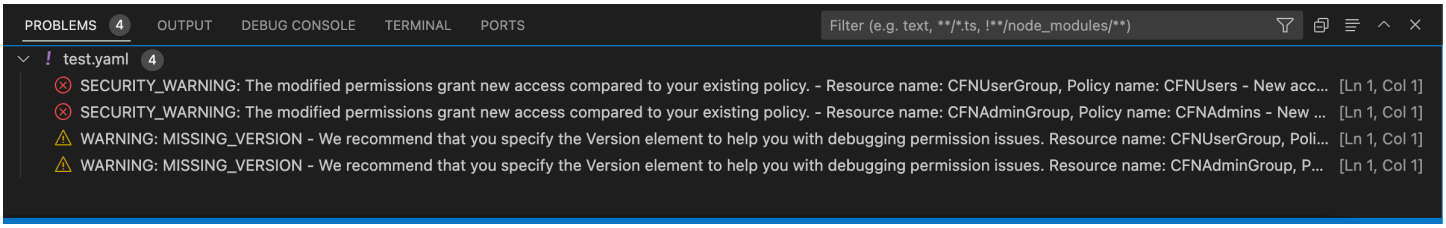
Les conditions préalables suivantes doivent être remplies avant de pouvoir utiliser les contrôles de politique d'IAM Access Analyzer à partir du Toolkit.

- Installez Python version 3.6 ou ultérieure.
- Installez le validateur de politique IAM pour CloudFormation ou le validateur de politique IAM pour Terraform qui est requis par les outils de la CLI Python et spécifié dans la fenêtre IAM Policy Checks.

- Configurez les informations d'identification de votre AWS rôle.

Vérifications de politiques IAM Access Analyzer

Vous pouvez effectuer des vérifications de politique pour les CloudFormation modèles, les plans Terraform et les documents de politique JSON à l'aide du. AWS Toolkit for Visual Studio Code Les résultats de vos vérifications sont visibles dans le panneau des problèmes de VS Code. L'image suivante montre le panneau des problèmes de VS Code.



IAM Access Analyzer propose 4 types de contrôles :

- Valider la politique
- CheckAccessNotGranted
- CheckNoNewAccess
- CheckNoPublicAccess

Les sections suivantes décrivent comment exécuter chaque type de vérification.

> **Note**
>
> Configurez les informations d'identification de votre AWS rôle avant d'exécuter tout type de vérification. Les fichiers pris en charge incluent les types de documents suivants : CloudFormation modèles, plans Terraform et documents de politique JSON
> Les références de chemin de fichier sont généralement fournies par votre administrateur ou votre équipe de sécurité. Il peut s'agir d'un chemin de fichier système ou d'un URI de compartiment Amazon S3. Pour utiliser l'URI d'un compartiment Amazon S3, votre rôle actuel doit avoir accès au compartiment Amazon S3.
> Des frais sont associés à chaque vérification de politique personnalisée. Pour plus d'informations sur la tarification des contrôles de politique personnalisés, consultez le guide de tarification d'AWS IAM Access Analyzer.

Exécution de la politique de validation

Le contrôle Validate Policy, également connu sous le nom de validation des politiques, valide votre politique par rapport à la grammaire des politiques IAM et aux AWS meilleures pratiques. Pour plus d'informations, consultez la Grammaire du langage de stratégie JSON IAM et les bonnes pratiques AWS de sécurité dans les rubriques IAM, disponibles dans le Guide de l'Gestion des identités et des accès AWSutilisateur.

1. Dans VS Code, ouvrez un fichier pris en charge contenant des politiques AWS IAM, dans l'éditeur VS Code.
2. Pour ouvrir les contrôles de politique IAM Access Analyzer, ouvrez la palette de commandes VS Code en appuyant sur**`CRTL+Shift+P`**, recherchez**`IAM Policy Checks`**, puis cliquez pour ouvrir le volet IAM Policy Checks dans l'éditeur VS Code.
3. Dans le volet IAM Policy Checks, sélectionnez votre type de document dans le menu déroulant.
4. Dans la section Valider les politiques, cliquez sur le bouton Exécuter la validation des politiques pour exécuter la vérification de validation des politiques.
5. Dans le panneau des problèmes de VS Code, passez en revue les résultats de votre vérification des politiques.
6. Mettez à jour votre politique et répétez cette procédure, en relançant la vérification de validation de la politique jusqu'à ce que les résultats de cette vérification n'affichent plus d'avertissements ou d'erreurs de sécurité.

Courir CheckAccessNotGranted

CheckAccessNotGranted est une vérification de stratégie personnalisée qui permet de vérifier que des actions IAM spécifiques ne sont pas autorisées par votre politique.

> **Note**
>
> Les références de chemin de fichier sont généralement fournies par votre administrateur ou votre équipe de sécurité. Il peut s'agir d'un chemin de fichier système ou d'un URI de compartiment Amazon S3. Pour utiliser l'URI d'un compartiment Amazon S3, votre rôle actuel doit avoir accès au compartiment Amazon S3. Au moins une action ou une ressource doit être spécifiée et le fichier doit être structuré selon l'exemple suivant :

```
            {"actions": ["action1", "action2", "action3"], "resources":
["resource1", "resource2", "resource3"]}
```

1. Dans VS Code, ouvrez un fichier pris en charge contenant des politiques AWS IAM, dans l'éditeur VS Code.
2. Pour ouvrir les contrôles de politique IAM Access Analyzer, ouvrez la palette de commandes VS Code en appuyant sur**CRTL+Shift+P**, recherchez**IAM Policy Checks**, puis cliquez pour ouvrir le volet IAM Policy Checks dans l'éditeur VS Code.
3. Dans le volet IAM Policy Checks, sélectionnez votre type de document dans le menu déroulant.
4. Dans la section Vérifications des politiques personnalisées, sélectionnez CheckAccessNotGranted.
5. Dans le champ de saisie de texte, vous pouvez saisir une liste séparée par des virgules contenant des actions et des ARN de ressources. Au moins une action ou une ressource doit être fournie.
6. Cliquez sur le bouton Exécuter une vérification de politique personnalisée.
7. Dans le panneau des problèmes de VS Code, passez en revue les résultats de votre vérification des politiques. Les vérifications de politique personnalisées renvoient un `FAIL` résultat `PASS` ou.
8. Mettez à jour votre politique et répétez cette procédure, en relançant la CheckAccessNotGranted vérification jusqu'à ce qu'elle revienne`PASS`.

Courir CheckNoNewAccess

CheckNoNewAccess est une vérification de politique personnalisée qui permet de vérifier si votre politique accorde un nouvel accès par rapport à une politique de référence.

1. Dans VS Code, ouvrez un fichier pris en charge contenant des politiques AWS IAM, dans l'éditeur VS Code.
2. Pour ouvrir les contrôles de politique IAM Access Analyzer, ouvrez la palette de commandes VS Code en appuyant sur**CRTL+Shift+P**, recherchez**IAM Policy Checks**, puis cliquez pour ouvrir le volet IAM Policy Checks dans l'éditeur VS Code.
3. Dans le volet IAM Policy Checks, sélectionnez votre type de document dans le menu déroulant.

4. Dans la section Vérifications des politiques personnalisées, sélectionnez CheckNoNewAccess.
5. Entrez un document de politique JSON de référence. Vous pouvez également fournir un chemin de fichier qui fait référence à un document de politique JSON.
6. Sélectionnez le type de politique de référence qui correspond au type de votre document de référence.
7. Cliquez sur le bouton Exécuter une vérification de politique personnalisée.
8. Dans le panneau des problèmes de VS Code, passez en revue les résultats de votre vérification des politiques. Les vérifications de politique personnalisées renvoient un `FAIL` résultat `PASS` ou.
9. Mettez à jour votre politique et répétez cette procédure, en relançant la CheckNoNewAccess vérification jusqu'à ce qu'elle revienne`PASS`.

Courir CheckNoPublicAccess

CheckNoPublicAccess est une vérification de politique personnalisée qui permet de vérifier si votre politique accorde un accès public aux types de ressources pris en charge dans votre modèle.

Pour des informations spécifiques sur les types de ressources pris en charge, consultez les référentiels cloudformation-iam-policy-validator et terraform-iam-policy-validator. https://github.com/awslabs/terraform-iam-policy-validator GitHub

1. Dans VS Code, ouvrez un fichier pris en charge contenant des politiques AWS IAM, dans l'éditeur VS Code.
2. Pour ouvrir les contrôles de politique IAM Access Analyzer, ouvrez la palette de commandes VS Code en appuyant sur**`CRTL+Shift+P`**, recherchez**`IAM Policy Checks`**, puis cliquez pour ouvrir le volet IAM Policy Checks dans l'éditeur VS Code.
3. Dans le volet IAM Policy Checks, sélectionnez votre type de document dans le menu déroulant.
4. Dans la section Vérifications des politiques personnalisées, sélectionnez CheckNoPublicAccess.
5. Cliquez sur le bouton Exécuter une vérification de politique personnalisée.
6. Dans le panneau des problèmes de VS Code, passez en revue les résultats de votre vérification des politiques. Les vérifications de politique personnalisées renvoient un `FAIL` résultat `PASS` ou.
7. Mettez à jour votre politique et répétez cette procédure, en relançant la CheckNoNewAccess vérification jusqu'à ce qu'elle revienne`PASS`.

Travailler avec AWS IoT in AWS Toolkit for Visual Studio Code

AWS IoT in vous AWS Toolkit for Visual Studio Code permet d'interagir avec le AWS IoT service, tout en minimisant les interruptions de votre flux de travail dans VS Code. Ce guide de l'utilisateur est destiné à vous aider à commencer à utiliser les fonctionnalités de AWS IoT service disponibles dans le AWS Toolkit for Visual Studio Code. Pour plus d'informations sur le AWS IoT service, consultez le guide du développeur Qu'est-ce que c'est AWS IoT ?

AWS IoT prérequis

Pour commencer AWS IoT à utiliser Toolkit for VS Code, assurez-vous que votre AWS compte et VS Code répondent aux exigences de ces guides :

- Pour connaître les exigences relatives au AWS compte et les autorisations AWS utilisateur spécifiques au AWS IoT service, consultez le guide du développeur Getting Started with AWS IoT Core.
- Pour connaître les exigences spécifiques du Toolkit for VS Code, consultez le guide d'utilisation de Setting up the Toolkit for VS Code.

AWS IoT Choses

AWS IoT connecte les appareils aux services et ressources AWS cloud. Vous pouvez connecter vos appareils à AWS IoT l'aide d'objets appelés objets. Un objet est une représentation d'un appareil spécifique ou d'une entité logique. Il peut s'agir d'un appareil physique ou d'un capteur (par exemple, une ampoule ou un interrupteur sur un mur). Pour plus d'informations à AWS IoT ce sujet, consultez le guide du développeur Managing devices with AWS IoT.

Gérer AWS IoT les choses

Le Toolkit for VS Code possède plusieurs fonctionnalités qui rendent la gestion AWS IoT de vos objets plus efficace. Voici comment vous pouvez utiliser le kit d'outils VS Code pour gérer vos AWS IoT affaires :

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

Pour créer un objet

1. Dans l' AWS explorateur, élargissez le titre du service IoT et sélectionnez des objets en maintenant le bouton droit de la souris en cliquant sur le bouton droit de la souris.
2. Choisissez Create Thing dans le menu contextuel pour ouvrir une boîte de dialogue.
3. Suivez les instructions en saisissant le nom de votre objet IoT dans le champ Nom de l'objet.
4. Lorsque cette opération est terminée, une icône d'objet suivie du nom que vous avez spécifié sera visible dans la section Objet.

Pour attacher un certificat à un objet

1. Dans l' AWS explorateur, développez la section des services IoT.
2. Dans la sous-section Objets, localisez l'objet auquel vous joignez le certificat.
3. Sélectionnez l'objet avec le bouton droit de la souris et choisissez Joindre un certificat dans le menu contextuel pour ouvrir un sélecteur de saisie contenant la liste de vos certificats.
4. Dans la liste, choisissez l'ID du certificat qui correspond au certificat que vous souhaitez associer à votre objet.
5. Lorsque cette opération est terminée, votre certificat est accessible dans l' AWS explorateur, en tant qu'élément de l'objet auquel vous l'avez joint.

Pour détacher un certificat d'un objet

1. Depuis l' AWS explorateur, élargissez la section des services IoT
2. Dans la sous-section Things (Objets), localisez l'objet dont vous voulez détacher un certificat.
3. Sélectionnez l'objet en maintenant la touche contextuelle (clic droit) et choisissez Détacher le certificat dans le menu contextuel.
4. Une fois cette opération terminée, le certificat détaché ne s'affichera plus sous cet élément dans l' AWS explorateur, mais il sera toujours accessible depuis la sous-section Certificats.

Pour supprimer un objet

1. Dans l' AWS explorateur, développez la section des services IoT.

2. Dans la sous-section Objets, recherchez l'élément que vous souhaitez supprimer.
3. Sélectionnez l'objet avec le bouton droit de la souris (cliquez avec le bouton droit de la souris) et choisissez Supprimer l'objet dans le menu contextuel pour le supprimer.
4. Une fois cette opération terminée, l'objet supprimé ne sera plus disponible dans la sous-section Objets.

> **Note**
>
> Remarque : Vous ne pouvez supprimer qu'un objet auquel aucun certificat n'est associé.

AWS IoT certificats

Les certificats constituent un moyen courant de créer une connexion sécurisée entre vos AWS IoT services et appareils. Les certificats X.509 sont des certificats numériques qui font appel à la norme d'infrastructure de clé publique X.509 pour associer une clé publique à une identité contenue dans un certificat. Pour plus d'informations sur AWS IoT les certificats, consultez le guide du développeur Authentication (IoT).

Gestion de certificats

La boîte à outils VS Code vous propose différentes manières de gérer vos AWS IoT certificats, directement depuis l' AWS explorateur.

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

Pour créer un AWS IoT certificat

Un certificat X.509 peut être utilisé pour se connecter à votre instance de AWS IoT.

1. Dans l' AWS explorateur, développez la section des services IoT et sélectionnez les certificats avec le bouton droit de la souris (clic droit).
2. Choisissez Créer un certificat dans le menu contextuel pour ouvrir une boîte de dialogue.

3. Sélectionnez un répertoire dans votre système de fichiers local pour enregistrer votre paire de clés RSA et votre certificat X.509.

> **Note**
>
> - Les noms de fichiers par défaut contiennent l'ID du certificat comme préfixe.
> - Seul le certificat X.509 est enregistré dans votre AWS compte, via le AWS IoT service.
> - Votre paire de clés RSA ne peut être émise qu'une seule fois, enregistrez-la dans un emplacement sécurisé de votre système de fichiers lorsque vous y êtes invité.
> - Si le certificat ou la paire de clés ne peuvent pas être enregistrés dans votre système de fichiers pour le moment, le AWS Toolkit supprime le certificat de votre AWS compte.

Pour modifier l'état d'un certificat

Le statut d'un certificat individuel est affiché à côté de son identifiant dans l' AWS explorateur et peut être défini comme suit : actif, inactif ou révoqué.

> **Note**
>
> - Votre certificat doit avoir un statut actif avant de pouvoir l'utiliser pour connecter votre appareil à votre AWS IoT service.
> - Un certificat inactif peut être activé, qu'il ait été précédemment désactivé ou qu'il soit inactif par défaut.
> - Un certificat qui a été révoqué ne peut pas être réactivé.

1. Dans l' AWS explorateur, développez la section des services IoT.
2. Dans la sous-section Certificats, recherchez le certificat que vous souhaitez modifier.
3. Sélectionnez le certificat avec le bouton droit de la souris pour ouvrir un menu contextuel qui affiche les options de changement de statut disponibles pour ce certificat.

- Si un certificat présente l'état inactif, choisissez activate (activer) pour changer l'état en actif.

- Si un certificat présente l'état actif, choisissez deactivate (désactiver) pour changer l'état en inactif.
- Si un certificat présente l'état actif ou inactif, choisissez revoke (révoquer) pour changer l'état en révoqué.

> **Note**
>
> Chacune de ces actions de modification de statut est également disponible si vous sélectionnez un certificat attaché à un objet alors qu'il est affiché dans la sous-section Objets.

Pour attacher une politique IoT à un certificat

1. Dans l' AWS explorateur, développez la section des services IoT.
2. Dans la sous-section Certificats, recherchez le certificat que vous souhaitez modifier.
3. Sélectionnez le certificat avec le bouton droit de la souris et choisissez Attacher une politique dans le menu contextuel pour ouvrir un sélecteur de saisie contenant la liste des politiques disponibles.
4. Choisissez la politique que vous souhaitez associer au certificat.
5. Une fois cette opération terminée, la politique que vous avez sélectionnée sera ajoutée au certificat en tant qu'élément de sous-menu.

Pour détacher une politique IoT d'un certificat

1. Dans l' AWS explorateur, développez la section des services IoT.
2. Dans la sous-section Certificats, recherchez le certificat que vous souhaitez modifier.
3. Développez le certificat et localisez la politique que vous souhaitez détacher.
4. Sélectionnez la politique avec le bouton droit de la souris (clic droit) et choisissez Détacher dans le menu contextuel.
5. Une fois cette opération terminée, la politique ne sera plus accessible depuis votre certificat, mais elle sera disponible depuis la sous-section Politique.

Pour supprimer un certificat

1. Dans l' AWS explorateur, élargissez la rubrique du service IoT.
2. Dans la sous-section Certificats, recherchez le certificat que vous souhaitez supprimer.
3. Sélectionnez le certificat dans le contexte (cliquez avec le bouton droit de la souris) et choisissez Supprimer le certificat dans le menu contextuel.

> **Note**
>
> Vous ne pouvez pas supprimer un certificat s'il est attaché à un objet ou s'il a un état actif. Vous pouvez supprimer un certificat auquel sont associées des politiques.

AWS IoT politiques

AWS IoT Les politiques de base sont définies par le biais de documents JSON, chacun contenant une ou plusieurs déclarations de politique. Les politiques définissent comment AWS IoT AWS, et votre appareil peut interagir les uns avec les autres. Pour plus d'informations sur la création d'un document de politique, consultez le guide du développeur IoT Polices.

> **Note**
>
> Les politiques nommées sont versionnées afin que vous puissiez les annuler. Dans The AWS Explorer, vos politiques IoT sont répertoriées dans la sous-section Politiques, dans le service IoT. Vous pouvez afficher les versions des politiques en développant une politique. La version par défaut est indiquée par un astérisque.

Gestion des politiques

Le Toolkit for VS Code propose plusieurs méthodes pour gérer vos politiques AWS IoT de service. Voici comment vous pouvez gérer ou modifier vos politiques directement depuis l' AWS explorateur dans VS Code :

- Create a policy
- Upload a new policy version
- Edit a policy version

- Change the policy version defualt
- Change the policy version defualt

Pour créer une AWS IoT politique

> **Note**
>
> Vous pouvez créer une nouvelle politique à partir de l' AWS explorateur, mais le document JSON qui définit la politique doit déjà exister dans votre système de fichiers.

1. Dans l' AWS explorateur, développez la section des services IoT.
2. Sélectionnez la sous-section Politiques tout en maintenant le bouton droit de la souris enfoncée et choisissez Créer une politique à partir d'un document pour ouvrir le champ de saisie Nom de la politique.
3. Saisissez un nom et suivez les instructions pour ouvrir une boîte de dialogue vous demandant de sélectionner un document JSON dans votre système de fichiers.
4. Choisissez le fichier JSON qui contient vos définitions de politique, la politique sera disponible dans l' AWS explorateur lorsque cela sera terminé.

Pour télécharger une nouvelle version AWS IoT de la politique

Une nouvelle version d'une politique peut être créée en téléchargeant un document JSON dans la politique.

> **Note**
>
> Le nouveau document JSON doit être présent sur votre système de fichiers pour créer une nouvelle version à l'aide de l' AWS explorateur.

1. Dans l' AWS explorateur, développez la section des services IoT.
2. Développez la sous-section Politiques pour afficher vos AWS IoT politiques
3. Sélectionnez dans le contexte (clic droit) la politique que vous souhaitez mettre à jour et choisissez Créer une nouvelle version à partir du document.

4. Lorsque la boîte de dialogue s'ouvre, choisissez le fichier JSON qui contient les mises à jour de vos définitions de politiques.
5. La nouvelle version sera accessible depuis votre politique dans l' AWS explorateur.

Pour modifier une version d'une AWS IoT politique

Un document de politique peut être ouvert et modifié à l'aide de VS Code. Lorsque vous avez terminé de modifier le document, vous pouvez l'enregistrer dans votre système de fichiers. Ensuite, vous pouvez le télécharger sur votre AWS IoT service depuis l' AWS explorateur.

1. Dans l' AWS explorateur, développez la section des services IoT.
2. Développez la sous-section Politiques et localisez la politique que vous souhaitez mettre à jour. Créez une politique à partir d'un document pour ouvrir le champ de saisie Nom de la politique.
3. Développez la politique que vous souhaitez mettre à jour, puis sélectionnez la version de la stratégie que vous souhaitez modifier en maintenant le bouton droit de la souris.
4. Choisissez Afficher dans le menu contextuel pour ouvrir la version de la politique dans VS Code
5. Lorsque le document de politique est ouvert, apportez et enregistrez les modifications souhaitées.

> **Note**
>
> À ce stade, les modifications que vous avez apportées à la politique ne sont enregistrées que dans votre système de fichiers local. Pour mettre à jour la version et en effectuer le suivi avec l' AWS explorateur, répétez les étapes décrites dans la Upload a new policy version procédure.

Pour sélectionner une nouvelle version de la politique par défaut

1. Dans l' AWS explorateur, développez la section des services IoT.
2. Développez la sous-section Policies (Politiques) et localisez la politique que vous voulez mettre à jour.
3. Développez la politique que vous souhaitez mettre à jour, puis sélectionnez la version de la stratégie que vous souhaitez définir en maintenant la touche contextuelle (clic droit), puis choisissez Définir par défaut.

4. Lorsque cette opération est terminée, la nouvelle version par défaut que vous avez sélectionnée sera placée à côté d'une étoile.

Pour supprimer des stratégies

> **Note**
>
> Avant de pouvoir supprimer une politique ou une version de politique, certaines conditions doivent être remplies.
>
> - Vous ne pouvez pas supprimer une politique si elle est attachée à un certificat.
> - Vous ne pouvez pas supprimer une politique si elle contient des versions autres que celles par défaut.
> - Vous ne pouvez pas supprimer la version par défaut d'une politique à moins qu'une nouvelle version par défaut ne soit sélectionnée ou que la politique soit entièrement supprimée.
> - Avant de pouvoir supprimer une politique dans son intégralité, toutes les versions autres que celles par défaut de cette politique doivent d'abord être supprimées.

1. Dans l' AWS explorateur, développez la section des services IoT.
2. Développez la sous-section Policies (Politiques) et localisez la politique que vous voulez mettre à jour.
3. Développez la politique que vous souhaitez mettre à jour, puis sélectionnez la version de la politique que vous souhaitez supprimer en maintenant le bouton droit de la souris et choisissez Supprimer.
4. Lorsqu'une version est supprimée, elle n'est plus visible dans l'explorateur.
5. Lorsque la seule version restante pour une politique est la version par défaut, vous pouvez sélectionner la stratégie parent dans le contexte (clic droit) et choisir Supprimer pour la supprimer.

AWS Lambda Fonctions

AWS Toolkit for Visual Studio Code Il fournit un support complet pour les AWS Lambda fonctions, vous permettant de créer, de tester et de déployer directement à partir de VS Code.

Lambda est un service de calcul entièrement géré et piloté par les événements qui exécute automatiquement votre code en réponse aux événements provenant de plus de 200 services AWS et applications (software-as-a-serviceSaaS). Pour obtenir des informations détaillées sur le AWS Lambda service, consultez le guide du AWS Lambdadéveloppeur.

Les rubriques suivantes décrivent comment travailler avec AWS Lambda . AWS Toolkit for Visual Studio Code

Rubriques

- Travailler avec AWS Lambda Fonctions
- AWS Lambda console vers IDE
- AWS Lambda avec LocalStack support
- AWS Lambda débogage à distance

Travailler avec AWS Lambda Fonctions

AWS Toolkit for Visual Studio Code Cela vous permet de travailler avec vos AWS Lambda fonctions dans votre environnement VS Code local. Avec le AWS Toolkit, vous pouvez créer, modifier, tester, déboguer et déployer vos fonctions Lambda sans avoir à quitter l'IDE. Pour obtenir des informations détaillées sur le AWS Lambda service, consultez le Guide du AWS Lambda développeur.

Les sections suivantes décrivent comment commencer à utiliser les fonctions Lambda du AWS Toolkit for Visual Studio Code.

> **Note**
>
> Si vous avez déjà créé des fonctions Lambda à l'aide de Console de gestion AWS, vous pouvez les invoquer depuis le Toolkit. En outre, vous pouvez ouvrir vos fonctions Lambda dans VS Code à partir de AWS Lambda console, pour plus d'informations, voir la AWS Lambda console vers IDE rubrique de ce guide de l'utilisateur. Pour créer une nouvelle fonction Lambda dans VS Code, suivez les étapes décrites dans la rubrique Création d'une nouvelle application sans serveur (locale) de ce guide de l'utilisateur.

Conditions préalables

Les conditions suivantes doivent être remplies pour utiliser le AWS Lambda service de la boîte à AWS outils.

- La dernière version de AWS Toolkit for Visual Studio Code est installée et configurée avec vos AWS informations d'identification.
- Vos autorisations et politiques gérées Gestion des identités et des accès AWS (IAM) sont configurées pour fonctionner avec le AWS Lambda service. Pour obtenir des informations détaillées sur la façon de configurer vos autorisations et de créer une politique AWS gérée compatible, consultez Gestion des identités et des accès AWS la AWS Lambda rubrique correspondante du Guide du AWS Lambda développeur.
- Vous disposez de AWS Lambda fonctions existantes ou savez comment en créer une. Pour obtenir des instructions sur la création d'une fonction Lambda, consultez la rubrique Créer votre première fonction Lambda dans le Guide du AWS Lambda développeur.

Invocation d'une fonction Lambda

Pour appeler une fonction Lambda depuis votre AWS compte dans VS Code, procédez comme suit.

1. À partir de AWS Toolkit for Visual Studio Code, développez l' AWS explorateur.
2. Depuis l' AWS explorateur, développez Lambda pour afficher vos ressources Lambda.
3. Ouvrez le menu contextuel (cliquez avec le bouton droit de la souris) sur la fonction Lambda que vous souhaitez invoquer, puis choisissez Invoke in the cloud ou choisissez l'icône Invoke in the cloud pour ouvrir le menu de configuration de l'invocation à distance dans VS Code.
4. Dans le menu de configuration de l'appel à distance, spécifiez vos paramètres de charge utile et ajoutez toute information supplémentaire requise pour l'événement.

   > **Note**
   >
   > Le premier processus d'invocation peut commencer à s'exécuter dès que vous choisissez Invoke in the cloud dans l' AWS explorateur. La sortie est affichée dans l'onglet OUTPUT du terminal VS Code.

5. Cliquez sur le bouton Remote Invoke pour appeler votre fonction. La sortie est affichée dans l'onglet OUTPUT du terminal VS Code.

Suppression d'une fonction Lambda

Pour supprimer une fonction Lambda, procédez comme suit.

> **Warning**
>
> N'utilisez pas cette procédure pour supprimer les fonctions Lambda qui sont associées à CloudFormation. Ces fonctions doivent être supprimées de votre CloudFormation stack.

1. À partir de AWS Toolkit for Visual Studio Code, développez l' AWS explorateur.
2. Depuis l' AWS explorateur, développez Lambda pour afficher vos ressources Lambda.
3. Right-click la fonction Lambda que vous souhaitez supprimer, puis choisissez Supprimer.
4. Lorsque vous y êtes invité, confirmez que vous souhaitez supprimer votre fonction.

Une fois la fonction supprimée, elle n'est plus répertoriée dans l' AWS explorateur.

Téléchargement d'une fonction Lambda

Vous pouvez télécharger du code depuis une fonction Lambda distante dans votre espace de travail VS Code pour le modifier et le déboguer.

> **Note**
>
> Pour télécharger votre fonction Lambda, vous devez travailler dans un espace de travail VS Code avec un dossier accessible et le AWS Toolkit ne prend en charge cette fonctionnalité qu'avec les fonctions Lambda Node.js et les environnements d'exécution Python.

1. À partir de AWS Toolkit for Visual Studio Code, développez l' AWS explorateur.
2. Depuis l' AWS explorateur, développez Lambda pour afficher vos ressources Lambda.
3. Right-click la fonction Lambda que vous souhaitez télécharger, puis choisissez Télécharger.
4. Votre fonction Lambda s'ouvre dans l'éditeur VS Code et s'affiche dans l' AWS explorateur une fois le téléchargement terminé. Le AWS Toolkit crée également une configuration de lancement dans le panneau d'exécution de VS Code vous permettant d'exécuter et de déboguer la fonction Lambda localement avec. AWS Serverless Application Model Pour plus d'informations sur l'utilisation AWS SAM, consultezthe section called “Exécution et débogage d'une application sans serveur à partir d'un modèle (local)”.

Déploiement de mises à jour pour les nouvelles fonctions Lambda

Vous pouvez déployer des mises à jour des nouvelles fonctions Lambda à partir d'un emplacement temporaire non spécifié sur votre machine locale.

> **Note**
>
> Lorsque des modifications non déployées sont apportées à vos fichiers lambda, vous êtes averti par l'icône M située à côté des fichiers modifiés dans l'éditeur VS Code et dans l' AWS explorateur.

Déploiement depuis l'éditeur VS Code

1. Ouvrez un fichier depuis votre fonction Lambda dans l'éditeur VS Code, puis modifiez le fichier.
2. Enregistrez manuellement depuis le menu principal de VS Code ou en appuyant sur **`option+s`** (Mac) **`ctrl+s`** (Windows).
3. VS Code vous invite automatiquement à déployer vos modifications dans le cloud. Cliquez sur le bouton Déployer pour confirmer le déploiement.
4. VS Code vous informe de l'état de votre déploiement et vous avertit lorsque le processus est terminé.

Déploiement depuis AWS Explorer

1. Ouvrez un fichier depuis votre fonction Lambda dans l'éditeur VS Code, puis modifiez le fichier.
2. À partir de la AWS boîte à outils, développez l' AWS explorateur.
3. Dans l' AWS explorateur, développez la AWS région à l'aide de la fonction Lambda pour laquelle vous souhaitez déployer des modifications.
4. Depuis la AWS région, développez Lambda et naviguez dans la fonction pour laquelle vous souhaitez déployer des modifications.
5. Dans le menu rapide situé à côté de votre fonction, choisissez l'icône Enregistrer et déployer votre code.
6. VS Code vous informe de l'état de votre déploiement et vous avertit lorsque le processus est terminé.

Téléchargement de mises à jour pour les fonctions Lambda existantes

Les procédures suivantes décrivent comment télécharger les modifications locales apportées à vos fonctions Lambda existantes. Cette fonctionnalité prend en charge les téléchargements avec n'importe quel environnement d'exécution compatible avec Lambda.

> **Warning**
>
> Avant de télécharger votre fonction lambda, tenez compte des points suivants :
>
> - La mise à jour du code de cette manière n'utilise pas l' AWS SAM interface de ligne de commande pour le déploiement et ne crée pas de CloudFormation pile
> - La AWS boîte à outils ne valide pas le code. Validez votre code et testez vos fonctions avant de télécharger les modifications dans le cloud.

Téléchargement d'une archive Zip

1. À partir de AWS Toolkit for Visual Studio Code, développez l' AWS explorateur.
2. Depuis l' AWS explorateur, développez Lambda pour afficher vos ressources Lambda.
3. Right-click la fonction Lambda vers laquelle vous souhaitez télécharger vos modifications, puis choisissez Upload Lambda... pour ouvrir le menu Sélectionner le type de téléchargement.
4. Choisissez ZIP Archive pour le localiser `ZIP Archive` dans votre répertoire local.
5. Lorsque vous y êtes invité, confirmez le téléchargement pour démarrer le téléchargement de l'élément sélectionné`ZIP Archive`.
6. L'état de votre téléchargement est affiché dans VS Code et vous êtes averti lorsque le processus de téléchargement est terminé.

Charger un répertoire sans le créer

1. À partir de AWS Toolkit for Visual Studio Code, développez l' AWS explorateur.
2. Depuis l' AWS explorateur, développez Lambda pour afficher vos ressources Lambda.
3. Right-click la fonction Lambda vers laquelle vous souhaitez télécharger vos modifications, puis choisissez Upload Lambda... pour ouvrir le menu Sélectionner le type de téléchargement.
4. Choisissez Répertoire pour accéder à l'écran Créer un répertoire.

5. Dans l'écran Créer un répertoire, choisissez Non pour choisir un répertoire local à télécharger.
6. Lorsque vous y êtes invité, confirmez le téléchargement pour télécharger le répertoire sélectionné.
7. L'état de votre téléchargement est affiché dans VS Code et vous êtes averti lorsque le processus de téléchargement est terminé.

Téléchargement d'un répertoire avec une compilation

> **Note**
>
> Tenez compte des points suivants :
>
> - Cette procédure nécessite l' AWS Serverless Application Model interface de ligne de commande.
> - Le AWS Toolkit vous indique qu'aucun gestionnaire correspondant ne peut être détecté avant le téléchargement.
> - Pour modifier le gestionnaire associé à votre fonction Lambda, utilisez le AWS Lambda console ou le. AWS Command Line Interface

1. À partir de AWS Toolkit for Visual Studio Code, développez l' AWS explorateur.
2. Depuis l' AWS explorateur, développez Lambda pour afficher vos ressources Lambda.
3. Right-click la fonction Lambda vers laquelle vous souhaitez télécharger vos modifications, puis choisissez Upload Lambda... pour ouvrir le menu Sélectionner le type de téléchargement.
4. Choisissez Répertoire pour accéder à l'écran Créer un répertoire.
5. Dans l'écran Créer un répertoire, choisissez Oui, puis sélectionnez un répertoire local à charger.
6. Lorsque vous y êtes invité, confirmez le téléchargement pour commencer à créer et à charger le répertoire sélectionné.
7. L'état de votre téléchargement est affiché dans VS Code et vous êtes averti lorsque le processus de téléchargement est terminé.

Conversion de votre fonction Lambda en AWS SAM project

Pour convertir votre fonction Lambda en AWS SAM pile, procédez comme suit.

> **Warning**
>
> Actuellement, seul un sous-ensemble de ressources est pris en charge lors de la conversion d'une fonction Lambda en projet. AWS SAM Pour localiser les ressources manquantes après une conversion, consultez la console Lambda et ajoutez-les manuellement à votre AWS SAM modèle. Pour plus de détails sur les ressources prises en charge et non prises en charge, consultez la rubrique relative à la prise en charge des types de ressources dans le Guide du AWS CloudFormation développeur.

1. À partir de la AWS boîte à outils, développez l' AWS explorateur.
2. Dans l' AWS explorateur, développez la AWS région à l'aide de la fonction Lambda que vous souhaitez convertir en AWS SAM projet.
3. Dans la AWS région, développez Lambda et naviguez dans la fonction que vous souhaitez convertir en AWS SAM pile.
4. Dans le menu rapide situé à côté de votre fonction Lambda, choisissez l'icône Convertir en application SAM pour parcourir votre système de fichiers local et spécifier l'emplacement de votre nouveau AWS SAM projet.
5. Après avoir spécifié un emplacement, le AWS Toolkit commence à convertir votre fonction Lambda en AWS SAM projet. VS Code fournit des mises à jour sur l'état du processus.

   > **Note**
   >
   > Ce processus peut prendre quelques minutes.

6. Lorsque VS Code vous y invite, entrez un nom de pile, puis appuyez sur la **`Enter`** touche pour continuer.
7. VS Code continue de vous informer de l'état de votre projet, puis vous informe lorsque le processus est terminé et ouvre votre nouveau AWS SAM projet en tant qu'espace de travail VS Code.

AWS Lambda console vers IDE

La fonctionnalité AWS Lambda console vers l'IDE vous permet de télécharger vos AWS Lambda fonctions depuis le AWS Lambda console dans VS Code. L'utilisation de vos fonctions Lambda dans

VS Code vous donne accès à d'autres options de développement local telles que AWS Serverless Application Model (AWS SAM) et le. Cloud Development Kit AWS (AWS CDK)

Pour plus d'informations à ce sujet AWS Lambda, consultez le Guide du AWS Lambda développeur. Pour commencer à utiliser votre fonction Lambda dans le AWS Toolkit, consultez la rubrique Utilisation des AWS Lambda fonctions de ce guide de l'utilisateur. Les sections suivantes décrivent comment déplacer votre flux de travail de la console Lambda vers VS Code. Pour des informations détaillées sur le transfert de vos fonctions Lambda de la console Lambda vers VS Code, notamment sur la façon de commencer à utiliser la console Lambda, consultez la rubrique Développement de fonctions Lambda localement avec VS Code dans le Guide du développeur. AWS Lambda

Déplacement de la console au développement local

Pour ouvrir une fonction Lambda depuis la console Lambda dans VS Code, procédez comme suit :

1. Dans votre navigateur Web, ouvrez la console Lambda.
2. Dans la console Lambda, choisissez la fonction que vous souhaitez ouvrir dans VS Code.
3. Dans la vue des fonctions, accédez à l'onglet Source du code.
4. Dans l'onglet Source du code, choisissez Ouvrir dans VS Code.

Utilisation de votre fonction Lambda dans VS Code

Lorsque votre fonction Lambda s'ouvre dans VS Code via la console Lambda :

- VS Code se lance automatiquement sur votre machine locale.
- Votre fonction Lambda s'ouvre en tant qu'espace de travail VS Code.
- Votre Lambda `handler file` s'ouvre dans l'éditeur VS Code.

> **Note**
>
> Si aucun fichier n'est correctement configuré `handler file` dans l'espace de travail, aucun fichier ne s'ouvre dans l'éditeur VS Code.

L'ouverture de votre fonction Lambda dans VS Code via la console Lambda vous permet d'accéder à toutes les fonctionnalités existantes du AWS Toolkit Lambda, notamment la possibilité de modifier le code de fonction avec une prise en charge linguistique complète, les tests locaux, le débogage à

distance, l'assistance au déploiement et la gestion des dépendances. Pour plus d'informations sur les fonctionnalités Lambda prises en charge dans le AWS Toolkit, consultez la table des matières des AWS Lambda services dans ce guide de l'utilisateur.

AWS Lambda avec LocalStack support

Créez, testez et déboguez vos applications sans serveur grâce à l' LocalStack assistance fournie dans le. AWS Toolkit for Visual Studio Code LocalStack est un émulateur AWS Cloud qui permet de tester localement des applications sans serveur.

Pour plus d'informations à ce sujet AWS Lambda, consultez le Guide du AWS Lambda développeur. Pour en savoir plus LocalStack, visitez leur site Web LocalStack.

Conditions préalables

Les conditions préalables à l'utilisation de VS Code sont LocalStack les suivantes.

> **Note**
>
> La LocalStack CLI est installée pendant le processus d'installation, mais si vous préférez une autre version de la LocalStack CLI, la version minimale requise est 4.8.0.

- Un compte d'application LocalStack Web est requis pour accéder à toutes les fonctionnalités disponibles pour les LocalStack niveaux gratuits et payants. LocalStack L'édition communautaire est disponible sans compte.
- Docker doit fonctionner avec LocalStack dans VS Code. Pour plus d'informations sur la LocalStack configuration requise pour Docker, consultez la rubrique LocalStack Docker Images dans la LocalStack documentation.
- Recommandé : le AWS Command Line Interface (AWS CLI) vous aide à utiliser les services de votre environnement cloud simulé.

Installation LocalStack

Pour installer des versions hiérarchisées LocalStack gratuites et payantes, procédez comme suit.

> **Note**
>
> Pour obtenir des instructions sur LocalStack la configuration de l'édition Community, consultez le contenu de la LocalStack communauté dans la LocalStack section Configuration de cette rubrique.

1. À partir de la AWS boîte à outils, développez l'explorateur APPLICATION BUILDER.
2. Cliquez sur le bouton Ouvrir la procédure pas à pas pour ouvrir l'onglet Commencer à créer votre application dans l'éditeur VS Code.
3. Dans la procédure pas à pas, choisissez Installer LocalStack pour démarrer le processus LocalStack d'installation dans VS Code.

Con LocalStackfiguration

Après avoir installé l' LocalStack extension pour VS Code, l'un des indicateurs suivants peut s'afficher lorsque la configuration est requise :

- Dans la barre d'état VS Code, située dans le coin inférieur gauche de l'IDE par défaut, l' LocalStack état est rouge.
- VS Code vous invite à configurer. LocalStack

Il existe deux types de configuration et de configuration LocalStack, selon la version que LocalStack vous utilisez. Les sections à onglets suivantes décrivent chaque processus de LocalStack configuration.

> **Note**
>
> LocalStack des jetons d'authentification sont requis pour les versions gratuites et payantes de LocalStack. Pour des informations spécifiques sur la LocalStack tarification, consultez leur guide de tarification Choisissez votre plan.

LocalStack niveaux gratuits et payants

Il existe deux méthodes de configuration LocalStack.

- Dans l'invite de configuration LocalStack de VS Code pour démarrer, cliquez sur le bouton Configuration.
- Dans la barre d'état de VS Code, choisissez l'icône d' LocalStack état pour ouvrir l'invite Configuration LocalStack pour démarrer, puis cliquez sur le bouton Configuration.

Lors de la configuration, le système suit les étapes suivantes :

1. Installe la LocalStack CLI.
2. Vérifie si vous avez un LocalStack compte.
3. Si vous possédez un LocalStack compte, le système vous guide tout au long du processus d'authentification dans votre navigateur Web par défaut. De même, si vous n'avez pas de LocalStack compte, le système vous guide tout au long de la configuration du compte avant le processus d'authentification.

Une fois configuré, le LocalStack statut LocalStack est mis à jour dans la barre d'état VS Code.

> **Note**
>
> Si vous n'avez pas créé de AWS profil pour LocalStack, un nouveau profil est automatiquement créé pour vous dans le cadre du processus de LocalStack configuration.

LocalStack Communauté

L'édition Community de LocalStack est gratuite et ne nécessite pas de création de compte. Elle fonctionne à partir d'une image Docker qui ne nécessite pas de licence. Pour plus de détails sur LocalStack Community Edition, consultez la documentation relative aux images de la LocalStack communauté. Les sections suivantes décrivent les prérequis et la configuration de base requise pour utiliser l'édition LocalStack communautaire dans VS Code.

Lancer une nouvelle instance

Pour lancer une nouvelle instance de LocalStack Community, procédez comme suit.

> **Note**
>
> L'exemple suivant démarre une instance de conteneur LocalStack sur le port 4566. Si vous spécifiez des valeurs de port différentes, vous devez mettre à jour la valeur de port spécifiée dans la procédure située dans la section Configuration de la AWS boîte à outils AWS CLI et.

1. Depuis VS Code, ouvrez le terminal VS Code en appuyant sur**ctrl + `(backtick)**.
2. Entrez les informations suivantes dans le terminal.

   Mac :

   ```
   docker run -d --name localstack_main \
   >> -p 4566:4566 \
   >> -v /var/run/docker.sock:/var/run/docker.sock \
   >> localstack/localstack
   ```

   Windows:

   ```
   docker run -d --name localstack_main `
   >> -p 4566:4566 `
   >> -v /var/run/docker.sock:/var/run/docker.sock `
   >> localstack/localstack
   ```
3. Le terminal est mis à jour avec l'état de votre instance Docker lorsque le processus est terminé.

Cette instance conteneurisée de vous LocalStack donne accès aux AWS services que vous avez spécifiés lors du processus de téléchargement.

Configuration de la CLI pour LocalStack et Docker.

Pour configurer la AWS boîte à outils AWS CLI et l' LocalStack utiliser dans Docker, créez un nouveau profil en suivant les étapes suivantes :

1. Depuis VS Code, ouvrez le terminal VS Code en appuyant sur**ctrl + `(backtick)**.
2. Entrez les informations suivantes dans le terminal.

   ```
   ~/.aws/credentials
   [localstack]
   ```

```
aws_access_key_id = test
aws_secret_access_key = test
~/.aws/config
[profile localstack]
region = us-east-1
output = json
endpoint_url = http://localhost:4566 [default localstack endpoint]
```

3. Le AWS Toolkit détecte votre LocalStack profil et met à jour le menu d'état de la connexion.

Après la configuration, le choix de votre LocalStack profil AWS dans la section profil de la barre d'état rend vos LocalStack ressources visibles dans l' AWS explorateur. En outre, vous pouvez consulter vos LocalStack journaux dans l'onglet Sortie du terminal VS Code.

Démarrage LocalStack dans VS Code

Vous pouvez commencer à LocalStack utiliser l'une des méthodes suivantes :

À LocalStack partir de la barre d'état de VS Code

1. Dans VS Code, accédez à la barre d'état, puis cliquez sur le LocalStack bouton Démarrer pour le lancer LocalStack.
2. La barre d'état VS Code est mise à jour lorsqu'elle LocalStack est lancée avec succès.

À LocalStack partir du code VS Palette de commandes

1. Dans VS Code, ouvrez la palette de commandes en appuyant sur **`Cmd + Shift + P`** (Mac) ou **`Control + Shift + P`** (Windows).
2. Dans la palette de commandes, entrez **`Start LocalStack`** dans la barre de recherche et choisissez-la dans la liste lorsqu'elle apparaît dans les résultats.
3. La barre d'état VS Code est mise à jour lorsqu'elle LocalStack est lancée avec succès.

À LocalStack partir du terminal VS Code

1. Depuis VS Code, ouvrez le terminal VS Code en appuyant sur**`ctrl + `(backtick)`**.
2. Depuis le terminal VS Code, entrez la commande **`localstack start`** CLI.
3. La barre d'état VS Code est mise à jour lorsqu'elle LocalStack est lancée avec succès.

Création d'un exemple d'application sans serveur

Pour commencer à travailler LocalStack dans VS Code, vous avez besoin d'un exemple d'application sans serveur. Si vous avez déjà une application existante dans votre AWS compte, vous pouvez la déployer localement à l'aide de Serverless Land LocalStack ou en créer une nouvelle avec AWS Serverless Land.

Pour plus d'informations sur la création d'une application avec Serverless Land dans la AWS boîte à outils, consultez la rubrique Utilisation de AWS Serverless Land dans ce guide de l'utilisateur. Pour des informations détaillées sur Serverless Land, consultez la page d'accueil principale de l'application Web Serverless Land.

Tester et déboguer les fonctions Lambda avec LocalStack

Tester et déboguer vos fonctions Lambda dans l'extension LocalStack VS Code revient à travailler avec vos fonctions déployées dans le AWS cloud. La principale différence est que votre instance AWS Toolkit doit être authentifiée auprès de votre LocalStack compte pour déployer et déboguer vos fonctions. LocalStack

> **Note**
>
> Les fonctionnalités de test et de débogage décrites dans cette section ne sont pas disponibles pour l'édition LocalStack Community.
> Pour travailler avec LocalStack VS Code, connectez-vous à votre LocalStack profil dans la AWS boîte à outils. Lorsque votre LocalStack profil est actif, la barre d'état de VS Code affiche AWS : profile:localstack (point de terminaison personnalisé) avec une coche.

Pour obtenir des informations détaillées sur l'utilisation de vos fonctions Lambda dans le AWS Toolkit, consultez la rubrique Utilisation des AWS Lambda fonctions de ce guide de l'utilisateur.

AWS Lambda débogage à distance

AWS Toolkit for Visual Studio Code Cela vous permet de déboguer vos AWS Lambda fonctions qui s'exécutent dans le cloud, directement dans VS Code. Le débogage à AWS Lambda distance vous permet d'inspecter les fonctions en cours d'exécution, de définir des points d'arrêt, d'examiner des variables et de procéder à un débogage étape par étape sans modifier leur flux de développement existant.

Les sections suivantes décrivent comment utiliser le débogage à distance Lambda dans le. AWS Toolkit for Visual Studio Code

Comment fonctionne le débogage à distance Lambda

Le AWS Toolkit permet le débogage à distance en modifiant temporairement vos fonctions Lambda avec une couche de débogage Lambda supplémentaire et en prolongeant le délai d'invocation Lambda à 900 secondes. Une connexion sécurisée est établie entre votre débogueur local et l'environnement d'exécution Lambda à l'aide AWS IoT du tunneling sécurisé. Cette connexion vous permet d'utiliser les points d'arrêt de votre code local pour parcourir la fonction lorsqu'elle s'exécute à distance. Une fois votre session de débogage terminée, toutes les modifications temporaires sont automatiquement rétablies à leurs paramètres d'origine.

Démarrage

Environnements d'exécution pris en charge

Les environnements d'exécution suivants sont pris en charge par le débogage à distance Lambda.

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

> **Note**
>
> Les instances gérées Lambda et les types de fonctions d'image OCI ne sont pas pris en charge par le débogage à distance Lambda.

Conditions préalables

Avant de commencer, les conditions suivantes doivent être remplies.

- Vous devez disposer d' AWS informations d'identification valides configurées dans le AWS Toolkit. Pour plus de détails sur l'installation du AWS Toolkit et la configuration de vos informations d'identification, consultez la rubrique Démarrage de ce guide de l'utilisateur.
- Une fonction Lambda a été déployée sur votre AWS compte. Pour plus de détails sur le déploiement d'une fonction Lambda, consultez la rubrique Créer votre première fonction Lambda dans le Guide du AWS Lambda développeur.

- Vous devez disposer de la politique Gestion des identités et des accès AWS (IAM) et des autorisations appropriées pour déboguer votre fonction. Pour plus de détails sur les autorisations Lambda, consultez la AWS Lambda rubrique consacrée aux politiques AWS gérées dans le Guide du AWS Lambda développeur. Voici un exemple de politique contenant les autorisations minimales requises pour utiliser le débogage à distance Lambda dans le AWS Toolkit.

> **Note**
>
> Le débogage à distance est activé via AWS AWS IoT Secure Tunneling. Cela permet à votre débogueur local d'établir une connexion sécurisée à l'environnement d'exécution Lambda.

```
{
  "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:OpenTunnel",
        "iot:RotateTunnelAccessToken",
        "iot:ListTunnels"
      ],
      "Resource": "*"
    }
  ]
}
```

Accès au débogage à distance Lambda

Il existe deux chemins principaux pour accéder au débogage à distance de Lambda dans le AWS Toolkit : l' AWS explorateur ou l'explorateur Application Builder. Depuis l' AWS explorateur, vous pouvez accéder au débogage à distance de Lambda via vos AWS Lambda nœuds. Depuis l'explorateur Application Builder, vous pouvez accéder au débogage à distance de Lambda via vos projets locaux AWS SAM .

Accès au débogage à distance Lambda depuis AWS explorateur

1. Dans VS Code, ouvrez l'extension AWS Toolkit.
2. À partir de la AWS boîte à outils, développez l' AWS explorateur.
3. Depuis l'explorateur, développez le nœud Lambda.
4. Accédez à la fonction que vous souhaitez déboguer, puis choisissez l'icône Invoquer à distance dans le menu contextuel pour ouvrir l'écran de configuration de l'appel à distance.

Accès au débogage à distance Lambda depuis l'explorateur Application Builder.

1. Dans VS Code, ouvrez l'extension AWS Toolkit.
2. À partir de la AWS boîte à outils, développez l'explorateur du générateur d'applications.
3. Dans l'explorateur, développez le `AWS SAM` projet qui contient le projet Lambda que vous souhaitez déboguer.
4. Développez la `Lambda` fonction déployée que vous souhaitez déboguer.
5. Accédez à la fonction télécommande, puis choisissez l'icône Invoquer à distance dans le menu contextuel pour ouvrir l'écran de configuration de l'appel à distance.

Utilisation du débogage à distance Lambda

Les sections suivantes décrivent comment utiliser le débogage à distance Lambda dans le. AWS Toolkit for Visual Studio Code

> **Note**
>
> Les fonctions Lambda ont une limite de 5 couches et une limite combinée de 250 Mo pour le code de fonction et toutes les couches associées. Le débogage à distance de Lambda nécessite au moins une couche libre pour s'exécuter.

Configuration d'une session de débogage

Avant de commencer, configurez votre session de débogage en suivant la procédure suivante.

1. Ouvrez le menu de configuration de l'appel à distance en effectuant la procédure Accès au débogage à distance Lambda depuis l' AWS explorateur ou la procédure Accès au débogage à distance Lambda depuis l'explorateur Application Builder, située dans la section précédente.
2. Dans le menu de configuration de l'appel à distance, cochez la case Débogage à distance pour afficher les propriétés du débogage à distance.
3. Spécifiez le chemin racine local vers votre fichier de gestion local.

   > **Note**
   >
   > Le chemin racine local est l'emplacement de votre code source qui correspond à la fonction Lambda déployée. Si vous travaillez à partir d'une fonction déployée dans l'explorateur Application Builder, votre chemin racine local est automatiquement détecté. Si le code source n'est pas stocké localement, cliquez sur le bouton Télécharger le code à distance pour récupérer le code source de votre fonction Lambda. Cela vous ouvrira `handler file` dans l'éditeur VS Code.

4. Dans la section Charge utile, indiquez où les données de vos événements de test sont obtenues.

Définition des points d'arrêt et débogage

Définissez des points d'arrêt et commencez le débogage en suivant la procédure suivante.

1. `handler file`Dans l'éditeur VS Code, cliquez dans la marge de la gouttière pour définir des points d'arrêt aux numéros de ligne où vous souhaitez suspendre le débogage.
2. Lorsque vous êtes satisfait des points d'arrêt, revenez au menu de configuration de l'invocation à distance pour vérifier que vos paramètres sont correctement configurés, puis cliquez sur le bouton d'invocation à distance pour démarrer le débogage.
3. Le AWS Toolkit met à jour votre fonction Lambda avec des fonctionnalités de débogage, établit un tunnel sécurisé pour la session de débogage, appelle votre fonction avec la charge utile spécifiée, puis interrompt le processus lorsqu'il atteint un point d'arrêt.
4. Lors d'une pause, utilisez le volet RUN AND DEBUG pour afficher vos VARIABLES, votre PILE D'APPELS et vos POINTS D'ARRÊT.

Mettre à jour et tester votre fonction

Pour modifier votre code et tester les modifications grâce à un déploiement rapide, effectuez la procédure suivante.

1. Avec votre session de débogage active, apportez des modifications à la vôtre `handler file` dans l'éditeur VS Code.
2. Enregistrez vos modifications (**`Command+S on macOS`**,**`Ctrl+S on Windows`**)
3. Lorsque vous y êtes invité, confirmez que vous souhaitez poursuivre le déploiement de vos modifications. Le AWS Toolkit mettra à jour votre fonction Lambda avec le code modifié.
4. Continuez à déboguer et à tester vos modifications en définissant de nouveaux points d'arrêt et en sélectionnant à nouveau le bouton d'invocation à distance.

> **Note**
>
> Vous pouvez également désélectionner l'option Attach Debugger dans les commandes de débogage VS Code et choisir le bouton Remote Invoke pour exécuter votre fonction sans déboguer.

Fin d'une session de débogage

Chacune des options suivantes met fin à votre session de débogage à distance et supprime la couche de débogage de votre projet.

- Choisissez l'option Supprimer la configuration du débogage dans l'écran de configuration de l'appel à distance.
- Choisir l'icône de déconnexion dans les commandes de débogage de VS Code.
- Fermez le `handler file` dans l'éditeur VS Code.

> **Note**
>
> Notez les informations suivantes :
>
> - La couche de débogage Lambda est automatiquement supprimée après 60 secondes d'inactivité. Le décompte commence lorsque votre dernier appel est terminé.

- Si vous avez apporté des modifications de code à vos fonctions gérées par infrastructure en tant que code (IaC) (AWS SAM, AWS CDK, Terraform) pendant le processus de débogage, enregistrez-les dans votre projet local et envisagez de mettre à jour votre référentiel de contrôle des sources. Les modifications non enregistrées sont remplacées lorsque votre fonction IaC est redéployée.
- Si vous avez apporté des modifications temporaires à des fins de débogage uniquement, vous souhaiterez peut-être redéployer votre fonction depuis votre contrôle de source pour vous assurer qu'elle correspond à votre code de production.

Débogage des fonctions TypeScript Lambda avec des cartes sources

Les sections suivantes décrivent comment déboguer vos fonctions TypeScript Lambda à l'aide de cartes sources.

Conditions préalables

Pour déboguer vos fonctions TypeScript Lambda, les conditions préalables suivantes doivent être remplies.

- Vous TypeScript devez être compilé avec l'option de carte source activée. Pour plus d'informations, consultez la rubrique relative à la prise en charge de la carte JavaScript source dans la documentation VS Code.
- In-line les cartes sources ne sont pas prises en charge. Vous devez utiliser un `.js.map` fichier distinct pour stocker la carte source.

Configuration

Pour configurer le débogage à distance Lambda pour les fonctions TypeScript Lambda dans le AWS Toolkit, procédez comme suit.

1. À partir de la AWS boîte à outils, développez l' AWS explorateur.
2. Depuis l'explorateur, développez le nœud Lambda.
3. Accédez à la fonction pour laquelle vous souhaitez configurer TypeScript, puis choisissez l'icône Invoquer à distance dans le menu contextuel pour ouvrir l'écran de configuration de l'appel à distance.
4. Activez le débogage à distance en cochant la case Débogage à distance.

5. Configurez votre chemin racine local en pointant vers le répertoire contenant votre`TypeScript handler file`.

   > **Note**
   >
   > `TypeScript handler file`C'est là que vous définissez vos points d'arrêt de débogage.

6. Développez les paramètres de configuration supplémentaires du débogage à distance.
7. Activez le mappage des sources en cochant la case Carte des sources.
8. Définissez le champ Fichiers de sortie sur le répertoire local de votre copie de fonction Lambda.

   Example

   Si `app.js` et `app.map` sont dedans`.aws-sam/build/HelloWorldFunction`, définissez l'emplacement des fichiers de sortie`/Users/`**`user`**`/`**`project`**`/aws-sam/build/`**`HelloWorldFunction`**`/*`.

   > **Note**
   >
   > Le chemin du fichier de sortie doit être un chemin absolu.
   > Pour les AWS CDK projets AWS SAM et, la AWS boîte à outils prend en charge la détection automatique des cartes sources. Si le champ Fichiers de sortie est laissé vide pour ces projets, la boîte à outils tentera automatiquement de détecter l'emplacement de la carte source.

9. Lorsque vous êtes satisfait des paramètres, cliquez sur le bouton d'invocation à distance pour commencer à déboguer votre TypeScript fonction.

Résolution des problèmes et cas d'utilisation avancés

Si votre session de débogage échoue, lancez le processus de dépannage en suivant ces étapes.

1. Mettez à jour la AWS boîte à outils vers la dernière version.
2. Actualisez la vue Web en fermant la vue Web de configuration de l'invocation à distance et en la rouvrant.
3. Redémarrez VS Code en le fermant complètement et en le rouvrant.

4. Ouvrez la palette de commandes VS Code et entrez la commande**`AWS: Reset Lambda Remote Debugging Snapshot`**, sélectionnez-la lorsqu'elle apparaît dans les résultats pour réinitialiser votre instantané de débogage à distance Lambda.
5. Si vous ne parvenez pas à résoudre le problème, soumettez-le à la section AWS Toolkit for Visual Studio Code GitHub Problèmes.

Cas d'utilisation avancé : configuration de signature de code

Le débogage à distance nécessite d'associer une couche de débogage à votre fonction Lambda. Si la configuration de signature de code est activée et appliquée, le AWS Toolkit ne peut pas automatiquement associer la couche de débogage à votre fonction.

Il existe deux options pour résoudre le problème de configuration de la signature de code.

- Supprimez temporairement la signature de code.
- Utilisez une couche de débogage signée.

Suppression temporaire de la signature de code

Mettez à jour la configuration de signature de code en paramétrant`UntrustedArtifactOnDeployment : Warn`, puis réactivez-la une `Enforced` fois le processus de débogage terminé.

Pour plus d'informations, consultez la UpdateCodeSigningConfig référence dans la référence de AWS Lambda l'API.

Utilisation d'une couche de débogage signée

1. Depuis Débogage à distance Lambda dans le AWS Toolkit, développez la section Configuration supplémentaire du débogage à distance.
2. Dans la section Configuration supplémentaire du débogage à distance, copiez l'ARN de votre couche régionale à partir du champ Layer Override.
3. À partir de AWS CLI, utilisez la commande suivante pour télécharger la version de la couche`aws lambda get-layer-version-by-arn --arn` **`layer-arn`**, en remplaçant layer-arn par l'ARN de votre couche. Pour obtenir des instructions détaillées sur le téléchargement de la couche de débogage signée, consultez la référence https://docs.aws.amazon.com/cli/latest/reference/lambda/get-layer-version-by-arn.html get-layer-version-by-arn dans la Référence des commandes. AWS CLI

4. Signez la couche avec votre configuration de signature de code et publiez-la sur votre compte. Pour obtenir des conseils en matière de signature et de publication, consultez la rubrique Configuration de la signature de code pour votre AWS SAM application dans le Guide du AWS Serverless Application Model développeur.
5. Une fois la couche signée et publiée sur votre compte, revenez à la section Configuration supplémentaire du débogage à distance du débogage à distance Lambda, puis saisissez l'ARN de la nouvelle couche dans le champ Layer Override. Une fois le processus terminé, le débogage à distance de Lambda utilise votre couche signée au lieu de la couche par défaut.

Cas d'utilisation avancé : fonctions de débogage avec SnapStart ou provisionnement simultané

Pour les fonctions Lambda configurées avec SnapStart ou provisionnées en simultanéité, la publication d'une nouvelle version prend beaucoup plus de temps. Pour accélérer votre flux de travail de débogage, vous pouvez configurer le débogage à distance Lambda pour mettre à jour uniquement la `$LATEST` version de votre fonction au lieu de publier une nouvelle version.

1. À partir de l'écran de configuration de l'appel à distance, développez les paramètres de configuration supplémentaires du débogage à distance.
2. Désélectionnez l'option Publier la version.
3. Le AWS Toolkit ne fera désormais que mettre à jour la `$LATEST` version de votre fonction et le déboguer à l'aide de celle-ci.

> **Note**
>
> Comme effet secondaire du débogage avec la `$LATEST` version, vous devez éviter tout autre trafic susceptible d'invoquer votre `$LATEST` version afin de garantir un environnement de débogage non perturbé.

Régions prises en charge

L'erreur suivante se produit lorsqu'une région ne prend pas en charge le débogage à distance.

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

La liste suivante répertorie les régions prises en charge.

- 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

Les fonctions Lambda ont une limite de 5 couches et une limite combinée de 250 Mo pour le code de fonction et toutes les couches associées. La couche de débogage à distance est d'environ 40 Mo, ce qui peut amener votre fonction à dépasser cette limite si vous disposez d'un package de fonctions volumineux ou de plusieurs couches. Pour plus de détails, consultez la section Lambda : InvalidParameterValueException ou RequestEntityTooLargeException rubrique du Guide du AWS Lambda développeur.

La liste suivante décrit les méthodes permettant de résoudre et de corriger cette erreur.

- Réduire la taille des fonctions : optimisez le code de votre fonction et supprimez les dépendances inutiles.

- Supprimer les couches non utilisées : Supprimez temporairement les couches non essentielles pendant le débogage.
- Utiliser des dépendances externes : déplacez les dépendances importantes vers un stockage externe, tel qu'Amazon S3, et chargez-les lors de l'exécution.

Résolution des problèmes de débogage Java

Pour déboguer une fonction Java Lambda, vous devez avoir installé localement la même version de Java qui correspond à la version d'exécution de votre fonction Lambda.

Par exemple, lors du débogage d'une fonction Java 25, Java 25 doit être installé dans votre environnement local dans lequel le AWS Toolkit est en cours d'exécution. Si vous tentez de déboguer une fonction Java 25 avec Java 21 ou une version antérieure installée localement, le débogage à distance ne pourra pas s'arrêter aux points d'arrêt que vous avez définis.

Assurez-vous que votre version locale de Java correspond à la version d'exécution de votre fonction Lambda avant de démarrer une session de débogage.

Le quota de tunneling sécurisé pour l'IoT a été dépassé

Voici un exemple de l'erreur de dépassement du quota de tunnel qui se produit lorsque vous avez atteint la limite quotidienne de connexions de tunneling AWS IoT sécurisées dans le cadre du débogage à distance Lambda.

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

AWS IoT Les quotas de connexion Secure Tunneling sont les suivants :

- Free-tier Le tunneling sécurisé IoT est attribué à 10 connexions par jour.
- Chaque tunnel prend en charge une instance VS Code pendant une durée maximale de 12 heures.
- Le quota s'applique par AWS compte et par jour.

Si vous rencontrez une erreur de tunneling AWS IoT sécurisé, attendez la réinitialisation quotidienne du quota ou contactez le AWS support pour demander une augmentation de la limite de quota. Pour obtenir les informations de contact de l' AWS assistance, consultez le portail de contact de l'AWS assistance. Pour des informations détaillées sur le tunneling AWS IoT sécurisé, consultez la rubrique sur le tunneling AWS IoT sécurisé dans le Guide du développeur. AWS IoT

Amazon Redshift dans le kit d'outils pour VS Code

Amazon Redshift est un service d'entrepôt des données entièrement géré dans le cloud. Pour obtenir des informations détaillées sur le service Amazon Redshift, consultez la table des matières des guides de l'utilisateur Amazon Redshift.

Les rubriques suivantes décrivent comment utiliser Amazon Redshift à partir du. AWS Toolkit for Visual Studio Code

Rubriques

- Utilisation d'Amazon Redshift à partir du Toolkit for VS Code

Utilisation d'Amazon Redshift à partir du Toolkit for VS Code

Les sections suivantes décrivent comment commencer à travailler avec Amazon Redshift à partir du. AWS Toolkit for Visual Studio Code

Pour obtenir des informations détaillées sur le service Amazon Redshift, consultez les rubriques du guide de l'utilisateur Amazon Redshift.

Premiers pas

Avant de commencer à travailler avec Amazon Redshift à partir du AWS Toolkit for Visual Studio Code, les conditions suivantes doivent être remplies.

1. Vous êtes connecté à votre/vos AWS compte (s) depuis le Toolkit. Pour plus d'informations sur la connexion à votre AWS compte depuis le kit d'outils, consultez la AWS rubrique Connexion à du présent guide de l'utilisateur.
2. Vous avez créé un entrepôt de données provisionné ou sans serveur.

Si vous n'avez pas encore créé de cluster Amazon Redshift Serverless ou Amazon Redshift provisionné, les procédures suivantes décrivent comment créer un entrepôt de données avec un exemple de jeu de données, à partir de la console. AWS

Création d'un entrepôt de données provisionné

Pour plus d'informations sur la création d'un entrepôt de données de clusters approvisionné par Amazon Redshift, consultez la rubrique Créer un exemple de cluster Amazon Redshift dans le guide de l'utilisateur d'Amazon Redshift pour démarrer.

1. Depuis votre navigateur Internet préféré, connectez-vous à la console de AWS gestion et ouvrez la console Amazon Redshift à l'adresse. https://console.aws.amazon.com/redshift/
2. Dans la console Amazon Redshift, choisissez le tableau de bord Provisioned Clusters.
3. Dans le tableau de bord des clusters provisionnés, cliquez sur le bouton Créer un cluster pour ouvrir le volet Créer un cluster.
4. Renseignez les champs obligatoires dans la section Configuration du cluster.
5. Dans la section Exemples de données, sélectionnez la case Charger des exemples de données pour charger l'exemple de jeu de données **`Tickit`** dans la base de données par défaut **`Dev`** avec le **`public`** schéma.
6. Dans la section Configurations de base de données, saisissez des valeurs pour les champs Nom d'utilisateur administrateur et Mot de passe utilisateur Administrateur.
7. Choisissez Create cluster pour créer votre entrepôt de données provisionné.

Création d'un entrepôt de données sans serveur

Pour plus d'informations sur la création d'un entrepôt de données sans serveur Amazon Redshift, consultez la section Création d'un entrepôt de données avec Amazon Redshift Serverless dans le guide de démarrage d'Amazon Redshift.

1. Depuis votre navigateur Internet préféré, connectez-vous à la console de AWS gestion et ouvrez la console Amazon Redshift à l'adresse. https://console.aws.amazon.com/redshift/
2. Depuis la console Amazon Redshift, cliquez sur le bouton Essayer Amazon Redshift Serverless pour ouvrir le volet Commencer avec Amazon Redshift Serverless.
3. Dans la section Configurations, choisissez le radial Utiliser les paramètres par défaut.
4. Au bas du volet Commencer avec Amazon Redshift Serverless, choisissez Enregistrer la configuration pour créer un entrepôt de données sans serveur avec des paramètres de groupe de travail, d'espace de noms, d'informations d'identification et de chiffrement par défaut.

Connexion à un entrepôt de données depuis le Toolkit

Il existe 3 méthodes pour se connecter à une base de données depuis le Toolkit :

- Nom d'utilisateur et mot de passe de base de données
- AWS Secrets Manager

- Informations d'identification temporaires

Pour vous connecter à une base de données située sur un cluster provisionné existant ou un entrepôt de données sans serveur à partir du Toolkit, procédez comme suit.

> **Important**
>
> Si vous avez effectué les étapes décrites dans la section Prérequis de cette rubrique du guide de l'utilisateur et que votre entrepôt de données n'est pas visible dans l'explorateur du Toolkit, assurez-vous que vous travaillez dans la bonne AWS région dans l'explorateur.

Connexion à votre entrepôt de données à l'aide de la méthode du nom d'utilisateur et du mot de passe de la base de données

1. À partir de l'explorateur du Toolkit, élargissez l' Région AWS emplacement de votre entrepôt de données.
2. Développez Redshift et choisissez votre entrepôt de données pour ouvrir la boîte de dialogue Sélectionner un type de connexion dans VS Code.
3. Dans la boîte de dialogue Sélectionnez un type de connexion, choisissez le nom d'utilisateur et le mot de passe de la base de données et fournissez les informations requises par chacune des invites.
4. Vos bases de données, tables et schémas disponibles sont visibles dans l'explorateur du Toolkit lorsque celui-ci se connecte à votre entrepôt de données et que la procédure est terminée.

Connexion à votre entrepôt de données avec AWS Secrets Manager

> **Note**
>
> Cette procédure nécessite un AWS secret de base de données du gestionnaire de secrets pour terminer. Pour savoir comment configurer un secret de base de données, consultez la section Créer un secret de AWS Secrets Manager base de données dans le Guide de l'utilisateur de AWS Secrets Manager.

1. À partir de l'explorateur du Toolkit, élargissez l' Région AWS emplacement de votre entrepôt de données.

2. Développez Redshift et choisissez votre entrepôt de données pour ouvrir la boîte de dialogue Sélectionner un type de connexion dans VS Code.
3. Dans la boîte de dialogue Sélectionnez un type de connexion, choisissez Secrets Manager et fournissez les informations requises par chacune des invites.
4. Vos bases de données, tables et schémas disponibles sont visibles dans l'explorateur du Toolkit lorsque celui-ci se connecte à votre entrepôt de données et que la procédure est terminée.

Connexion à votre entrepôt de données avec des informations d'identification temporaires

1. À partir de l'explorateur de boîtes à outils, développez la AWS région dans laquelle se trouve votre entrepôt de données.
2. Développez Redshift et choisissez votre entrepôt de données pour ouvrir la boîte de dialogue Sélectionner un type de connexion dans VS Code.
3. Dans la boîte de dialogue Sélectionnez un type de connexion, choisissez Informations d'identification temporaires et fournissez les informations requises par chacune des invites.
4. Vos bases de données, tables et schémas disponibles sont visibles dans l'explorateur du Toolkit lorsque celui-ci se connecte à votre entrepôt de données et que la procédure est terminée.

Modification de la connexion à votre entrepôt de données

Vous pouvez modifier la connexion à votre entrepôt de données pour modifier la base de données à laquelle vous souhaitez vous connecter.

1. À partir de l'explorateur du Toolkit, élargissez l' Région AWS emplacement de votre entrepôt de données.
2. Développez Redshift, cliquez avec le bouton droit sur l'entrepôt de données auquel vous êtes connecté, choisissez Modifier la connexion et indiquez le nom de la base de données à laquelle vous souhaitez vous connecter.
3. Vos bases de données, tables et schémas disponibles sont visibles dans l'explorateur du Toolkit lorsque celui-ci se connecte à votre entrepôt de données et que la procédure est terminée.

Supprimer la connexion à votre entrepôt de données

1. À partir de l'explorateur du Toolkit, élargissez l' Région AWS emplacement de votre entrepôt de données.

2. Développez Redshift, cliquez avec le bouton droit sur l'entrepôt de données contenant la connexion que vous souhaitez supprimer, puis choisissez Supprimer la connexion. Cela supprime les bases de données, les tables et les schémas disponibles de l'explorateur du Toolkit.
3. Pour vous reconnecter à votre entrepôt de données, choisissez Cliquez pour vous connecter et fournissez les informations requises par chacune des instructions. Par défaut, la reconnexion utilise la méthode d'authentification précédente pour se connecter à l'entrepôt de données. Pour utiliser une autre méthode, cliquez sur la flèche de retour dans la boîte de dialogue jusqu'à ce que vous obteniez l'invite d'authentification.

Exécution d'instructions SQL

Les procédures suivantes décrivent comment créer et exécuter des instructions SQL dans votre base de données à partir du AWS Toolkit for Visual Studio Code.

> **Note**
>
> Pour effectuer les étapes de chacune des procédures suivantes, vous devez d'abord suivre la section Connexion à un entrepôt de données à partir du kit d'outils, située dans cette rubrique du guide de l'utilisateur.

1. Dans l'explorateur du Toolkit, développez Redshift, puis développez l'entrepôt de données qui contient la base de données que vous souhaitez interroger.
2. Choisissez Create-Notebook pour spécifier un nom de fichier et un emplacement pour stocker votre bloc-notes localement, puis cliquez sur OK pour ouvrir le bloc-notes dans votre éditeur VS Code.
3. Dans l'éditeur VS Code, entrez les instructions SQL que vous souhaitez stocker dans ce bloc-notes.
4. Cliquez sur le bouton Tout exécuter pour exécuter les instructions SQL que vous avez saisies.
5. Le résultat de vos instructions SQL est affiché sous les instructions que vous avez saisies.

Ajouter Markdown à un bloc-notes

1. Dans votre bloc-notes dans l'éditeur VS Code, cliquez sur le bouton Markdown pour ajouter une cellule Markdown à votre bloc-notes.

2. Entrez votre Markdown dans la cellule prévue à cet effet.
3. La cellule Markdown peut être modifiée à l'aide des outils d'édition situés dans le coin supérieur droit de la cellule Markdown.

Ajouter du code à un bloc-notes

1. Depuis votre bloc-notes dans l'éditeur VS Code, cliquez sur le bouton Code pour ajouter une cellule Code à votre bloc-notes.
2. Entrez votre code dans la cellule prévue à cet effet.
3. Vous pouvez choisir d'exécuter votre code au-dessus ou en dessous de la cellule Code en sélectionnant le bouton approprié dans les outils de l'éditeur de cellules, situés dans le coin supérieur droit de la cellule Code.

Travailler avec Amazon S3

Amazon Simple Storage Service (Amazon S3) est un service de stockage de données évolutif. AWS Toolkit for Visual Studio Code Cela vous permet de gérer vos objets et ressources Amazon S3 directement depuis VS Code.

Pour obtenir des informations détaillées sur le service Amazon S3, consultez le guide de l'utilisateur Amazon S3.

Les rubriques suivantes décrivent comment utiliser les objets et les ressources Amazon S3 à partir du AWS Toolkit for Visual Studio Code.

Rubriques

- Utilisation des ressources Amazon S3
- Utilisation des objets Amazon S3

Utilisation des ressources Amazon S3

Vous pouvez utiliser Amazon S3 depuis le AWS Toolkit for Visual Studio Code pour consulter, gérer et modifier vos compartiments Amazon S3 et d'autres ressources.

Les sections suivantes décrivent comment utiliser les ressources Amazon S3 à partir du. AWS Toolkit for Visual Studio Code Pour plus d'informations sur l'utilisation d'objets Amazon S3, tels que

des dossiers et des fichiers, depuis le AWS Toolkit for Visual Studio Code, consultez la rubrique Utilisation des objets S3 dans ce guide de l'utilisateur.

Création d'un compartiment Amazon S3

1. Dans l'explorateur du Toolkit, ouvrez le menu contextuel (clic droit) du service S3, puis choisissez Create Bucket... . Vous pouvez également choisir l'icône Create Bucket pour ouvrir la boîte de dialogue Create Bucket.
2. Dans le champ Nom du compartiment, saisissez le nom du compartiment.

   Appuyez sur Entrée pour créer le compartiment et fermer la boîte de dialogue. Votre nouveau compartiment est ensuite affiché sous le service S3 dans le kit d'outils.

   > **Note**
   >
   > Amazon S3 autorisant l'utilisation de votre compartiment en tant qu'URL accessible publiquement, le nom du compartiment que vous choisissez doit être unique au niveau mondial. Si un autre compte a déjà créé un bucket portant le nom que vous souhaitez utiliser, vous devez utiliser un autre nom.
   > Si vous ne parvenez pas à créer un nouveau bucket, consultez les journaux du AWS Toolkit dans l'onglet Output. Si vous essayez d'utiliser un nom de compartiment non valide, une `BucketAlreadyExists` erreur se produit.
   > Pour de plus amples informations, consultez Limites et restrictions applicables aux compartiments dans le Guide de l'utilisateur Amazon Simple Storage Service.

Ajout d'un dossier à un compartiment Amazon S3

Vous pouvez organiser le contenu d'un compartiment S3 en regroupant vos objets dans des dossiers. Vous pouvez également créer des dossiers dans des dossiers.

1. Dans l'explorateur du Toolkit, étendez le service S3 pour afficher la liste de vos ressources S3.
2. Cliquez sur l'icône Créer un dossier pour ouvrir la boîte de dialogue Créer un dossier. Vous pouvez également ouvrir le menu contextuel (clic droit) d'un compartiment ou d'un dossier, puis choisir Créer un dossier.
3. Entrez une valeur dans le champ Nom du dossier et appuyez sur Entrée pour créer le dossier et fermer la boîte de dialogue. Votre nouveau dossier s'affiche sous la ressource S3 correspondante dans le menu de la boîte à outils.

Suppression d'un compartiment Amazon S3

Lorsque vous supprimez un compartiment S3, vous supprimez également les dossiers et les objets qu'il contient. Ainsi, lorsque vous tentez de supprimer un bucket, il vous est demandé de confirmer que vous souhaitez le supprimer.

1. Dans le menu principal du kit d'outils, développez le service Amazon S3 pour afficher la liste de vos ressources S3.
2. Ouvrez le menu contextuel (clic droit) d'un compartiment ou d'un dossier, puis choisissez Supprimer le compartiment S3.
3. Lorsque vous y êtes invité, entrez le nom du compartiment dans le champ de texte, puis appuyez sur Entrée pour supprimer le compartiment et fermer l'invite de confirmation.

> **Note**
>
> Si votre compartiment contient des objets, il est vidé avant d'être supprimé. Si vous tentez de supprimer un grand nombre de ressources ou d'objets à la fois, leur suppression peut prendre un certain temps. Une fois qu'ils ont été supprimés, vous recevez une notification indiquant qu'ils ont été supprimés avec succès.

Utilisation des objets Amazon S3

Vos fichiers, dossiers et toutes les autres données stockées dans un compartiment de ressources S3 sont appelés objets S3.

Les sections suivantes décrivent comment utiliser les objets Amazon S3 à partir du AWS Toolkit for Visual Studio Code. Pour en savoir plus sur l'utilisation des ressources Amazon S3, telles que les compartiments S3, depuis le AWS Toolkit for Visual Studio Code, consultez la rubrique Utilisation des ressources S3 dans ce guide de l'utilisateur.

Pagination des objets

Si vous travaillez avec un grand nombre d'objets et de dossiers Amazon S3, la pagination vous permet de spécifier le nombre d'éléments que vous souhaitez afficher sur une page.

1. Accédez à la barre d'activité VS Code et choisissez Extensions.
2. Dans l'extension AWS Toolkit, choisissez l'icône des paramètres, puis sélectionnez Paramètres de l'extension.

3. Sur la page Paramètres, faites défiler la page vers le bas jusqu'au paramètre AWS > S3 : nombre maximum d'éléments par page.
4. Modifiez la valeur par défaut par le nombre d'éléments S3 que vous souhaitez afficher avant que « en charger plus » ne soit affiché.

> **Note**
>
> Les valeurs valides incluent tout nombre compris entre 3 et 1 000. Ce paramètre s'applique uniquement au nombre d'objets ou de dossiers affichés simultanément. Tous les compartiments que vous avez créés sont affichés en même temps. Par défaut, vous pouvez créer jusqu'à 100 compartiments par compte AWS .

5. Fermez la page Paramètres pour confirmer vos modifications.

Vous pouvez également mettre à jour les paramètres d'un fichier au format JSON en choisissant l'icône Ouvrir les paramètres (JSON) en haut à droite de la page des paramètres.

Chargement et téléchargement d'objets Amazon S3

Vous pouvez télécharger des fichiers stockés localement dans vos compartiments Amazon S3 ou télécharger des objets Amazon S3 distants sur votre système local, depuis le. AWS Toolkit for Visual Studio Code

Téléchargez un fichier à l'aide de la boîte à outils

1. Dans l'explorateur du Toolkit, étendez le service Amazon S3 pour afficher la liste de vos ressources S3.
2. Cliquez sur l'icône Upload File située à côté d'un bucket ou d'un dossier pour ouvrir la boîte de dialogue Upload File. Vous pouvez également ouvrir le menu contextuel (clic droit) et choisir Charger un fichier.

> **Note**
>
> Pour télécharger un fichier dans le dossier ou la ressource parent de l'objet, ouvrez le menu contextuel (clic droit) de n'importe quel objet S3 et choisissez Upload to Parent.

3. Utilisez le gestionnaire de fichiers de votre système pour sélectionner un fichier, puis choisissez Charger un fichier pour fermer la boîte de dialogue et charger le fichier.

Téléchargez un fichier à l'aide de la palette de commandes

Vous pouvez utiliser l'interface du Toolkit ou la palette de commandes pour télécharger un fichier dans un bucket.

1. Pour sélectionner un fichier à télécharger, choisissez l'onglet correspondant dans VS Code.
2. Appuyez sur Ctrl+Shift+P pour afficher la palette de commandes.
3. Dans la palette de commandes, entrez la phrase `upload file` pour afficher la liste des commandes recommandées.
4. Choisissez la commande AWS: Charger un fichier pour ouvrir la boîte de dialogue AWS: Charger un fichier.
5. Lorsque vous y êtes invité, choisissez le fichier que vous souhaitez télécharger, puis le compartiment dans lequel vous souhaitez télécharger ce fichier.
6. Confirmez votre téléchargement pour fermer la boîte de dialogue et commencer le processus de téléchargement. Lorsque le téléchargement est terminé, l'objet s'affiche dans le menu de la boîte à outils avec des métadonnées qui incluent la taille de l'objet, la date de dernière modification et le chemin.

Téléchargement d'un objet Amazon S3

1. À partir de l'explorateur du Toolkit, étendez le service S3.
2. À partir d'un compartiment ou d'un dossier, ouvrez le menu contextuel (clic droit) de l'objet que vous souhaitez télécharger. Choisissez ensuite Télécharger sous pour ouvrir la boîte de dialogue Télécharger sous. Vous pouvez également choisir l'icône Télécharger sous située à côté de l'objet.
3. À l'aide du gestionnaire de fichiers de votre système, choisissez un dossier de destination, entrez un nom de fichier, puis choisissez Télécharger pour fermer la boîte de dialogue et démarrer le téléchargement.

Modification d'objets distants

Vous pouvez utiliser le AWS Toolkit for Visual Studio Code pour modifier vos objets Amazon S3 stockés dans vos ressources Amazon S3 distantes.

1. À partir de l'explorateur du Toolkit, étendez le service S3.
2. Développez la ressource S3 qui contient le fichier que vous souhaitez modifier.

3. Pour modifier le fichier, cliquez sur l'icône en forme de crayon (Modifier le fichier).
4. Pour modifier un fichier ouvert en mode lecture seule, visualisez-le dans l'éditeur VS Code, puis choisissez l'icône en forme de crayon située dans le coin supérieur droit de l'interface utilisateur.

> **Note**
>
> - Si vous redémarrez ou quittez VS Code, votre IDE se déconnecte de vos ressources S3. Si des fichiers S3 distants sont en cours de modification lorsque vous vous déconnectez, la modification s'arrête. Vous devez redémarrer VS Code et rouvrir l'onglet d'édition pour reprendre la modification.
> - Le bouton Modifier le fichier se trouve dans le coin supérieur droit de l'interface utilisateur. Il n'est visible que lorsque vous visualisez activement un fichier en lecture seule dans l'éditeur VS Code.
> - Les fichiers autres que du texte ne peuvent pas être ouverts en mode lecture seule. Ils s'ouvrent toujours en mode édition.
> - Vous ne pouvez pas passer du mode édition seule au mode lecture seule, mais inversement.

Copier le chemin d'un objet Amazon S3

La procédure suivante décrit comment copier le chemin d'un objet Amazon S3 depuis le AWS Toolkit for Visual Studio Code.

1. À partir de l'explorateur du Toolkit, étendez le service S3.
2. Développez le compartiment de ressources qui contient l'objet dont vous souhaitez copier le chemin.
3. Ouvrez le menu contextuel (clic droit) de l'objet dont vous souhaitez copier le chemin, puis choisissez Copier le chemin pour copier le chemin de l'objet dans votre presse-papiers local.

Génération d'une URL présignée pour un objet Amazon S3

Vous pouvez partager des objets Amazon S3 privés avec d'autres personnes en accordant des autorisations de téléchargement limitées dans le temps via la fonctionnalité d'URL présignée. Pour plus d'informations, consultez la section Partage d'un objet avec une URL présignée.

1. À partir de l'explorateur du Toolkit, étendez le service S3.
2. À partir d'un compartiment ou d'un dossier, ouvrez le menu contextuel (clic droit) de l'objet que vous souhaitez partager. Choisissez ensuite Générer une URL présignée pour ouvrir la palette de commandes.
3. Dans la palette de commandes, entrez le nombre de minutes pendant lesquelles l'URL peut être utilisée pour accéder à votre objet. Choisissez ensuite Entrée pour confirmer et fermer la boîte de dialogue.
4. Une fois l'URL présignée générée, la barre d'état du code VS affiche l'URL présignée de l'objet qui a été copié dans votre presse-papiers local.

Suppression d'un objet Amazon S3

Si un objet se trouve dans un compartiment non versionné, vous pouvez le supprimer définitivement. Pour les buckets dont le versionnement est activé, une demande de suppression ne supprime pas définitivement cet objet. Dans ce cas, Amazon S3 insère un marqueur de suppression. Pour plus d'informations, consultez la section Suppression de versions d'objets.

1. Dans l'explorateur du Toolkit, étendez le service S3 pour afficher la liste de vos ressources S3.
2. Ouvrez le menu contextuel (clic droit) de l'objet que vous souhaitez supprimer, puis choisissez Supprimer pour ouvrir la boîte de dialogue de confirmation.
3. Choisissez Supprimer. pour confirmer que vous souhaitez supprimer l'objet S3. Fermez ensuite la boîte de dialogue.

Amazon SageMaker Unified Studio pour VS Code

Faisant partie de la prochaine génération d'Amazon SageMaker, Amazon SageMaker Unified Studio est une expérience de développement unifiée qui réunit AWS des services de données, d'analyse, d'intelligence artificielle (IA) et d'apprentissage automatique (ML). Il fournit un endroit pour créer, déployer, exécuter et surveiller les flux de travail à partir d'une interface unique. Pour plus d'informations sur la configuration de l'intégration d'Amazon SageMaker Unifed Studio avec l'IDE VS Code, consultez Configuration de l'intégration d'Amazon SageMaker Unified Studio dans VS Code dans le guide de l'utilisateur d'Amazon SageMaker Unified Studio.

Utilisation d'applications sans serveur

AWS Toolkit for Visual Studio Code Il fournit un support pour Application sans serveur AWS. Les rubriques suivantes décrivent comment commencer à créer et à utiliser des applications AWS Serverless Application Model (AWS SAM), à partir du AWS Toolkit for Visual Studio Code.

Rubriques

- Débuter avec les applications sans serveur
- Utilisation de AWS Serverless Land
- Exécution et débogage de fonctions Lambda directement à partir du code
- Exécution et débogage des ressources locales Amazon API Gateway
- Options de configuration pour le débogage d'applications sans serveur
- Résolution des problèmes liés aux applications sans serveur

Débuter avec les applications sans serveur

Les sections suivantes décrivent comment commencer à créer un à Application sans serveur AWS partir de AWS Toolkit for Visual Studio Code, using AWS Serverless Application Model (AWS SAM) et CloudFormation stacks.

Conditions préalables

Avant de pouvoir créer ou utiliser un Application sans serveur AWS, les conditions préalables suivantes doivent être remplies.

> **Note**
>
> Les opérations suivantes peuvent nécessiter que vous quittiez ou redémarriez VS Code avant que les modifications ne soient terminées.

- Installez l'interface de ligne de AWS SAM commande (CLI). Pour plus d'informations et d'instructions sur l'installation de la AWS SAM CLI, consultez la rubrique Installation de la AWS SAM CLI dans ce guide de AWS Serverless Application Model l'utilisateur.
- Dans votre fichier de AWS configuration, identifiez votre AWS région par défaut. Pour plus d'informations sur votre fichier de configuration, consultez la rubrique Configuration et paramètres du fichier d'identification dans le Guide de l'AWS Command Line Interface utilisateur.

- Installez votre SDK de langue et configurez votre chaîne d'outils. Pour plus d'informations sur la configuration de votre chaîne d'outils, AWS Toolkit for Visual Studio Code consultez la rubrique Configuration de votre chaîne d'outils dans ce guide de l'utilisateur.
- Installez l'extension de support du langage YAML depuis le marché VS Code. Cela est nécessaire pour que CodeLens les fichiers AWS SAM modèles soient accessibles. Pour plus d'informations sur CodeLens, consultez la CodeLenssection de la documentation de VS Code

Autorisations IAM pour les applications sans serveur

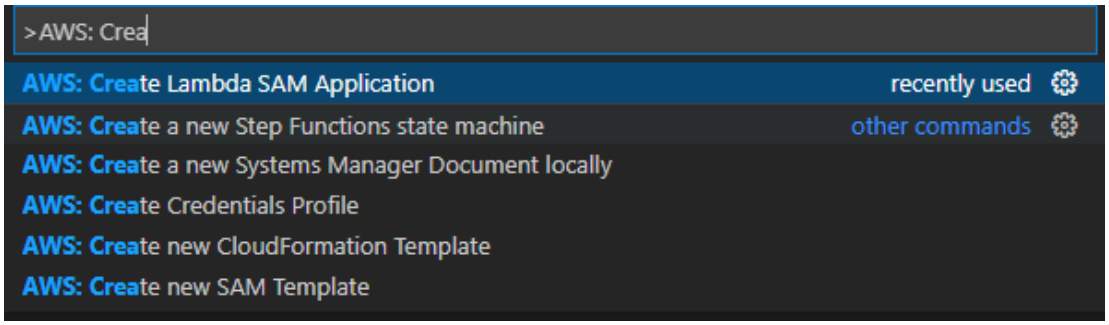
Dans le Toolkit for VS Code, vous devez disposer d'un profil d'identification contenant les autorisations Gestion des identités et des accès AWS (IAM) nécessaires pour déployer et exécuter des applications sans serveur. Vous devez disposer read/write d'un accès approprié aux services suivants : IAM CloudFormation, Lambda, Amazon API Gateway, Amazon Simple Storage Service (Amazon S3) et Amazon Elastic Container Registry (Amazon ECR).

Pour plus d'informations sur la configuration de l'authentification requise pour déployer et exécuter des applications sans serveur, consultez la section Gestion de l'accès aux ressources et des autorisations dans le Guide du AWS Serverless Application Model développeur. Pour plus d'informations sur la configuration de vos informations d'identification, consultez le AWS Informations d'identification IAM présent guide de l'utilisateur.

Création d'une nouvelle application sans serveur (locale)

Cette procédure montre comment créer une application sans serveur avec le Toolkit for VS Code en utilisant AWS SAM. Le résultat de cette procédure est un répertoire local sur votre hôte de développement contenant un exemple d'application sans serveur, que vous pouvez créer, tester localement, modifier et déployer AWS dans le cloud.

1. Pour ouvrir la palette de commandes, choisissez Affichage, Palette de commandes, puis entrez AWS.
2. Choisissez AWS Toolkit Create Lambda SAM Application.



> **Note**
>
> Si la AWS SAM CLI n'est pas installée, vous obtenez une erreur dans le coin inférieur droit de l'éditeur VS Code. Dans ce cas, vérifiez que vous répondez à toutes les hypothèses et à toutes les conditions préalables.

3. Choisissez le moteur d'exécution de votre AWS SAM application.

> **Note**
>
> Si vous sélectionnez l'un des environnements d'exécution avec « (Image) », votre application est de type package `Image`. Si vous sélectionnez l'un des environnements d'exécution sans « (Image) », votre application est de type `Zip`. Pour plus d'informations sur la différence entre les types de packages `Image` et `Zip`, consultez Packages de déploiement Lambda dans le guide du développeur AWS Lambda .

4. En fonction de l'environnement d'exécution sélectionné, il peut vous être demandé de sélectionner un gestionnaire de dépendances et une architecture d'exécution pour votre application SAM.

   Dependency Manager

   Choisissez entre Gradle ou Maven.

   > **Note**
   >
   > Ce choix d'outils d'automatisation de construction n'est disponible que pour les environnements d'exécution Java.

   Architecture

   Choisissez entre x86_64 ou arm64.

   L'option permettant d'exécuter votre application sans serveur dans un environnement émulé ARM64 basé sur x86_64 par défaut est disponible pour les environnements d'exécution suivants :

- nodejs12.x (ZIP et image)
- nodejs14.x (ZIP et image)
- python3.8 (ZIP et image)
- python3.9 (ZIP et image)
- python3.10 (ZIP et image)
- python3.11 (ZIP et image)
- python3.12 (ZIP et image)
- java8.al2 avec Gradle (ZIP et image)
- java8.al2 avec Maven (ZIP uniquement)
- java11 avec Gradle (ZIP et image)
- java11 avec Maven (ZIP uniquement)

> **Important**
>
> Vous devez installer AWS CLI la version 1.33.0 ou ultérieure pour permettre aux applications de s'exécuter dans des environnements ARM64 basés. Pour de plus amples informations, veuillez consulter Prérequis.

5. Choisissez l'emplacement de votre nouveau projet. Vous pouvez utiliser un dossier d'espace de travail existant s'il y en a un ouvert, sélectionner un autre dossier qui existe déjà, ou créer un dossier et le sélectionner. Dans cet exemple, choisissez There are no workspace folders open (Aucun dossier d'espace de travail n'est ouvert) pour créer un dossier nommé `MY-SAM-APP`.
6. Saisissez un nom pour votre projet. Pour cet exemple, utilisez `my-sam-app-nodejs`. Une fois que vous avez appuyé sur Entrée, le Toolkit for VS Code met quelques instants à créer le projet.

Lorsque le projet est créé, votre application est ajoutée à votre espace de travail actuel. Il doit être répertorié dans la fenêtre Explorer.

Ouverture d'une application sans serveur (locale)

Pour ouvrir une application sans serveur sur votre hôte de développement local, ouvrez le dossier contenant le fichier modèle de l'application.

1. Dans le fichier, choisissez Ouvrir un dossier... .

2. Dans la boîte de dialogue Ouvrir un dossier, accédez au dossier de l'application sans serveur que vous souhaitez ouvrir.
3. Cliquez sur le bouton Sélectionner un dossier.

Lorsque vous ouvrez le dossier d'une application, celui-ci est ajouté à la fenêtre de l'explorateur.

Exécution et débogage d'une application sans serveur à partir d'un modèle (local)

Vous pouvez utiliser le Toolkit for VS Code pour configurer comment déboguer des applications sans serveur et les exécuter localement dans votre environnement de développement.

Vous commencez à configurer le comportement de débogage en utilisant la CodeLensfonctionnalité VS Code pour identifier une fonction Lambda éligible. CodeLens permet des interactions basées sur le contenu avec votre code source. Pour savoir comment vous assurer que vous pouvez accéder à cette CodeLens fonctionnalité, consultez la Conditions préalables section ci-dessus dans cette rubrique.

> **Note**
>
> Dans cet exemple, vous devez déboguer une application qui utilise JavaScript. Cependant, vous pouvez utiliser les fonctionnalités de débogage de Toolkit for VS Code avec les langages et environnements d'exécution suivants :
>
> - C# — .NET Core 2.1, 3.1 ; .NET 5.0
> - JavaScript/TypeScript — Node.js 12. x, 14 ans. x
> - Python — 3,6, 3,7, 3,8, 3,9, 3,10, 3,11, 3,12
> - Java — 8, 8.al2, 11
> - Go — 1.x
>
> Votre choix de langue influe également sur la façon dont CodeLens les gestionnaires Lambda éligibles sont détectés. Pour de plus amples informations, veuillez consulter Exécution et débogage de fonctions Lambda directement à partir du code.

Dans cette procédure, vous allez utiliser l'exemple d'application créé dans la Création d'une nouvelle application sans serveur (locale) section précédente de cette rubrique.

1. Pour afficher les fichiers de votre application dans l'explorateur de fichiers de VS Code, choisissez Afficher, Explorateur.
2. Dans le dossier de l'application (par exemple, my-sample-app), ouvrez le `template.yaml` fichier.

   > **Note**
   >
   > Si vous utilisez un modèle dont le nom est différent de`template.yaml`, l' CodeLens indicateur n'est pas automatiquement disponible dans le fichier YAML. Cela signifie que vous devez ajouter manuellement une configuration de débogage.

3. Dans l'éditeur de`template.yaml`, accédez à la `Resources` section du modèle qui définit les ressources sans serveur. Dans ce cas, il s'agit de la `HelloWorldFunction` ressource de type`AWS::Serverless::Function`.

   Dans l' CodeLens indicateur de cette ressource, choisissez Ajouter une configuration de débogage.
   Utilisation de l' CodeLens indicateur dans le `template.yaml` fichier pour ajouter une configuration de débogage.
4. Dans la palette de commandes, sélectionnez le moteur d'exécution dans lequel votre AWS SAM application sera exécutée.
5. Dans l'éditeur du fichier `launch.json`, modifiez ou confirmez les valeurs des propriétés de configuration suivantes :

   - `"name"` – Saisissez un nom convivial à afficher dans le champ déroulant Configuration dans la vue Exécuter.
   - `"target"`— Assurez-vous que la valeur est `"template"` telle que le AWS SAM modèle soit le point d'entrée de la session de débogage.
   - `"templatePath"` – Saisissez un chemin relatif ou absolu pour le fichier `template.yaml`.
   - `"logicalId"`— Assurez-vous que le nom correspond à celui spécifié dans la section Ressources du AWS SAM modèle. Dans ce cas, il s'agit de `HelloWorldFunction` de type `AWS::Serverless::Function`.

   Configuration du `launch.json` fichier pour le débogage basé sur un modèle.

Pour plus d'informations sur ces entrées et sur d'autres entrées dans le fichier `launch.json`, consultez Options de configuration pour le débogage d'applications sans serveur.

6. Si la votre configuration de débogage vous convient, enregistrez `launch.json`. Ensuite, pour démarrer le débogage, cliquez sur le bouton vert « play » dans la vue RUN.

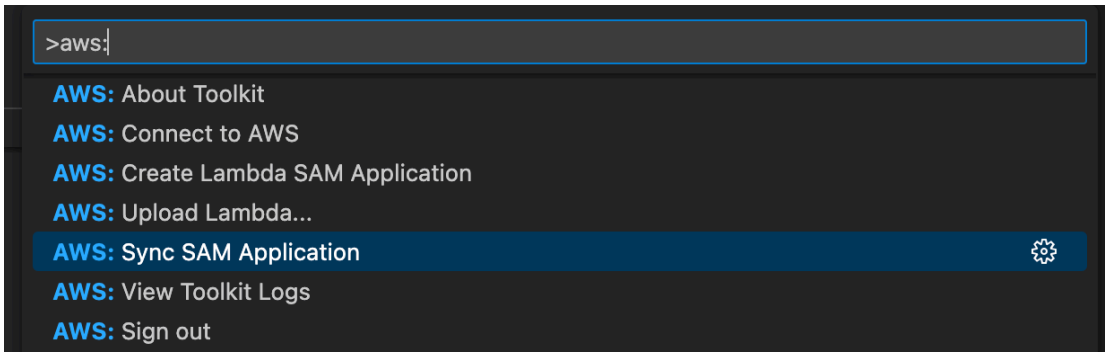
   Lorsque les sessions de débogage démarrent, le panneau CONSOLE DE DÉBOGAGE affiche la sortie de débogage et toutes les valeurs renvoyées par la fonction Lambda. (Lors du débogage d' AWS SAM applications, le AWS Toolkit est sélectionné comme canal de sortie dans le panneau de sortie.)

Synchronisation des applications AWS SAM

AWS Toolkit for Visual Studio Code Exécute la commande AWS SAM CLI `sam sync` pour déployer vos applications sans serveur sur. AWS Cloud Pour plus d'informations sur la AWS SAM synchronisation, consultez la rubrique de référence des commandes AWS SAM CLI dans le Guide du AWS Serverless Application Model développeur

La procédure suivante décrit comment déployer vos applications sans serveur vers le AWS Cloud with à `sam sync` partir du Toolkit for VS Code.

1. Dans le menu principal de VS Code, ouvrez la palette de commandes en développant la vue et en choisissant Palette de commandes.
2. Dans la palette de commandes, recherchez AWSet choisissez Sync SAM Application pour commencer à configurer votre synchronisation.



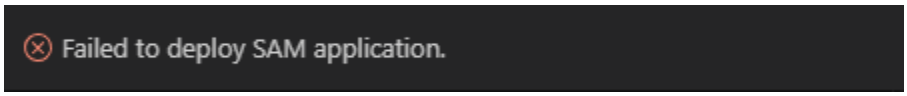
3. Choisissez la AWS région avec laquelle synchroniser votre application sans serveur.
4. Choisissez le fichier `template.yaml` à utiliser pour le déploiement.
5. Sélectionnez un compartiment Amazon S3 existant ou entrez un nouveau nom de compartiment Amazon S3 dans lequel déployer votre application.

> **Important**
>
> Votre compartiment Amazon S3 doit répondre aux exigences suivantes :
>
> - Le compartiment doit se trouver dans la région avec laquelle vous effectuez la synchronisation.
> - Le nom de compartiment S3 doit être unique parmi tous les noms de compartiment existants dans Amazon S3.

6. Si votre application sans serveur inclut une fonction avec le type de package`Image`, saisissez le nom d'un référentiel Amazon ECR que ce déploiement peut utiliser. Le compartiment doit se trouver dans la région dans laquelle vous effectuez le déploiement.
7. Sélectionnez une pile de déploiement dans la liste de vos déploiements précédents, ou créez une nouvelle pile de déploiement en saisissant un nouveau nom de pile. Ensuite, lancez le processus de synchronisation.

> **Note**
>
> Les piles utilisées lors des déploiements précédents sont rappelées par espace de travail et par région.

8. Pendant le processus de synchronisation, l'état de votre déploiement est capturé dans l'onglet Terminal de VS Code. Vérifiez que votre synchronisation est réussie depuis l'onglet du terminal. Si une erreur survient, vous recevez une notification.



> **Note**
>
> Pour plus de détails sur votre synchronisation, les AWS Toolkit for Visual Studio Code journaux sont accessibles depuis la palette de commandes.

Pour accéder à vos AWS Toolkit for Visual Studio Code journaux depuis la palette de commandes, développez Afficher, choisissez Palette de commandes**`AWS: View AWS Toolkits Logs`**, puis recherchez et sélectionnez-la lorsqu'elle apparaît dans la liste.

Lorsque le déploiement est terminé, votre application apparaît dans l'AWS explorateur. Pour plus d'informations sur la façon d'invoquer la fonction Lambda créée dans le cadre de l'application, consultez la Travailler avec AWS Lambda Fonctions rubrique de ce guide de l'utilisateur.

Suppression d'une application sans serveur de AWS Cloud

La suppression d'une application sans serveur implique la suppression de la CloudFormation pile que vous avez précédemment déployée dans le AWS cloud. Notez que cette procédure ne supprime pas votre répertoire d'application de votre hôte local.

1. Ouvrez la AWS Explorateur.
2. Dans la fenêtre AWS Toolkit Explorer, développez la région contenant l'application déployée que vous souhaitez supprimer, puis développez-la CloudFormation.
3. Ouvrez le menu contextuel (clic droit) pour le nom de la CloudFormation pile correspondant à l'application sans serveur que vous souhaitez supprimer, puis choisissez Supprimer la CloudFormation pile.
4. Pour confirmer que vous souhaitez supprimer la pile sélectionnée, choisissez Oui.

Si la suppression de la pile réussit, le Toolkit for VS Code supprime le nom de la pile de la CloudFormation liste dans AWS Explorer.

Utilisation de AWS Serverless Land

AWS Serverless Land in the AWS Toolkit for Visual Studio Code est un ensemble de fonctionnalités qui vous aident à créer des architectures axées sur les événements. Les sections thématiques suivantes décrivent comment utiliser Serverless Land dans le AWS Toolkit. Pour des informations détaillées sur Serverless Land, consultez l'application Web Serverless Land.

Accès à Serverless Land

Il existe 3 points d'entrée principaux pour accéder à Serverless Land dans le AWS Toolkit :

- La palette de commandes VS Code
- L'explorateur de AWS boîtes à outils
- L'explorateur du générateur d'applications AWS Toolkit

Ouvrir Serverless Land à partir de la palette de commandes VS Code

Pour ouvrir Serverless Land à partir de la palette de commandes VS Code, procédez comme suit.

1. Dans VS Code, ouvrez la palette de commandes en appuyant sur **`option+shift+p`** (Mac) ou **`control+shift+p`** (Windows).
2. Dans la palette de commandes VS Code, entrez **`AWS Create application with Serverless template`** dans la barre de recherche.
3. Choisissez AWS: créer une application avec un modèle sans serveur lorsqu'il est renseigné dans la liste.
4. L'assistant Serverless Land s'ouvre sur l'écran Select a Pattern for your application (1/5) dans VS Code une fois le processus terminé.

Ouverture de Serverless Land à partir de l'explorateur de AWS boîtes à outils.

Pour ouvrir Serverless Land à partir de l'explorateur de boîtes à AWS outils, procédez comme suit.

1. À partir de l'explorateur de boîtes à AWS outils, étendez la région dans laquelle vous souhaitez ouvrir Serverless Land.
2. Ouvrez le menu contextuel du nœud Lambda (clic droit).
3. Choisissez Créer une application avec un modèle sans serveur dans le menu contextuel.
4. L'assistant Serverless Land s'ouvre sur l'écran Select a Pattern for your application (1/5) dans VS Code une fois le processus terminé.

Ouvrir Serverless Land à partir de l'explorateur Application Builder

Pour ouvrir Serverless Land à partir de l'explorateur AWS Toolkit Application Builder, procédez comme suit.

1. Dans l'explorateur du AWS Toolkit, accédez à l'explorateur Application Builder.
2. Cliquez avec le bouton droit sur l'explorateur Application Builder et choisissez Create application with Serverless template dans le menu contextuel.
3. L'assistant Serverless Land s'ouvre sur l'écran Select a Pattern for your application (1/5) dans VS Code une fois le processus terminé.

Création d'une application avec un modèle Serverless

Pour créer une application avec un modèle Serverless, procédez comme suit.

1. Dans l'écran de l'assistant Serverless Land, sélectionnez un modèle pour votre application (1/5), choisissez un modèle pour la base de votre application.

   > **Note**
   >
   > Pour afficher un aperçu et plus de détails sur un modèle particulier, cliquez sur l'icône Ouvrir dans Serverless Land située à côté du modèle que vous souhaitez afficher. Le Serverless Land Pattern s'ouvre dans votre navigateur Web par défaut.

2. Dans l'écran Select Runtime (2/5), choisissez un runtime pour votre projet.
3. Sur l'écran Select iAC (3/5), choisissez une option iAc pour votre projet.
4. Dans l'écran Sélectionnez un emplacement de projet (4/5), choisissez un emplacement pour stocker votre projet.
5. Sur l'écran Entrez le nom du projet (5/5), saisissez le nom de votre nouvelle application.
6. Votre nouvelle application s'affiche dans l'explorateur VS Code et votre projet `readme.md` s'ouvre dans l'éditeur VS Code, une fois la procédure terminée.

   > **Note**
   >
   > Une fois votre nouvelle application créée, des actions supplémentaires spécifiques à votre type d'application sont disponibles dans le `readme.md` fichier. En outre, vos applications AWS Serverless Application Model (AWS SAM) peuvent être ouvertes avec AWS Application Builder à des fins de test local, de débogage, etc.
   > Pour plus d'informations sur l'utilisation du Générateur d'applications dans le AWS kit d'outils, consultez la rubrique Utilisation de l'explorateur AWS Application Builder dans ce guide de l'utilisateur.

Exécution et débogage de fonctions Lambda directement à partir du code

Lorsque vous testez l' AWS SAM application, vous pouvez choisir d'exécuter et de déboguer uniquement la fonction Lambda et d'exclure les autres ressources définies par AWS SAM le modèle.

Cette approche implique l'utilisation de cette CodeLensfonctionnalité pour identifier les gestionnaires de fonctions Lambda dans le code source que vous pouvez invoquer directement.

Les gestionnaires Lambda détectés CodeLens dépendent de la langue et de l'environnement d'exécution que vous utilisez pour votre application.

Language/runtime	Critères pour que les fonctions Lambda soient identifiées par des indicateurs CodeLens
C# (dotnetcore2.1, 3.1 ; .NET 5.0)	La fonction de recherche dispose des fonctions suivantes : • C'est une fonction publique d'une classe publique. • Il possède un ou deux paramètres. Avec deux paramètres, le second doit implémenter l'`ILambdaContext` interface. • Il possède un `*.csproj` fichier dans son dossier parent dans le dossier de l'espace de travail VS Code. L'extension ms-dotnettools.csharp (ou toute extension fournissant des symboles de langage pour C#) est installée et activée.
JavaScript/TypeScript (Node.js 12, 14 x)	La fonction de recherche dispose des fonctions suivantes : • Il s'agit d'une fonction exportée avec jusqu'à trois paramètres. • Il possède un `package.json` fichier dans son dossier parent dans le dossier de l'espace de travail VS Code.
Python (3,7, 3,8, 3,9, 3,10, 3,11, 3,12)	La fonction de recherche possède les fonctions suivantes : • Il s'agit d'une fonction de niveau supérieur.

Language/runtime	Critères pour que les fonctions Lambda soient identifiées par des indicateurs CodeLens
	• Il possède un `requirements.txt` fichier dans son dossier parent dans le dossier de l'espace de travail VS Code. L'extension ms-python.python (ou toute extension fournissant des symboles de langage pour Python) est installée et activée.

Language/runtime	Critères pour que les fonctions Lambda soient identifiées par des indicateurs CodeLens

Java (8, 8.al2, 11)	La fonction de recherche dispose des fonctions suivantes :
	• C'est une fonction publique d'une classe publique non abstraite.
	• Il possède un, deux ou trois paramètres :
	  • Un paramètre : le paramètre peut être n'importe quoi.
	  • Deux paramètres : les paramètres doivent être a `java.io.InputStream` et a `java.io.OutputStream` OU le dernier paramètre doit être `com.amazonaws.services.lambda.runtime.Context` a.
	  • Trois paramètres : les paramètres doivent être a `java.io.InputStream` `java.io.OutputStream` et le dernier paramètre doit être `com.amazonaws.services.lambda.runtime.Context` a.
	• Il possède un fichier `build.gradle` (Gradle) ou `pom.xml` (Maven) dans son dossier parent dans le dossier de l'espace de travail VS Code.
	
	L'extension redhat.java (ou toute extension fournissant des symboles de langage pour Java) est installée et activée. Cette extension nécessite Java 11, quel que soit le moteur d'exécution Java que vous utilisez.

Language/runtime	Critères pour que les fonctions Lambda soient identifiées par des indicateurs CodeLens
	L'extension vscjava.vscode-java-debug (ou toute extension fournissant un débogueur Java) est installée et activée.
Go (1.x)	La fonction de recherche possède les fonctions suivantes : • Il s'agit d'une fonction de niveau supérieur. • Il prend entre 0 et 2 arguments. S'il y a deux arguments, le premier argument doit implémenter `context.Context` . • Elle renvoie entre 0 et 2 arguments. S'il y a plus de 0 arguments, le dernier argument doit être implémenté`error`. • Il contient un `go.mod` fichier dans le dossier de l'espace de travail VS Code. L'extension golang.go est installée, configurée et activée.

Pour exécuter et déboguer une application sans serveur directement à partir du code de l'application

1. Pour afficher les fichiers de votre application dans l'explorateur de fichiers VS Code, choisissez Afficher, Explorateur.
2. Dans le dossier de l'application (par exemple, my-sample-app), développez le dossier des fonctions (dans ce cas, hello-world) et ouvrez le fichier. `app.js`
3. Dans l' CodeLens indicateur qui identifie un gestionnaire de fonctions Lambda éligible, choisissez. `Add Debug Configuration`
   Accédez à l'option Ajouter une configuration de débogage dans l' CodeLens indicateur d'un gestionnaire de fonctions Lambda.
4. Dans la palette de commandes, sélectionnez le moteur d'exécution dans lequel votre AWS SAM application sera exécutée.

5. Dans l'éditeur du fichier `launch.json`, modifiez ou confirmez les valeurs des propriétés de configuration suivantes :

   - `"name"` : saisissez un nom convivial à afficher dans le champ déroulant Configuration dans la vue Run (Exécuter).
   - `"target"` – Vérifiez que la valeur est `"code"` afin qu'un gestionnaire de fonction Lambda soit directement appelé.
   - `"lambdaHandler"` – Saisissez le nom de la méthode dans votre code que Lambda appelle pour exécuter votre fonction. Par exemple, pour les applications dans JavaScript, la valeur par défaut est`app.lambdaHandler`.
   - `"projectRoot"` – Saisissez le chemin du fichier d'application qui contient la fonction Lambda.
   - `"runtime"` – Saisissez ou confirmez un environnement d'exécution valide pour l'environnement d'exécution Lambda, par exemple `"nodejs12.x"`.
   - `"payload"` – Choisissez l'une des options suivantes pour définir la charge utile d'événement à fournir à votre fonction Lambda comme entrée :
     - `"json"`: paires JSON-formatted clé-valeur qui définissent la charge utile de l'événement.
     - `"path"` : chemin du fichier utilisé comme charge utile d'événement.

     Dans l'exemple ci-dessous, l'`"json"`option définit la charge utile.

   Configuration du `launch.json` fichier pour appeler directement les fonctions Lambda.

   Pour plus d'informations sur ces entrées et sur d'autres entrées dans le fichier `launch.json`, consultez Options de configuration pour le débogage d'applications sans serveur.

6. Si vous êtes satisfait de la configuration de débogage, pour démarrer le débogage, cliquez sur la flèche verte de lecture à côté de RUN.

   Lorsque les sessions de débogage démarrent, le panneau DEBUG CONSOLE affiche les résultats de débogage et affiche toutes les valeurs renvoyées par la fonction Lambda. (Lors du débogage d' AWS SAM applications, AWS Toolkit est sélectionné comme canal de sortie dans le panneau de sortie.)

Exécution et débogage des ressources locales Amazon API Gateway

Vous pouvez exécuter ou déboguer les ressources locales d' AWS SAM API Gateway, spécifiées dans`template.yaml`, en exécutant une configuration de lancement VS Code `type=aws-sam` avec le`invokeTarget.target=api`.

> **Note**
>
> API Gateway prend en charge deux types de protocole APIs, REST et HTTP. Cependant, la fonctionnalité API Gateway prend AWS Toolkit for Visual Studio Code uniquement en charge le protocole REST APIs. Parfois, le protocole HTTP APIs est appelé « API Gateway V2 » APIs.

Pour exécuter et déboguer des ressources API Gateway locales

1. Choisissez l'une des méthodes suivantes pour créer une configuration de lancement pour une ressource API Gateway AWS SAM :

   - Option 1 : Accédez au code source du gestionnaire (fichier .js, .cs ou .py) de votre AWS SAM projet, survolez le gestionnaire Lambda et choisissez Ajouter une configuration de débogage. CodeLens Ensuite, dans le menu, choisissez l'élément marqué API Event.
   - Option 2 : modifiez `launch.json` et créez une nouvelle configuration de lancement à l'aide de la syntaxe suivante.

```
{
  "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. Dans le panneau VS Code Run, choisissez la configuration de lancement (nommée `myConfig` dans l'exemple ci-dessus).
3. (Facultatif) Ajoutez des points d'arrêt à votre code de projet Lambda.
4. Tapez F5 ou choisissez Play dans le panneau Exécuter.
5. Consultez les résultats dans le volet de sortie.

Configuration

Lorsque vous utilisez la valeur de propriété `invokeTarget.target` `api`, la boîte à outils modifie la validation et le comportement de la configuration de lancement pour prendre en charge un champ `api`.

```
{
  "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": {}
}
```

Remplacez les valeurs dans l'exemple comme suit :

invokeTarget.logicalId

Ressource d'API.

path

Chemin d'API demandé par la configuration de lancement, par exemple,`"path": "/hello"`.

Doit être un chemin d'API valide résolu à partir de `template.yaml` spécifié par `invokeTarget.templatePath`.

httpMethod

Un des verbes suivants : « delete », « get », « head », « options », « patch », « post », « put ».

payload

Charge utile JSON (corps HTTP) à envoyer dans la demande, avec la même structure et les mêmes règles que le champ lambda.payload.

`payload.path` pointe vers un fichier contenant la charge utile JSON.

`payload.json` spécifie une charge utile JSON en ligne.

headers

Mappage facultatif de paires nom-valeur que vous utilisez pour spécifier les en-têtes HTTP à inclure dans la demande, comme indiqué dans l'exemple suivant.

```
"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

Chaîne facultative qui définit la `querystring` de la demande, par exemple, `"querystring": "abc=def&ghi=jkl"`.

AWS

Comment les informations de AWS connexion sont fournies. Pour plus d'informations, consultez le tableau des propriétés de AWS connexion (« aws ») dans la Options de configuration pour le débogage d'applications sans serveur section.

sam

Comment la AWS SAM CLI crée l'application. Pour plus d'informations, consultez le tableau des propriétés de la CLI AWS SAM (« sam ») dans la Options de configuration pour le débogage d'applications sans serveur section.

Options de configuration pour le débogage d'applications sans serveur

Lorsque vous ouvrez le `launch.json` fichier pour modifier les configurations de débogage, vous pouvez utiliser la IntelliSensefonctionnalité VS Code pour afficher et compléter automatiquement les propriétés valides. Pour déclencher IntelliSense dans l'éditeur, appuyez sur Ctrl + barre d'espace.

IntelliSense vous permet de rechercher et de définir des propriétés pour appeler des fonctions Lambda directement ou avec AWS SAM le modèle. Vous pouvez également définir des propriétés pour `"lambda"` (comment la fonction s'exécute), `"sam"` (comment la AWS SAM CLI crée l'application) et `"aws"` (comment les informations de AWS connexion sont fournies).

AWS SAM: appel direct du gestionnaire Lambda//Invocation Lambda basée sur un modèle

Propriété	Description
`type`	Spécifie l'extension qui gère la configuration de lancement. Configurez toujours `aws-sam` sur pour utiliser la AWS SAM CLI pour créer et déboguer localement.
`name`	Spécifie un nom convivial de lecteur qui doit apparaître dans la liste Configuration du lancement de débogage.
`request`	Spécifie le type de configuration à effectuer en fonction de l'extension désignée (`aws-sam`). Toujours défini sur `direct-invoke` pour démarrer la fonction Lambda.
`invokeTarget`	Spécifie le point d'entrée pour l'appel de la ressource. Pour appeler directement la fonction Lambda, définissez des valeurs pour les champs `invokeTarget` : • `target` – Défini sur `code`. • `lambdaHandler` – Nom du gestionnaire de fonction Lambda à appeler. • `projectRoot` — Le chemin du fichier d'application contenant le gestionnaire de fonctions Lambda. • `architecture` — Architecture du processeur de l'environnement émulé dans lequel s'exécute votre application SAM Lambda locale. Pour certains environnements d'exécution, vous pouvez choisir l'`x86_64`architecture par défaut à la `arm64` place. Pour de plus amples informations, veuillez consulter Création d'une nouvelle application sans serveur (locale). Pour appeler les ressources Lambda avec AWS SAM le modèle, définissez les valeurs des champs suivants : `invokeTarget` • `target` – Défini sur `template`.

Propriété	Description
	• `templatePath` — Le chemin d'accès au fichier AWS SAM modèle. • `logicalId` – Nom de la ressource de la fonction `AWS::Lambda::Function` ou de la fonction `AWS::Serverless::Function` à appeler. Le nom de la ressource se trouve dans le modèle au format YAML AWS SAM . Notez que les fonctions définies dans le AWS SAM modèle sont AWS Toolkit implicitement reconnues comme `PackageType: Image` des fonctions Lambda basées sur des images. Pour plus d'informations, consultez Packages de déploiement M Lambda dans le Guide du développeur AWS Lambda .

Propriétés Lambda (**`"lambda"`**)

Propriété	Description

`environmentVariables`	Transmet les paramètres opérationnels à votre fonction Lambda. Par exemple, si vous écrivez dans un compartiment Amazon S3, au lieu de coder en dur le nom du compartiment dans lequel vous écrivez, configurez-le en tant que variable d'environnement.
	
	Note
	Lorsque vous spécifiez des variables d'environnement pour une application sans serveur, vous devez ajouter des configurations au AWS SAM modèle (`template.yaml`) et au `launch.json` fichier.
	Exemple de mise en forme d'une variable d'environnement dans le AWS SAM modèle :
	
	`Resources:`
	`HelloWorldFunction:`
	`Type: AWS::Serverless::Function`
	`Properties:`
	`CodeUri: hello-world/`

Propriété	Description

	<pre>Handler: app.lambdaHandlerN10
	Runtime: nodejs10.x
	Environment:
	  Variables:
	  SAMPLE1: Default Sample 1 Value</pre>
	Exemple de mise en forme d'une variable d'environnement dans le `launch.json` fichier :
	<pre>"environmentVariables": {
	  "SAMPLE1": "My sample 1 value"
	}</pre>

`payload`	Fournit deux options pour la charge utile d'événement que vous fournissez à votre fonction Lambda en entrée. • `"json"` : paires de clé-valeur au format JSON qui définissent la charge utile d'événement. • `"path"` : chemin du fichier utilisé comme charge utile d'événement.
`memoryMB`	Spécifie les mégaoctets (Mo) de mémoire fournis pour exécuter une fonction Lambda invoquée.
`runtime`	Spécifie le temps d'exécution utilisé par la fonction Lambda. Pour plus d'informations, consultez Environnement d'exécution AWS Lambda.
`timeoutSec`	Définit le temps alloué en secondes, avant l'expiration de la séance de débogage.

Propriété	Description
`pathMappings`	Spécifie où se trouve le code local par rapport à l'endroit où il s'exécute dans le conteneur. Par défaut, le Toolkit for VS Code définit `localRoot` la racine de code de la fonction Lambda dans l'espace de travail local, et `remoteRoot` le répertoire de travail par défaut pour le code exécuté dans Lambda. `/var/task` Si le répertoire de travail est modifié dans le Dockerfile ou avec le `WorkingDirectory` paramètre dans le fichier CloudFormation modèle, au moins une `pathMapping` entrée doit être spécifiée afin que le débogueur puisse mapper avec succès les points d'arrêt définis localement au code exécuté dans le conteneur Lambda. Exemple de formatage pour `pathMappings` dans le `launch.json` fichier : `"pathMappings": [ { "localRoot": " ${workspaceFolder}/sam-app/HelloWorldFunction ", "remoteRoot": " /var/task " } ]` Mises en garde : • Pour les fonctions Lambda basées sur des images .NET, `remoteRoot` l'entrée doit être le répertoire de construction. • Pour les fonctions Lambda basées sur Node.js, vous ne pouvez spécifier qu'une seule entrée de mappage de chemin.

Le Toolkit for VS Code utilise la AWS SAM CLI pour créer et déboguer des applications sans serveur localement. Vous pouvez configurer le comportement des commandes AWS SAM CLI à l'aide des propriétés de `"sam"` configuration du `launch.json` fichier.

AWS SAM Propriétés de la CLI (**`"sam"`**)

Propriété	Description	Valeur par défaut
`buildArguments`	Configure la façon dont la commande `sam build` génère le code source Lambda. Pour afficher les options de génération, consultez Fénération sam dans le guide du développeur AWS Serverless Application Model .	Chaîne vide
`containerBuild`	Indique s'il faut créer votre fonction dans un conteneur Docker de type Lambda.	`false`
`dockerNetwork`	Spécifie le nom ou l'ID d'un réseau Docker existant auquel les conteneurs Docker Lambda doivent se connecter , avec le réseau de pont par défaut. Si cela n'est pas spécifié, les conteneur s Lambda se connectent uniquement au réseau Docker de pont par défaut.	Chaîne vide
`localArguments`	Spécifie des arguments d'appel locaux supplémen taires.	Chaîne vide
`skipNewImageCheck`	Indique si la commande doit ignorer l'extraction de la dernière image Docker pour l'exécution de Lambda.	`false`

Propriété	Description	Valeur par défaut
`template`	Personnalise votre AWS SAM modèle à l'aide de paramètre s pour saisir les valeurs des clients. Pour plus d'informa tions, consultez Paramètres dans le Guide de l'utilisateur AWS CloudFormation .	`"parameters":{}`

AWS propriétés de connexion (**`"aws"`**)

Propriété	Description	Valeur par défaut
`credentials`	Sélectionne un profil spécifiqu e (par exemple,`profile:d efault`) dans votre fichier d'informations d'identif ication pour obtenir des AWS informations d'identification.	Les AWS informations d'identif ication que votre fichier de AWS configuration partagé ou votre fichier AWS d'inf ormations d'identification partagé existant fournissent au Toolkit for VS Code.
`region`	Définit la AWS région du service (par exemple, us-east-1).	AWS Région par défaut associée au profil d'identif ication actif.

Exemple : configuration de lancement du modèle

Voici un exemple de fichier de configuration de lancement pour un AWS SAM modèle de cible :

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

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

Exemple : configuration de lancement de code

Voici un exemple de fichier de configuration de lancement pour une cible de fonction Lambda :

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

Résolution des problèmes liés aux applications sans serveur

Cette rubrique décrit les erreurs courantes que vous pouvez rencontrer lors de la création d'applications sans serveur avec le Toolkit for VS Code et explique comment les résoudre.

Rubriques

- Comment puis-je utiliser un fichier samconfig.toml avec une configuration de lancement SAM ?

- Erreur : « RuntimeError : Le conteneur n'existe pas »
- Erreur : « docker.errors ». APIError: 500 Erreur du serveur... Vous avez atteint votre limite de taux d'attraction. »
- Erreur : « Erreur du serveur 500 : montage de C:\Users \... »
- Avec WSL, les vues Web (par exemple, le formulaire « Invoke on AWS ») sont interrompues
- Déboguer une TypeScript application, mais les points d'arrêt ne fonctionnent pas

Comment puis-je utiliser un fichier samconfig.toml avec une configuration de lancement SAM ?

Spécifiez l'emplacement de votre CLI SAM samconfig.toml en configurant l'`--config-file`argument dans la `sam.localArguments` propriété de votre configuration de lancement. Par exemple, si le fichier samconfig.toml se trouve au niveau supérieur de votre espace de travail :

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

Erreur : « RuntimeError : Le conteneur n'existe pas »

La `sam build` commande peut afficher cette erreur si votre système ne dispose pas de suffisamment d'espace disque pour le conteneur Docker. Si le stockage de votre système ne dispose que de 1 à 2 Go d'espace disponible, `sam build` cela peut échouer pendant le traitement, même si le stockage système n'est pas complètement plein avant le début de la compilation. Pour plus d'informations, consultez ce GitHub numéro.

Erreur : « docker.errors ». APIError: 500 Erreur du serveur... Vous avez atteint votre limite de taux d'attraction. »

Docker Hub limite les demandes que les utilisateurs anonymes peuvent faire. Si votre système atteint la limite, Docker échoue et cette erreur apparaît dans la vue OUTPUT de VS Code :

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

Assurez-vous que le service Docker de votre système s'est authentifié avec vos informations d'identification Docker Hub.

Erreur : « Erreur du serveur 500 : montage de C:\Users \... »

Les utilisateurs de Windows peuvent voir cette erreur de montage Docker lors du débogage AWS SAM d'applications :

```
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 ...
```

Essayez d'actualiser les informations d'identification de vos disques partagés (dans les paramètres Docker).

Avec WSL, les vues Web (par exemple, le formulaire « Invoke on AWS ») sont interrompues

Il s'agit d'un problème de code VS connu pour les utilisateurs du VPN Cisco. Pour plus d'informations, consultez ce GitHub numéro.

Une solution de contournement est suggérée dans ce problème de suivi WSL.

Déboguer une TypeScript application, mais les points d'arrêt ne fonctionnent pas

Cela se produira s'il n'existe pas de carte source pour lier le JavaScript fichier compilé au TypeScript fichier source. Pour corriger cela, ouvrez votre `tsconfig.json` fichier et assurez-vous que l'option et la valeur suivantes sont définies : `"inlineSourceMap": true`

Utilisation des documents d'automatisation de Systems Manager

AWS Systems Manager vous donne la visibilité et le contrôle de votre infrastructure sur AWS. Systems Manager fournit une interface utilisateur unifiée qui vous permet de consulter les données opérationnelles de plusieurs AWS services et d'automatiser les tâches opérationnelles sur l'ensemble de vos AWS ressources.

Un document de Systems Manager définit les actions que Systems Manager effectue sur vos instances gérées. Un document d'automatisation est un type de document Systems Manager que

vous utilisez pour effectuer des tâches de maintenance et de déploiement courantes, telles que la création ou la mise à jour d'une Amazon Machine Image (AMI). Cette rubrique explique comment créer, modifier, publier et supprimer des documents Automation avec AWS Toolkit for Visual Studio Code.

Rubriques

- Hypothèses et prérequis
- Autorisations IAM pour les documents d'automatisation de Systems Manager
- Création d'un nouveau document d'automatisation de Systems Manager
- Ouverture d'un document Systems Manager Automation existant
- Modification d'un document d'automatisation de Systems Manager
- Publication d'un document sur l'automatisation de Systems Manager
- Supprimer un document d'automatisation de Systems Manager
- Exécution d'un document d'automatisation de Systems Manager
- Résolution des problèmes liés aux documents d'automatisation de Systems Manager dans Toolkit for VS Code

Hypothèses et prérequis

Avant de commencer, assurez-vous que :

- Vous avez installé Visual Studio Code et la dernière version de AWS Toolkit for Visual Studio Code. Pour plus d'informations, consultez Installation du AWS Toolkit for Visual Studio Code.
- Vous connaissez Systems Manager. Pour plus d'informations, consultez le Guide de l'utilisateur AWS Systems Manager .
- Vous connaissez les cas d'utilisation de Systems Manager Automation. Pour plus d'informations, consultez Automatisation de Systems Manager AWS dans le Guide de l'utilisateur AWS Systems Manager .

Autorisations IAM pour les documents d'automatisation de Systems Manager

Dans le Toolkit for VS Code, vous devez disposer d'un profil d'informations d'identification contenant les autorisations Gestion des identités et des accès AWS (IAM) nécessaires pour créer, modifier,

publier et supprimer des documents Systems Manager Automation. Le document de politique suivant définit les autorisations IAM nécessaires pouvant être utilisées dans une politique principale :

JSON

```
{
    "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": "*"
        }
    ]
}
```

Pour des informations sur la création de politiques IAM, consultez Création de politiques dans le Guide de l'utilisateur IAM. Pour plus d'informations sur la configuration de votre profil d'informations d'identification, consultezAWS Informations d'identification IAM.

Création d'un nouveau document d'automatisation de Systems Manager

Vous pouvez créer un nouveau document d'automatisation dans `JSON` ou à `YAML` l'aide de Visual Studio Code. Lorsque vous créez un nouveau document d'automatisation, il est présenté dans un fichier sans titre. Vous pouvez nommer votre fichier et l'enregistrer dans VS Code, mais le nom du fichier n'est pas visible pour AWS.

Pour créer un nouveau document d'automatisation

1. Ouvrez VS Code.

2. Dans le menu Affichage, choisissez Palette de commandes pour ouvrir la Palette de commandes.
3. Dans la palette de commandes, saisissez AWS Toolkit Create a new Systems Manager Document Locally.
4. Choisissez l'un des modèles de démarrage pour un exemple Hello World.
5. Choisissez `JSON` ou `YAML`.

   Un nouveau document d'automatisation est créé.

> **Note**
>
> Votre nouveau document d'automatisation dans VS Code n'apparaît pas automatiquement dans AWS. Vous devez le publier AWS pour pouvoir l'exécuter.

Ouverture d'un document Systems Manager Automation existant

Vous utilisez l' AWS explorateur pour rechercher des documents Systems Manager Automation existants. Lorsque vous ouvrez un document Automation existant, il apparaît sous la forme d'un fichier sans titre dans VS Code.

Pour ouvrir votre document d'automatisation

1. Ouvrez VS Code.
2. Dans la barre de navigation de gauche, choisissez AWS d'ouvrir l' AWS explorateur.
3. Dans l' AWS Explorateur, pour Systems Manager, choisissez l'icône de téléchargement sur le document que vous souhaitez ouvrir, puis choisissez la version du document. Le fichier s'ouvrira au format correspondant à cette version. Sinon, choisissez Télécharger au format JSON ou Télécharger au format YAML.

> **Note**
>
> L'enregistrement local d'un document Automation en tant que fichier dans VS Code ne le fait pas apparaître dans AWS. Il doit être publié AWS avant d'être exécuté.

Modification d'un document d'automatisation de Systems Manager

Si vous possédez des documents d'automatisation, ils apparaissent dans la catégorie Owned by Me des documents Systems Manager de l' AWS Explorateur. Vous pouvez être propriétaire de documents Automation qui existent déjà dans AWS, et vous pouvez posséder des documents nouveaux ou mis à jour sur lesquels vous avez précédemment publié AWS depuis VS Code.

Lorsque vous ouvrez un document d'automatisation pour le modifier dans VS Code, vous pouvez en faire plus que dans le Console de gestion AWS. Par exemple :

- Il existe une validation de schéma sur `JSON` les deux `YAML` formats.
- Des extraits de code sont disponibles dans l'éditeur de documents pour vous permettre de créer n'importe quel type d'étape d'automatisation.
- Il existe une prise en charge de la saisie semi-automatique pour diverses options dans `JSON` et`YAML`.

Utilisation des versions

Les documents d'automatisation de Systems Manager utilisent des versions pour la gestion des modifications. Vous pouvez choisir la version par défaut d'un document d'automatisation dans VS Code.

Pour définir une version par défaut

- Dans l' AWS Explorateur, accédez au document pour lequel vous souhaitez définir la version par défaut, ouvrez le menu contextuel (clic droit) du document, puis choisissez Définir la version par défaut.

> **Note**
>
> Si le document choisi ne comporte qu'une version, vous ne pourrez pas modifier la version par défaut.

Publication d'un document sur l'automatisation de Systems Manager

Après avoir modifié votre document d'automatisation dans VS Code, vous pouvez le publier sur AWS.

Pour publier votre document d'automatisation

1. Ouvrez le document d'automatisation que vous souhaitez publier en suivant la procédure décrite dansOuverture d'un document Systems Manager Automation existant.
2. Apportez les modifications que vous souhaitez publier. Pour plus d'informations, consultez Modification d'un document d'automatisation de Systems Manager.
3. Dans le coin supérieur droit du fichier ouvert, choisissez l'icône de téléchargement.
4. Dans la boîte de dialogue du flux de publication, choisissez la AWS région dans laquelle vous souhaitez publier le document Automation.
5. Si vous publiez un nouveau document, choisissez Création rapide. Sinon, choisissez Mise à jour rapide pour mettre à jour un document d'automatisation existant dans cette AWS région.
6. Entrez le nom de ce document d'automatisation.

Lorsque vous publiez une mise à jour d'un document Automation existant dans AWS, une nouvelle version est ajoutée au document.

Supprimer un document d'automatisation de Systems Manager

Vous pouvez supprimer des documents d'automatisation dans VS Code. La suppression d'un document d'automatisation supprime le document et toutes ses versions.

> **Important**
>
> - La suppression est une action destructrice qui ne peut être annulée.
> - La suppression d'un document d'automatisation qui a déjà été exécuté ne supprime pas les AWS ressources qui ont été créées ou modifiées lors de son démarrage.

Pour supprimer votre document d'automatisation

1. Ouvrez VS Code.
2. Dans la barre de navigation de gauche, choisissez AWS d'ouvrir l' AWS explorateur.
3. Dans l' AWS Explorateur, pour Systems Manager, ouvrez le menu contextuel (clic droit) du document que vous souhaitez supprimer, puis choisissez Supprimer le document.

Exécution d'un document d'automatisation de Systems Manager

Une fois votre document Automation publié AWS, vous pouvez l'exécuter pour effectuer des tâches en votre nom sur votre AWS compte. Pour exécuter votre document d'automatisation, vous utilisez les Console de gestion AWS API Systems Manager, le AWS CLI, ou le Outils AWS pour PowerShell. Pour obtenir des instructions sur la façon d'exécuter un document d'automatisation, consultez la section Exécution d'une automatisation simple dans le Guide de AWS Systems Manager l'utilisateur.

Sinon, si vous souhaitez utiliser l'un des AWS kits SDK avec les API Systems Manager pour exécuter votre document d'automatisation, consultez les références du AWS SDK.

> **Note**
>
> L'exécution d'un document d'automatisation peut créer de nouvelles ressources AWS et entraîner des frais de facturation. Nous vous recommandons vivement de comprendre ce que votre document d'automatisation créera dans votre compte avant de le créer.

Résolution des problèmes liés aux documents d'automatisation de Systems Manager dans Toolkit for VS Code

J'ai enregistré mon document d'automatisation dans VS Code, mais je ne le vois pas dans Console de gestion AWS.

L'enregistrement d'un document d'automatisation dans VS Code ne publie pas le document d'automatisation dans AWS. Pour plus d'informations sur la publication de votre document d'automatisation, consultez Publication d'un document sur l'automatisation de Systems Manager.

La publication de mon document Automation a échoué en raison d'une erreur d'autorisations.

Assurez-vous que votre profil AWS d'informations d'identification dispose des autorisations nécessaires pour publier des documents d'automatisation. Pour obtenir un exemple de politique d'autorisations, consultez Autorisations IAM pour les documents d'automatisation de Systems Manager.

J'ai publié mon document d'automatisation sur AWS, mais je ne le vois pas dans Console de gestion AWS.

Assurez-vous d'avoir publié le document dans la même AWS région que celle dans laquelle vous naviguez Console de gestion AWS.

J'ai supprimé mon document d'automatisation, mais les ressources qu'il a créées me sont toujours facturées.

La suppression d'un document d'automatisation ne supprime pas les ressources qu'il a créées ou modifiées. Vous pouvez identifier les AWS ressources que vous avez créées à partir de la console de gestion de la AWS facturation, explorer vos frais et choisir les ressources à supprimer à partir de cette console.

AWS Step Functions

Avec AWS Step Functions, vous pouvez créer des flux de travail (également appelés machines à états) pour créer des applications distribuées, automatiser des processus, orchestrer des microservices et créer des pipelines de données et d'apprentissage automatique. Les rubriques suivantes décrivent comment utiliser AWS Step Functions dans le AWS Toolkit for Visual Studio Code. Pour obtenir des informations détaillées sur le AWS Step Functions service, consultez le Guide du AWS Step Functions développeur.

Rubriques

- Travailler avec AWS Step Functions
- Travailler avec AWS Step Functions Workflow Studio

Travailler avec AWS Step Functions

Les sections suivantes décrivent comment utiliser des AWS Step Functions `Amazon State Language (ASL)` fichiers contenant des définitions de machine à états dans le AWS Toolkit. Pour des informations détaillées sur les machines à AWS Step Functions états, consultez la rubrique En savoir plus sur les machines à états dans Step Functions du Guide du AWS Step Functions développeur.

Affichage des machines à états Step Functions

Pour afficher vos `ASL` fichiers existants contenant des définitions de machine à états dans l'explorateur AWS Toolkit, procédez comme suit.

1. Dans l'explorateur de la AWS boîte à outils, développez la zone qui contient le `ASL` fichier que vous souhaitez afficher.
2. Développez la rubrique Step Functions.

3. Vos ASL fichiers s'affichent dans l' AWS explorateur.

Création d'une machine à états Step Functions

Dans la AWS boîte à outils, vous pouvez créer une nouvelle machine à états Step Functions à partir d'un fichier ou utiliser un modèle. La procédure suivante décrit comment créer une machine d'état Step Functions à partir d'un fichier. Pour plus de détails sur la création d'une machine à états SFN ; à partir d'un modèle, consultez la section Modèles de machine à états située ci-dessous, dans cette rubrique du Guide de l'utilisateur.

> **Note**
>
> Pour utiliser Step Functions dans VS Code, l'extension de votre fichier `Amazon State Language` (ASL) contenant la définition de votre machine à états doit se terminer par `asl.jsonasl.yml`, ou `.asl.yaml`.
> Par défaut, les fichiers Step Functions pertinents s'ouvrent dans Workflow Studio. Pour obtenir des informations détaillées sur l'utilisation de Workflow Studio via la AWS boîte à outils, consultez la rubrique Utilisation de Workflow Studio dans ce guide de l'utilisateur.

1. Depuis votre espace de travail dans VS Code, créez un nouveau fichier.
2. Donnez un nom à votre fichier et spécifiez son extension comme `asl.json` suit :`asl.yml`, ou `.asl.yaml`.
3. Lors de la création, le AWS Toolkit ouvre le nouveau fichier dans AWS Step Functions Workflow Studio.
4. Dans Workflow Studio, cliquez sur le bouton Enregistrer dans le menu utilitaire pour enregistrer votre nouveau ASL fichier.

Création d'une machine à états Step Functions à partir d'un modèle

Dans la AWS boîte à outils, vous pouvez créer une machine à états Step Functions à partir d'un modèle. Le processus de modèle crée un ASL fichier contenant une définition de machine à états, fournissant un point de départ pour votre projet. La procédure suivante décrit comment créer une machine à états Step Functions à partir d'un modèle du AWS Toolkit.

1. Dans l'explorateur de AWS boîte à outils, développez la région dans laquelle vous souhaitez créer une machine à états Step Functions.

2. Ouvrez le menu contextuel de Step Functions (cliquez avec le bouton droit de la souris) et choisissez Créer une nouvelle machine à états Step Functions pour ouvrir l'assistant de sélection d'un modèle de démarrage (1/2) dans VS Code.
3. Dans l'assistant Sélectionnez un modèle de démarrage (1/2), choisissez le type de modèle pour votre machine à états Step Functions pour continuer.
4. Dans l'écran Sélectionner le format du modèle (2/2), choisissez YAML ou JSON pour le format de votre modèle.
5. Un nouveau ASL fichier contenant la définition de votre machine à états s'ouvre dans l'éditeur VS Code.

Téléchargement d'une machine à états Step Functions

Pour télécharger une machine d'état Step Functions stockée à distance sur votre instance locale de VS Code, procédez comme suit.

1. Dans l'explorateur de AWS boîte à outils, développez la région contenant la machine à états Step Functions que vous souhaitez télécharger.
2. Développez Step Functions, puis cliquez avec le bouton droit sur la machine d'état Step Functions que vous souhaitez télécharger et choisissez Télécharger la définition... .
3. Spécifiez un emplacement pour stocker localement votre machine d'état Step Functions pour continuer.
4. La machine d'état Step Functions s'ouvre dans Workflow Studio une fois la procédure terminée.

Enregistrement des modifications sur une machine à états Step Functions

La procédure suivante décrit comment enregistrer les modifications apportées à votre machine à états Step Functions.

> **Note**
>
> Les modifications effectuées dans Workflow Studio sont synchronisées avec votre fichier local, mais ne sont pas enregistrées tant que votre travail n'est pas enregistré dans l'éditeur VS Code ou Workflow Studio. Si votre fichier local est modifié et enregistré alors que Workflow Studio est ouvert et qu'aucune erreur n'est détectée dans votre ASL fichier, vous recevez une notification de réussite dans Workflow Studio une fois l'enregistrement terminé.

Toutefois, si le contenu de votre fichier local n'est pas valide `JSON` ou `YAML` si vous tentez de l'enregistrer, la synchronisation de votre fichier local échoue et vous recevez une notification d'avertissement dans Workflow Studio.

1. À partir d'un `ASL` fichier ouvert contenant une définition de machine à états dans Workflow Studio, accédez aux boutons Utilitaires.
2. Choisissez le bouton Enregistrer.
3. VS Code vous avertit lorsque le fichier a été enregistré.

Exécution d'une machine à états Step Functions

La procédure suivante décrit comment exécuter une machine d'état Step Functions dans le AWS Toolkit.

1. Dans l'explorateur de AWS boîte à outils, développez la zone contenant la machine d'état Step Functions que vous souhaitez exécuter.
2. Développez Step Functions, puis cliquez avec le bouton droit sur la machine d'état Step Functions que vous souhaitez exécuter.
3. Dans le menu contextuel, choisissez Démarrer l'exécution pour lancer le processus de lancement.
4. L'état du lancement est affiché dans la fenêtre de sortie du AWS Toolkit dans VS Code.

Utilisation d'extraits de code

Les extraits de code sont des suggestions automatisées qui sont générées en fonction du code sur lequel vous travaillez. Pour utiliser des extraits de code avec Step Functions dans la boîte à outils, procédez comme suit.

> **Note**
>
> Pour utiliser des extraits de code Step Functions dans VS Code, l'extension de votre `ASL` fichier contenant la définition de votre machine à états doit se terminer par `.asl.json`.`asl.yml`, ou. `.asl.yaml`
> Par défaut, vos fichiers Step Functions pertinents s'ouvrent dans Workflow Studio.

1. Dans VS Code, ouvrez un ASL fichier contenant la définition de machine à états que vous souhaitez modifier ou créez un nouveau ASL fichier.
2. Dans Workflow Studio, passez en mode Code si vous êtes en mode Conception.
3. Depuis l'éditeur de code Workflow Studio, placez votre curseur dans la `"States"` propriété.
4. Appuyez **`control + space`** pour ouvrir le menu des extraits de code. Des propriétés supplémentaires sont accessibles en appuyant sur **`control + space`** et sont basées sur. `"State"` `"Type"`
5. Choisissez l'extrait de code que vous souhaitez dans la liste.

Validation du code

Lorsque vous travaillez sur Step Functions dans Workflow Studio, la validation du code identifie activement les erreurs et propose les solutions suivantes :

- Propriétés manquantes
- Valeurs incorrectes
- État non terminal
- États pointés non existants

Travailler avec AWS Step Functions Workflow Studio

Les sections suivantes décrivent comment utiliser AWS Step Functions Workflow Studio dans le AWS Toolkit for Visual Studio Code. Pour des informations détaillées sur AWS Step Functions Workflow Studio, consultez la rubrique Développement de flux de travail dans le Guide du AWS Step Functions développeur

Ouverture de Workflow Studio

La liste suivante décrit les différents chemins disponibles pour ouvrir Workflow Studio dans VS Code.

> **Note**
>
> Pour utiliser Workflow Studio dans VS Code, l'extension de votre fichier `Amazon State Language` (ASL) qui contient la définition de votre machine à états doit se terminer par `asl.json` `asl.yml` ou`asl.yaml`. Pour plus de détails sur le téléchargement ou la création

d'une nouvelle définition de machine à états dans la AWS boîte à outils, consultez les sections Téléchargement de machines à états et Création d'une machine à états de la AWS Step Functions rubrique Utilisation de ce guide de l'utilisateur.

- Dans l' AWS Explorateur, ouvrez le menu contextuel (cliquez avec le bouton droit de la souris) sur un ASL fichier contenant une définition de machine à états, puis choisissez Ouvrir dans Workflow Studio.
- Dans un ASL fichier ouvert contenant une définition de machine à états, choisissez l'icône Ouvrir avec Workflow Studio située à côté des onglets de la fenêtre de l'éditeur VS Code.
- Dans un ASL fichier ouvert contenant une définition de machine à états, choisissez la CodeLens commande Ouvrir avec Workflow Studio, située en haut du fichier.
- La fermeture et la réouverture d'un ASL fichier contenant une définition de machine à états rouvrent automatiquement le fichier dans Workflow Studio, sauf si le Workflow Studio par défaut est désactivé manuellement.

Mode conception et mode code

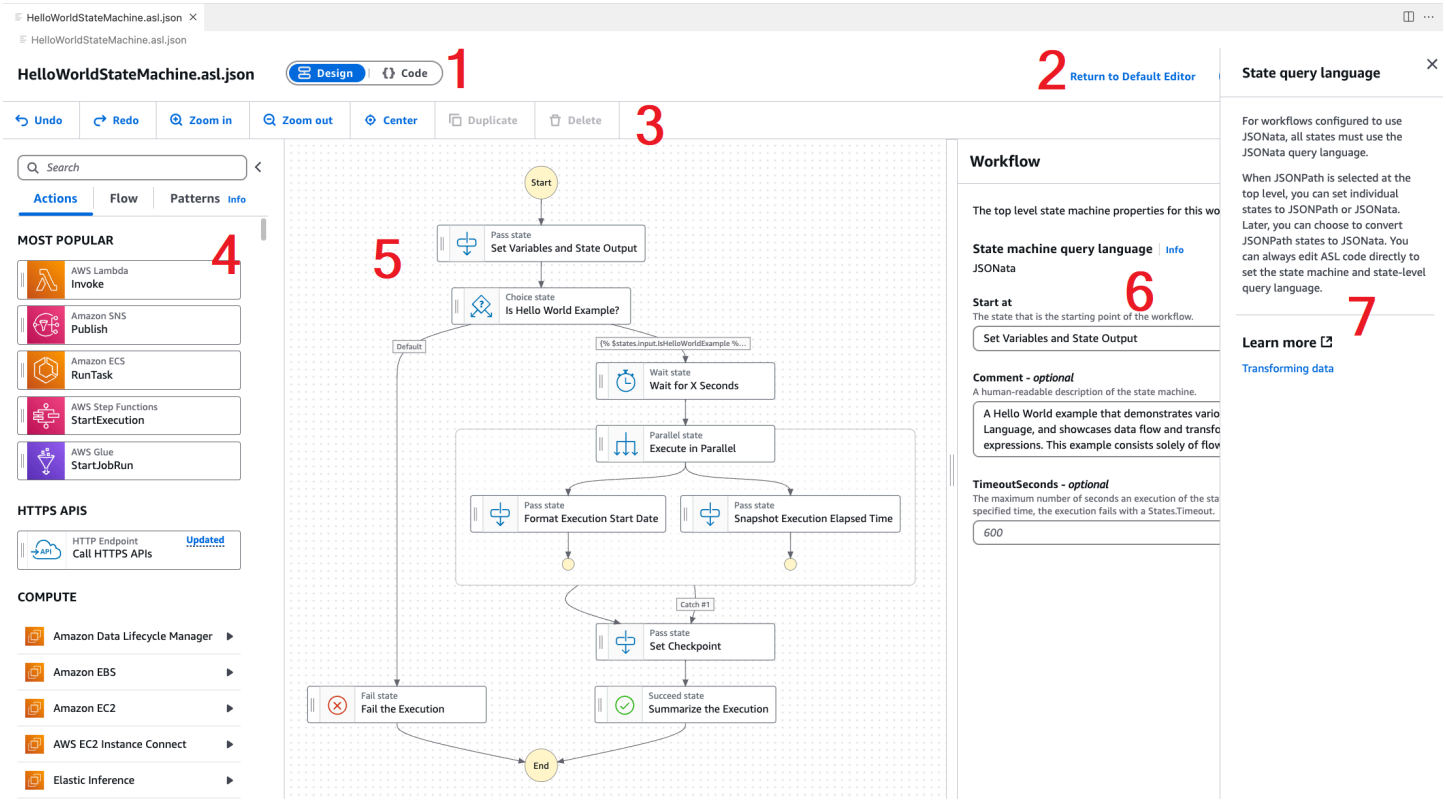
Workflow Studio propose deux modes pour travailler avec vos ASL fichiers contenant une définition de machine à états : le mode Conception et le mode Code. Le mode Conception fournit une interface graphique permettant de visualiser vos flux de travail lorsque vous créez des prototypes. Le mode Code dispose d'un éditeur de code intégré dans lequel vous pouvez afficher, écrire et modifier les ASL définitions de vos flux de travail.

> **Note**
>
> Pour obtenir des informations détaillées sur chacune des sections de l'interface utilisateur en mode Conception et en mode Code, consultez la rubrique Utilisation de Workflow Studio dans le Guide du AWS Step Functions développeur. Les fonctionnalités de Workflow Studio ne sont pas toutes disponibles dans le AWS Toolkit, comme le mode Configuration, par exemple.

L'interface utilisateur du mode Conception comporte 7 sections principales, comme indiqué et décrit dans l'image suivante.

1. Boutons de mode : boutons permettant de basculer entre les modes Design et Code.

2. Boutons utilitaires : ensemble de boutons permettant d'effectuer des tâches, telles que quitter Workflow Studio, enregistrer vos flux de travail ou exporter des ASL définitions dans un fichier JSON ou YAML.
3. Barre d'outils de conception : barre d'outils contenant un ensemble de boutons permettant d'effectuer des actions courantes, telles que l'annulation, la suppression et le contrôle du zoom.
4. Navigateur d'états : navigateur contenant des états par glisser-déposer pour votre canevas de flux de travail. Les états sont organisés en onglets et définis comme Actions , Flux et Modèles.
5. Le canevas et le graphique du flux de travail : rendu visuel de votre flux de travail dans lequel vous pouvez supprimer, réorganiser et sélectionner des états pour la configuration.
6. Panneau d'inspection : permet d'afficher et de modifier les propriétés de tout état sélectionné sur le canevas. En fonction de l'état sélectionné dans le graphique du flux de travail du canevas, les onglets sont remplis d'options spécifiques à l'état pour la configuration Input/Output , les variables et la gestion des erreurs.
7. Liens d'information : Ouvre un panneau contenant des informations contextuelles lorsque vous avez besoin d'aide. Ces panneaux incluent également des liens vers des sujets connexes dans le Guide du AWS Step Functions développeur.



Utilisation de tests à état unique lors de la conception

À partir de l'interface utilisateur d'état de test de Workflow Studio, vous pouvez tester les états individuels de votre machine à états. Cela inclut la possibilité de fournir des entrées d'état, de définir des variables et d'effectuer à la fois des substitutions AWS SAM et des CloudFormation définitions.

Pour en savoir plus sur l'infrastructure en tant que code (IaC), les définitions de ressources et la transformation des données, consultez les rubriques Utilisation AWS SAM pour créer des flux de travail Step Functions et Transformer des données avec JSonata dans Step Functions dans le Guide du AWS Step Functions développeur.

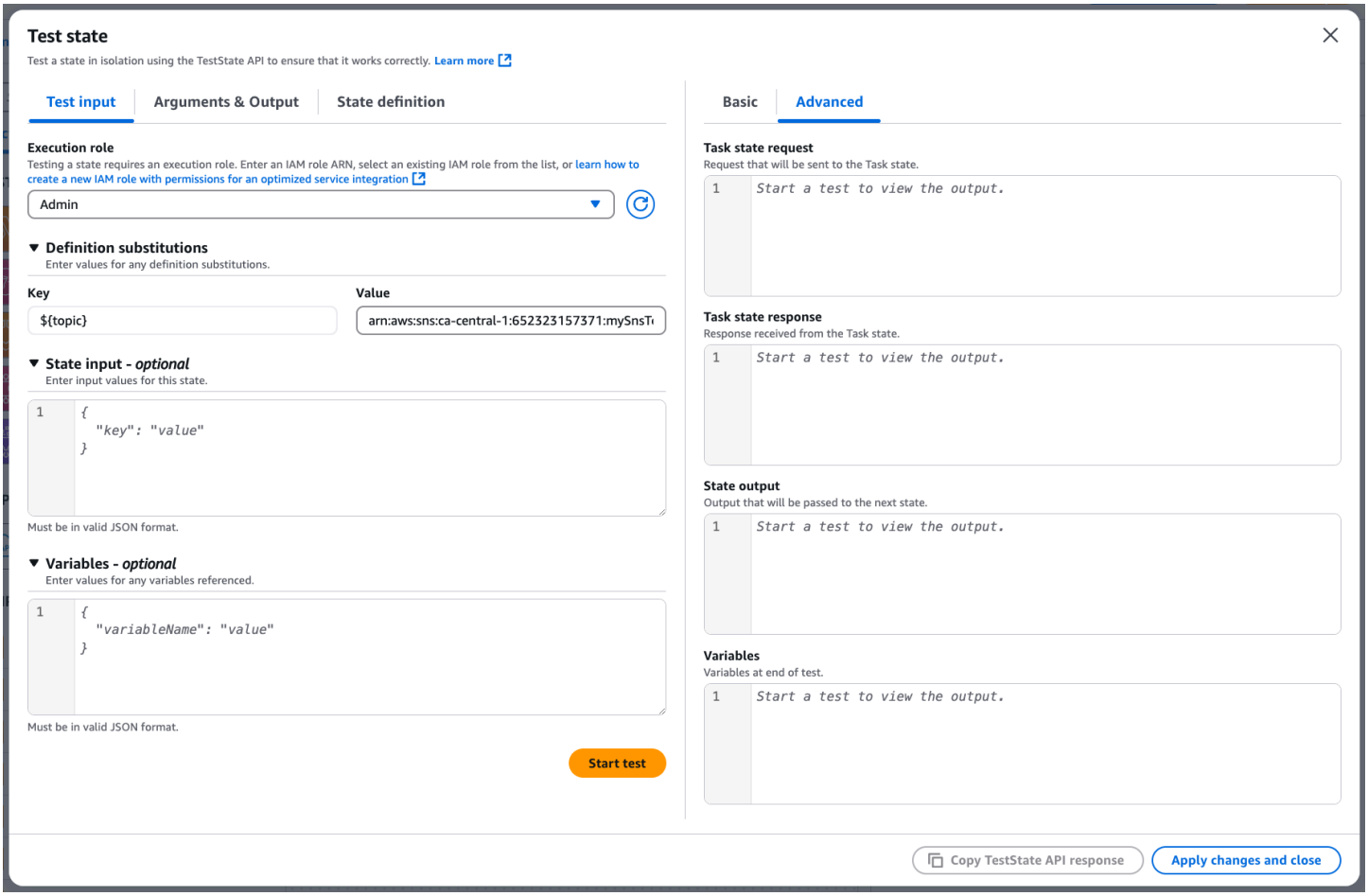
La procédure suivante décrit comment ouvrir l'interface utilisateur de l'état de test dans Workflow Studio.

Ouverture de l'interface utilisateur de l'état du test

1. Dans l'onglet Mode conception de Workflow Studio, accédez au canevas et choisissez un état pour l'ouvrir dans le panneau Inspecteur.
2. Dans le panneau Inspecteur, cliquez sur le bouton État du test.
3. L'interface utilisateur de l'état du test s'ouvre dans VS Code.

L'interface utilisateur de l'état de test comporte 3 onglets principaux, Entrée de test, Arguments et sortie, Définition de l'état. L'onglet Entrée de test comporte 3 champs supplémentaires qui vous permettent de fournir une entrée d'état, de définir des variables et de spécifier des substitutions de définitions à partir de vos CloudFormation modèles AWS SAM ou modèles. Dans l'onglet Définition de l'état, vous pouvez ajuster le flux de travail et effectuer un nouveau test. Lorsque vous avez terminé d'exécuter les tests, vous pouvez appliquer et enregistrer les modifications apportées à la définition de votre machine à états.

La capture d'écran suivante montre l'interface utilisateur de l'état du test, qui inclut une définition des ressources thématiques.



Désactivation de Workflow Studio par défaut

Par défaut, Workflow Studio est l'éditeur par défaut pour ASL les fichiers contenant une définition de machine à états. Vous pouvez désactiver le paramètre par défaut en modifiant votre `settings.json` fichier dans votre `.vscode` répertoire local. Si vous désactivez Workflow Studio par défaut, il est toujours accessible via les méthodes répertoriées dans la section Ouverture de Workflow Studio, située dans cette rubrique.

Pour modifier votre `settings.json` fichier depuis VS Code, procédez comme suit.

1. Dans VS Code, ouvrez la palette de commandes en appuyant sur **`option+shift+p`** (Mac) ou **`ctrl+shift+p`** (Windows).
2. Dans la palette de commandes VS Code, entrez **`Open User Settings (JSON)`** dans le champ de recherche et choisissez l'option lorsqu'elle apparaît dans la liste.
3. `settings.json`Dans votre éditeur, ajoutez la modification suivante à votre fichier.

```
                {
                    "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. Enregistrez vos modifications `settings.json` et actualisez ou redémarrez VS Code.

Utilisation de Threat Composer

Vous pouvez utiliser le AWS Toolkit for Visual Studio Code pour utiliser l'outil Threat Composer. Threat Composer est un outil de modélisation des menaces qui peut simplifier votre processus de modélisation des menaces.

Pour des informations détaillées sur l'outil Threat Composer, consultez le GitHub référentiel Threat Composer.

Les rubriques suivantes décrivent comment utiliser Threat Composer dans le AWS Toolkit for Visual Studio Code.

Rubriques

- Utilisation de Threat Composer depuis le Toolkit

Utilisation de Threat Composer depuis le Toolkit

Avec Threat Composer, vous pouvez créer, afficher et modifier les modèles de menace de Threat Composer directement dans VS Code. Pour des informations détaillées sur l'outil Threat Composer, consultez le GitHub référentiel Threat Composer.

Les sections suivantes décrivent comment accéder aux outils Threat Composer dans le AWS Toolkit for Visual Studio Code.

Accès à Threat Composer depuis le Toolkit

Il existe 3 manières principales d'accéder à Threat Composer à partir du Toolkit.

Accès à Threat Composer via un modèle de menace existant

Pour ouvrir Threat Composer, ouvrez un fichier de modèle de menace (extension `.tc.json`) existant dans VS Code. Threat Composer ouvre et affiche automatiquement une visualisation de votre fichier de modèle de menace dans la fenêtre de l'éditeur VS Code.

Création d'un nouveau modèle de menace Threat Composer

1. Dans le menu principal de VS Code, développez le fichier, puis choisissez Nouveau fichier.
2. Dans la boîte de dialogue Nouveau fichier, choisissez Threat Composer File... .
3. Lorsque vous y êtes invité, entrez un`file name`, puis appuyez sur la **`enter`** touche pour ouvrir Threat Composer et créer une visualisation de votre fichier de modèle de menace vide dans une nouvelle fenêtre de l'éditeur VS Code.

Création d'un nouveau modèle de menace Threat Composer à partir de la palette de commandes

1. Dans VS Code, ouvrez la palette de commandes en appuyant sur **`Cmd + Shift + P`** ou **`Ctrl + Shift + P`** (Windows).
2. Dans le champ de recherche, entrez **`Threat Composer`** et choisissez Create New Threat Composer File lorsque les résultats sont renseignés.
3. Lorsque vous y êtes invité, entrez un`file name`, puis appuyez sur la **`enter`** touche pour ouvrir Threat Composer et créer une visualisation de votre fichier de modèle de menace vide dans une nouvelle fenêtre de l'éditeur VS Code.

Utilisation des ressources

En plus d'accéder aux AWS services répertoriés par défaut dans l' AWS Explorateur, vous pouvez également accéder à Ressources et choisir parmi des centaines de ressources à ajouter à l'interface. Dans AWS, une ressource est une entité avec laquelle vous pouvez travailler. Parmi les ressources qui peuvent être ajoutées, citons Amazon AppFlow, Amazon Kinesis Data Streams, les rôles AWS IAM, Amazon VPC et les distributions Amazon. CloudFront

Après avoir fait votre sélection, vous pouvez aller dans Ressources et développer le type de ressource pour lister les ressources disponibles pour ce type. Par exemple, si vous sélectionnez le type de ressource `AWS Toolkit:Lambda::Function`, vous pouvez accéder aux ressources qui définissent différentes fonctions, leurs propriétés et leurs attributs.

Après avoir ajouté un type de ressource à Resources (Ressources), vous pouvez interagir avec lui et ses ressources de la manière suivante :

- Afficher la liste des ressources existantes disponibles dans la AWS région actuelle pour ce type de ressource.
- Afficher une version en lecture seule du fichier JSON qui décrit une ressource.
- Copier l'identifiant de la ressource.
- Consultez la AWS documentation qui explique l'objectif du type de ressource et le schéma (aux formats JSON et YAML) pour modéliser une ressource.
- Créez une nouvelle ressource en modifiant et en enregistrant un JSON-formatted modèle conforme à un schéma. *
- Mettre à jour ou supprimer une ressource existante.*

> **Important**
>
> * Dans la version actuelle, AWS Toolkit for Visual Studio Code l'option permettant de créer, modifier et supprimer des ressources est une fonctionnalité expérimentale. Comme les fonctionnalités expérimentales continuent d'être testées et mises à jour, elles peuvent présenter des problèmes d'utilisation. Et les fonctionnalités expérimentales peuvent être supprimées du AWS Toolkit for Visual Studio Code sans préavis.
> Pour autoriser l'utilisation de fonctionnalités expérimentales pour les ressources, ouvrez le volet Paramètres de votre IDE VS Code, développez Extensions et choisissez AWS Toolkit. Dans AWS Toolkit Experiments, sélectionnez json ResourceModification pour vous permettre de créer, de mettre à jour et de supprimer des ressources.
> Pour plus d'informations, consultez Travailler avec des fonctionnalités expérimentales.

Autorisations IAM pour l'accès aux ressources

Vous avez besoin d' Gestion des identités et des accès AWS autorisations spécifiques pour accéder aux ressources associées aux AWS services. Par exemple, une entité IAM, telle qu'un utilisateur ou un rôle, a besoin des autorisations Lambda pour accéder aux ressources `AWS Toolkit:Lambda::Function`.

Outre les autorisations pour les ressources de service, une entité IAM a besoin d'autorisations pour permettre au Toolkit for VS Code d'appeler les opérations de l'API AWS Cloud Control en son nom.

Les opérations d'API Cloud Control permettent à l'utilisateur ou au rôle IAM d'accéder aux ressources distantes et de les mettre à jour.

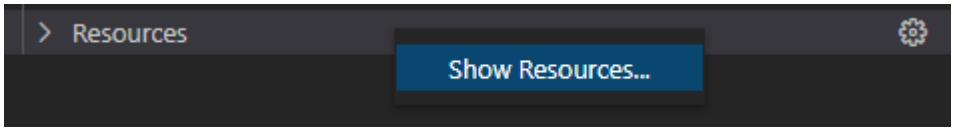
Le moyen le plus simple d'accorder des autorisations consiste à associer la politique AWS gérée à l'entité IAM qui appelle ces opérations d'API à l'aide de l'interface Toolkit. PowerUserAccess Cette politique gérée accorde une série d'autorisations pour effectuer des tâches de développement d'applications, y compris l'appel d'opérations d'API.

Pour connaître les autorisations spécifiques qui définissent les opérations d'API autorisées sur les ressources distantes, consultez le Guide de l'utilisateur de l'API AWS Cloud Control.

Ajout et interaction avec des ressources existantes

1. Dans l'AWS explorateur, cliquez avec le bouton droit sur Ressources et choisissez Afficher les ressources.

   Un volet affiche la liste des types de ressources disponibles pour la sélection.

2. Dans le volet de sélection, sélectionnez les types de ressources à ajouter à l'AWS Explorateur et appuyez sur Retour ou choisissez OK pour confirmer.

   Les types de ressources que vous avez sélectionnés sont répertoriés sous Ressources.

   > **Note**
   >
   > Si vous avez déjà ajouté un type de ressource à AWS Explorer et que vous décochez ensuite la case de ce type, il n'est plus répertorié sous Ressources après que vous avez cliqué sur OK. Seuls les types de ressources actuellement sélectionnés sont visibles dans AWS Explorer.

3. Pour afficher les ressources qui existent déjà pour un type de ressource, développez l'entrée correspondant à ce type.

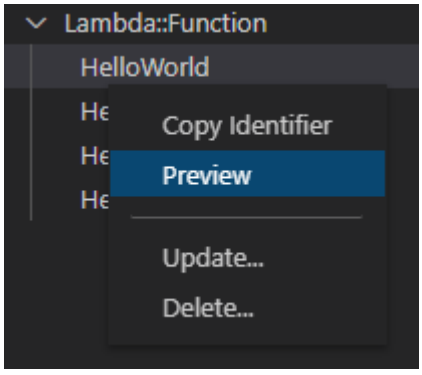
   Une liste des ressources disponibles s'affiche sous son type de ressource.

4. Pour interagir avec une ressource spécifique, cliquez avec le bouton droit de la souris sur son nom et choisissez l'une des options suivantes :

- Copier l'identifiant de la ressource : copiez l'identifiant de la ressource spécifique dans le presse-papiers. (Par exemple, la ressource `AWS Toolkit:DynamoDB::Table` peut être identifiée à l'aide de la propriété `TableName`.)
- Aperçu : affichez une version en lecture seule du JSON-formatted modèle qui décrit la ressource.

  Une fois le modèle de ressource affiché, vous pouvez le modifier en choisissant l'icône Mettre à jour à droite de l'onglet de l'éditeur. Pour mettre à jour une ressource, vous devez ??? activer les paramètres requis.
- Mise à jour : modifiez le JSON-formatted modèle de la ressource dans un éditeur VS Code. Pour plus d'informations, consultez Création et modification de ressources.
- Supprimer : supprimez la ressource en confirmant la suppression dans une boîte de dialogue qui s'affiche. (La suppression de ressources est actuellement ??? disponible dans cette version de AWS Toolkit for Visual Studio Code.)

> **Warning**
>
> Si vous supprimez une ressource, aucune AWS CloudFormation pile qui utilise cette ressource ne sera pas mise à jour. Pour corriger cet échec de mise à jour, vous devez soit recréer la ressource, soit supprimer la référence à celle-ci dans le CloudFormation modèle de la pile. Pour plus d'informations, consultez cet article du Centre de connaissances.



Création et modification de ressources

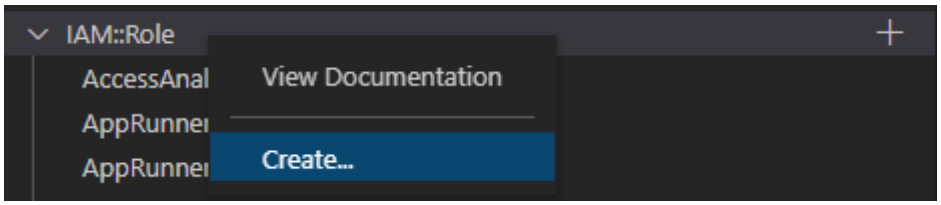
> **Important**
>
> La création et la mise à jour des ressources est actuellement un ??? dans cette version d' AWS Toolkit for Visual Studio Code.

La création d'une nouvelle ressource implique l'ajout d'un type de ressource à la liste des ressources, puis la modification d'un JSON-formatted modèle qui définit la ressource, ses propriétés et ses attributs.

Par exemple, une ressource appartenant au type de `AWS Toolkit:SageMaker::UserProfile` ressource est définie à l'aide d'un modèle qui crée un profil utilisateur pour Amazon SageMaker AI Studio. Le modèle qui définit cette ressource de profil utilisateur doit être conforme au schéma du type de ressource `AWS Toolkit:SageMaker::UserProfile`. Si le modèle n'est pas conforme au schéma en raison de propriétés manquantes ou incorrectes, par exemple, la ressource ne peut pas être créée ou mise à jour.

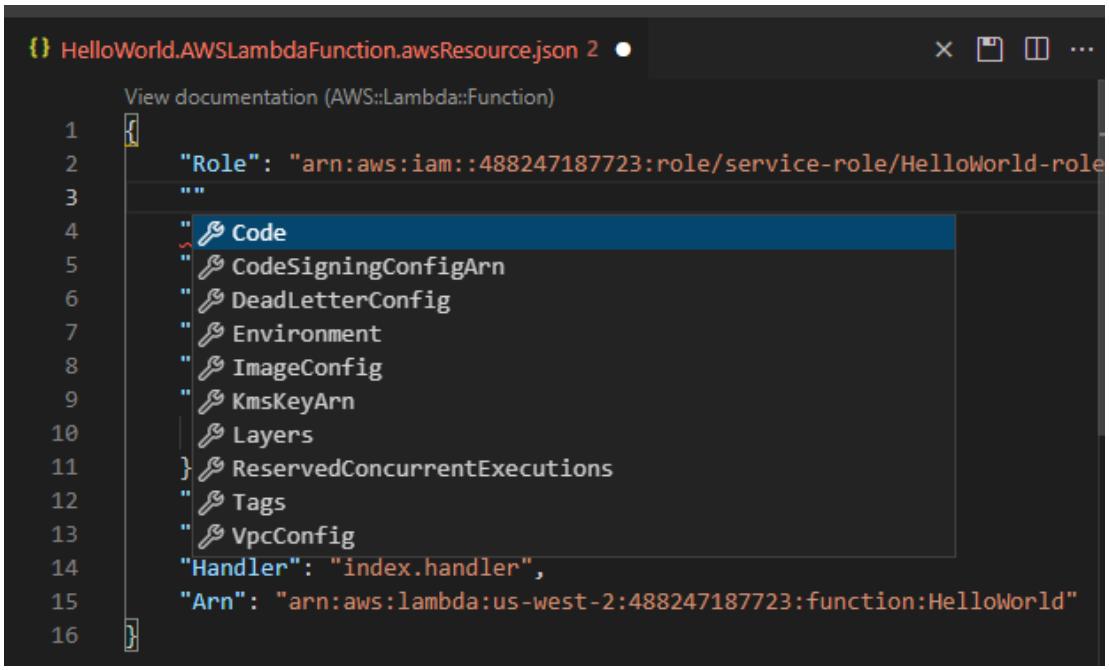
1. Ajoutez le type de ressource à la ressource que vous souhaitez créer en cliquant avec le bouton droit sur Ressources et en choisissant Afficher les ressources.
2. Une fois le type de ressource ajouté sous Ressources, cliquez sur l'icône plus (« + ») pour ouvrir le fichier modèle dans un nouvel éditeur.

   Vous pouvez également cliquer avec le bouton droit sur le nom du type de ressource et sélectionner Créer. Vous pouvez également accéder à des informations sur la façon de modéliser la ressource en choisissant Afficher la documentation.



3. Dans l'éditeur, commencez à définir les propriétés qui composent le modèle de ressource. La fonction de saisie semi-automatique suggère des noms de propriétés conformes au schéma de votre modèle. Lorsque vous passez la souris sur un type de propriété, un volet affiche une description de son utilisation. Pour obtenir des informations détaillées sur le schéma, choisissez Afficher la documentation.

Tout texte qui n'est pas conforme au schéma des ressources est indiqué par un soulignement rouge ondulé.



4. Une fois que vous avez fini de déclarer votre ressource, cliquez sur l'icône Enregistrer pour valider votre modèle et enregistrer la ressource dans le AWS Cloud distant.

   Si votre modèle définit la ressource conformément à son schéma, un message s'affiche pour confirmer la création de la ressource. (Si la ressource existe déjà, le message confirme que la ressource a été mise à jour.)

   Une fois la ressource créée, elle est ajoutée à la liste sous l'en-tête du type de ressource.

5. Si votre fichier contient des erreurs, un message s'affiche pour expliquer que la ressource n'a pas pu être créée ou mise à jour. Choisissez Afficher les journaux pour identifier les éléments du modèle que vous devez corriger.

Résolution des problèmes AWS Toolkit for Visual Studio Code

Les sections suivantes contiennent des informations générales de résolution des problèmes concernant AWS Toolkit for Visual Studio Code les AWS services du kit d'outils et son utilisation. Pour les problèmes spécifiquement liés à la résolution des problèmes SAM dans le AWS kit d'outils, consultez la rubrique Résolution des problèmes liés aux applications sans serveur dans ce guide de l'utilisateur.

Rubriques

- Bonnes pratiques de résolution des problèmes
- Le profil... est introuvable dans le fichier de configuration
- Schéma SAM json : impossible de modifier le schéma dans le fichier template.yaml

Bonnes pratiques de résolution des problèmes

Les meilleures pratiques recommandées pour résoudre les AWS Toolkit for Visual Studio Code problèmes sont les suivantes. Pour des informations détaillées sur la manière dont vous pouvez contribuer au AWS Toolkit for Visual Studio Code, consultez la AWS Toolkit for Visual Studio Code rubrique Contribuer au AWS Toolkit for Visual Studio Code GitHub référentiel.

- Essayez de recréer votre problème ou votre erreur avant d'envoyer un rapport.
- Prenez des notes détaillées sur chaque étape, chaque réglage et chaque message d'erreur pendant le processus de recréation.
- Collectez les journaux de débogage de votre AWS boîte à outils. Pour une description détaillée de la localisation de vos journaux de débogage du AWS Toolkit, consultez la procédure Comment localiser vos AWS journaux, qui se trouve dans cette rubrique du guide de l'utilisateur.
- Vérifiez les demandes ouvertes, les solutions connues ou signalez votre problème non résolu dans la section AWS Toolkit for Visual Studio Code Problèmes du AWS Toolkit for Visual Studio Code GitHub référentiel.

> **Note**
>
> La procédure suivante décrit comment afficher les journaux de débogage de votre AWS Toolkit. Le processus d'affichage de vos journaux de débogage Amazon Q est identique, sauf que vous choisissez Amazon Q : View Logs dans la palette de commandes VS Code.

Comment localiser votre AWS Toolkit for Visual Studio Code Journaux de débogage

1. À partir du VS Code, ouvrez la palette de commandes en appuyant sur **`Cmd + Shift + P`** ou **`Ctrl + Shift + P`** (Windows) et entrez **`AWS View Logs`** dans le champ de recherche.
2. Choisissez AWS Afficher les journaux pour ouvrir les journaux de votre AWS boîte à outils dans la fenêtre de sortie du terminal VS Code.
3. Dans la fenêtre de sortie du terminal VS Code, élargissez le menu de l'icône Gear et choisissez Debug.
4. Développez à nouveau le menu de l'icône en forme d'engrenage et choisissez Définir par défaut.
5. Re-open dans la palette de commandes en appuyant sur **`Cmd + Shift + P`** ou **`Ctrl + Shift + P`** (Windows) et en recherchant**`Reload Window`**, puis en choisissant Développeur : Recharger la fenêtre.
6. VS Code se recharge et la fenêtre de sortie du terminal VS Code affiche vos journaux de débogage du AWS Toolkit mis à jour.

Le profil... est introuvable dans le fichier de configuration

Problème

> **Note**
>
> Ce problème ne concerne que le `~/.aws/config` fichier et non le `~/.aws/credentials` fichier. Pour obtenir des informations détaillées sur les fichiers de AWS configuration et AWS d'informations d'identification, consultez la rubrique Fichiers de configuration et d'informations d'identification partagés dans le guide de référence du AWS SDK et des outils.

Lorsque vous choisissez les informations d'identification, les journaux du AWS Toolkit affichent un message avec cette structure :`Profile name could not be found in shared credentials file`.

Voici un exemple de ce à quoi ressemble cette erreur dans les journaux de votre AWS boîte à outils :

```
        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.
```

Solution

Si votre profil existe déjà dans`~/.aws/config`, vérifiez qu'il commence par`[profile` . Voici un exemple de profil utilisateur correctement structuré :

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

Voici un exemple de profil utilisateur mal structuré :

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

Schéma SAM json : impossible de modifier le schéma dans le fichier template.yaml

Problème

Vous ne pouvez pas sélectionner manuellement un autre schéma JSON dans SAM template.yaml

Solution

Après la mise à jour vers la version 1.11+ de vscode-yaml, vous pouvez ajouter une **`yaml-language-server`** ligne de modèle en haut d'un fichier YAML pour forcer l'utilisation d'un schéma par URI. Pour plus d'informations sur l'utilisation du schéma intégré, consultez la section relative au serveur de langage yaml du référentiel de développement Redhat. GitHub Voici un exemple de **`yaml-language-server`** modeline.

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

Sécurité dans AWS Toolkit for VS Code

La sécurité du cloud AWS est la priorité absolue. En tant que AWS client, vous bénéficiez de centres de données et d'architectures réseau conçus pour répondre aux exigences des entreprises les plus sensibles en matière de sécurité.

La sécurité est une responsabilité partagée entre vous AWS et vous. Le modèle de responsabilité partagée décrit ceci comme la sécurité du cloud et la sécurité dans le cloud :

- Sécurité du cloud : AWS est chargée de protéger l'infrastructure qui exécute les AWS services dans le AWS Cloud. AWS vous fournit également des services que vous pouvez utiliser en toute sécurité. Third-party les auditeurs testent et vérifient régulièrement l'efficacité de notre sécurité dans le cadre des programmes de AWS conformité Programmes de de conformité. Pour en savoir plus sur les programmes de conformité qui s'appliquent à AWS Toolkit for VS Code, voir AWS Services concernés par programme de conformitéAWS .
- Sécurité dans le cloud — Votre responsabilité est déterminée par le AWS service que vous utilisez. Vous êtes également responsable d'autres facteurs, y compris la sensibilité de vos données, les exigences de votre entreprise, ainsi que la législation et la réglementation applicables.

Cette documentation vous aide à comprendre comment appliquer le modèle de responsabilité partagée lors de l'utilisation de Toolkit for VS Code. Les rubriques suivantes expliquent comment configurer Toolkit for VS Code afin d'atteindre vos objectifs de sécurité et de conformité. Vous apprendrez également à utiliser d'autres AWS services qui vous aident à surveiller et à sécuriser les ressources de votre Toolkit for VS Code.

Rubriques

- Protection des données dans AWS Toolkit for VS Code

Protection des données dans AWS Toolkit for VS Code

Le modèle de responsabilité AWS partagée (modèle de) s'applique à la protection des données dans AWS Toolkit for VS Code. Comme décrit dans ce modèle, AWS est chargé de protéger l'infrastructure mondiale qui gère tous les AWS Cloud. La gestion du contrôle de votre contenu hébergé sur cette infrastructure relève de votre responsabilité. Vous êtes également responsable des tâches de configuration et de gestion de la sécurité des Services AWS que vous utilisez. Pour plus d'informations sur la confidentialité des données, consultez la FAQ sur la confidentialité des données

et les . Pour plus d'informations sur la protection des données en Europe, consultez le Centre du règlement général sur la protection des données (RGPD).

À des fins de protection des données, nous vous recommandons de protéger les Compte AWS informations d'identification et de configurer les utilisateurs individuels avec AWS IAM Identity Center ou Gestion des identités et des accès AWS (IAM). Ainsi, chaque utilisateur se voit attribuer uniquement les autorisations nécessaires pour exécuter ses tâches. Nous vous recommandons également de sécuriser vos données comme indiqué ci-dessous :

- Utilisez l'authentification multifactorielle (MFA) avec chaque compte.
- SSL/TLS À utiliser pour communiquer avec AWS les ressources. Nous exigeons TLS 1.2 et recommandons TLS 1.3.
- Configurez l'API et la journalisation de l'activité des utilisateurs avec AWS CloudTrail. Pour plus d'informations sur l'utilisation des CloudTrail sentiers pour capturer AWS des activités, consultez la section Utilisation des CloudTrail sentiers dans le guide de AWS CloudTrail l'utilisateur.
- Utilisez des solutions de AWS chiffrement, ainsi que tous les contrôles de sécurité par défaut qu'ils contiennent Services AWS.
- Utilisez des services de sécurité gérés avancés tels qu'Amazon Macie, qui contribuent à la découverte et à la sécurisation des données sensibles stockées dans Amazon S3.
- Si vous avez besoin de modules cryptographiques validés par la norme FIPS 140-3 pour accéder AWS via une interface de ligne de commande ou une API, utilisez un point de terminaison FIPS. Pour plus d'informations sur les points de terminaison FIPS disponibles, consultez Norme FIPS (Federal Information Processing Standard) 140-3.

Nous vous recommandons fortement de ne jamais placer d'informations confidentielles ou sensibles, telles que les adresses e-mail de vos clients, dans des balises ou des champs de texte libre tels que le champ Nom. Cela inclut lorsque vous travaillez avec Toolkit for VS Code ou un autre outil Services AWS à l'aide de la console, de l'API ou AWS des SDK. AWS CLI Toutes les données que vous entrez dans des balises ou des champs de texte de forme libre utilisés pour les noms peuvent être utilisées à des fins de facturation ou dans les journaux de diagnostic. Si vous fournissez une adresse URL à un serveur externe, nous vous recommandons fortement de ne pas inclure d'informations d'identification dans l'adresse URL permettant de valider votre demande adressée à ce serveur.

Historique du document pour AWS Toolkit for Visual Studio Code Guide de l'utilisateur

Le tableau ci-après décrit les modifications importantes dans chaque édition du AWS Toolkit for Visual Studio Code. Pour recevoir les notifications concernant les mises à jour de cette documentation, abonnez-vous à un flux RSS.

Modification	Description	Date
Amazon SageMaker Unified Studio	Intégration avec le service Amazon SageMaker Unified Studio.	18 septembre 2025
LocalStack	Nouvelle rubrique du guide de l'utilisateur ajoutée pour soutenir le lancement de LocalStack.	11 septembre 2025
Utilisation des AWS Lambda fonctions	Mise à jour de la rubrique du guide de l'utilisateur pour inclure les fonctionnalités Lambda mises à jour dans le kit d'outils.	17 juillet 2025
AWS Lambda débogage à distance	Ajout d'une nouvelle rubrique sur le débogage à AWS Lambda distance dans le guide de AWS Toolkit for Visual Studio Code l'utilisateur.	17 juillet 2025
AWS Lambda De la console à l'IDE	Ajout AWS Lambda console d'une nouvelle rubrique relative à l'IDE dans le guide de AWS Toolkit for Visual Studio Code l'utilisateur.	17 juillet 2025

Mises à jour AWS Step Functions du contenu et prise en charge accrue de Workflow Studio	Ajout de mises à jour au contenu existant AWS Step Functions et à la rubrique du guide de l'utilisateur AWS Step Functions de Workflow Studio, à l'appui du lancement des fonctionnalités.	6 mars 2025
AWS Terre sans serveur	Ajout d'une nouvelle rubrique AWS Serverless Land à la table des matières AWS d'Application Builder.	6 mars 2025
Mettre à jour les pare-feux et les passerelles pour autoriser l'accès	Listes des points de terminaison et des ressources qui doivent être autorisés pour accéder à tous les services et fonctionnalités des extensions AWS Toolkit for Visual Studio Code et Amazon Q for VS Code.	28 février 2025
Support pour Amazon ECR App Runner	Ajout d'un support de documentation pour le lancement d'un AWS App Runner service depuis le nœud Amazon Elastic Container Registry, dans le AWS Toolkit.	6 février 2025
Amazon DocumentDB	Ajout d'une nouvelle rubrique Amazon DocumentDB au guide de l' AWS Toolkit for Visual Studio Code utilisateur.	6 février 2025

Assistance EC2	Support pour le service Amazon Elastic Compute Cloud a été ajouté à la boîte à outils.	31 janvier 2025
AWS Documents	Ajout d'une nouvelle rubrique dans le guide de l'utilisateur pour AWS Documents.	20 janvier 2025
Amazon CloudWatch Logs Live Tail	Ajout d'un nouveau sous-thème pour prendre en charge la fonctionnalité Amazon CloudWatch Logs Live Tail dans le AWS Toolkit for Visual Studio Code.	15 décembre 2024
AWS Générateur d'applications	Ajout AWS d'une nouvelle rubrique sur le générateur d'applications au guide de AWS Toolkit for Visual Studio Code l'utilisateur.	30 octobre 2024
Compositeur d'infrastructure	AWS Application Composer est désormais AWS Infrastructure Composer.	3 octobre 2024
AWS Mises à jour de Identity and Access Management (IAM) Access Analyzer	Le contenu d'IAM Access Analyzer a été mis à jour pour inclure de nouvelles références d'API.	10 juillet 2024
AWS Analyseur d'accès Identity and Access Management (IAM)	Ajout d'une nouvelle rubrique dans le guide de l'utilisateur pour IAM Access Analyzer.	23 mai 2024

Connect to AWS Authorization flow a été mis à jour	Le flux d'autorisation a été mis à jour pour refléter les modifications apportées au processus d'authentification et la séparation d'Amazon Q du AWS Toolkit for Visual Studio Code.	30 avril 2024
Extension Amazon Q pour VS Code	Depuis le 30 avril 2024, CodeWhisperer il fait désormais partie d'Amazon Q et Amazon Q est disponible en tant qu'extension pour VS Code.	30 avril 2024
Support du cloud privé virtuel dans les environnements de développement	Contenu mis à jour concernant les modifications apportées à l'interface utilisateur pour prendre en charge le VPC dans les environnements de développement.	21 janvier 2024
Compositeur d'infrastructure	Ajout d'une nouvelle rubrique Infrastructure Composer au guide de AWS Toolkit for Visual Studio Code l'utilisateur.	28 novembre 2023
Support SSO pour CodeCatalyst	Contenu mis à jour pour couvrir le support d'IAM Identity Center pour CodeCatalyst les environnements de développement.	17 novembre 2023
Liens de téléchargement du code VS et du kit d'outils ajoutés	Contenu mis à jour avec des liens de téléchargement pour VS Code et le AWS Toolkit for Visual Studio Code.	1er novembre 2023

Rubrique Amazon Redshift	Ajout d'une nouvelle rubrique Amazon Redshift au guide de l' AWS Toolkit for Visual Studio Code utilisateur.	17 octobre 2023
Connect to AWS Authorization flow a été mis à jour	Le flux d'autorisation a été mis à jour pour se concentrer sur les méthodes d'authentification spécifiques au service.	29 septembre 2023
Guide de l'utilisateur créé : Création d'un modèle CloudFormation	Création d'un nouveau guide d'utilisation décrivant comment créer un CloudFormation modèle à l'aide du Toolkit for VS Code	17 décembre 2021
Mise à jour mineure de l'interface	Le texte existant a été mis à jour pour « Preview Machine State » en « Render graph » afin de mieux correspondre à l'interface utilisateur.	14 décembre 2021
Guide utilisateur créé pour Amazon Elastic Container Service Exec	Ceci est une présentation d'Amazon ECS Exec.	13 décembre 2021
Guide de l'utilisateur créé pour le service AWS IoT Toolkit for VS Code	Ce guide de l'utilisateur a pour but de vous aider à commencer à utiliser le AWS IoT service Toolkit for VS Code.	22 novembre 2021
Prise en charge des fonctionnalités expérimentales	Ajout de la prise en charge de l'activation des fonctionnalités expérimentales pour les AWS services.	14 octobre 2021

Support pour les AWS ressources	Ajout de la prise en charge de l'accès aux types de ressources ainsi que des options d'interface pour créer, modifier et supprimer des ressources.	14 octobre 2021
Présentation du service Amazon ECR pour AWS Toolkit for Visual Studio Code	Ajout d'une présentation et d'une présentation des fonctionnalités du service Amazon ECR accessibles dans VS Code	14 octobre 2021
Support pour les environnements ARM64	Vous pouvez désormais exécuter des applications sans serveur dans des environnements ARM64-based émulés ainsi que dans des environnements basés sur x86_64.	1er octobre 2021
Application sans serveur AWS	Ajout du support pour l'exécution d' AWS SAM applications sur la plate-forme ARM64	30 septembre 2021
Node.js Section de mise à jour du format	Selon les commentaires des clients, mise à jour du formatage pour Node.js/ TypeScript.	12 août 2021
Prise en charge d'App Runner	Ajout du support AWS App Runner pour AWS Toolkit for Visual Studio Code.	11 août 2021
Fonctions de Debugging Go	Ajout du support pour le débogage des fonctions Go locales.	10 mai 2021

Débogage des fonctions Java	Ajout du support pour le débogage des fonctions Java locales.	22 avril 2021
Support YAML pour AWS Step Functions	Ajout du support YAML pour. AWS Step Functions	4 mars 2021
Ressources de débogage d'Amazon API Gateway	Ajout de la prise en charge du débogage des ressources Amazon API Gateway locales.	1er décembre 2020
Amazon API Gateway	Ajout de la prise en charge d'Amazon API Gateway.	1er décembre 2020
Application sans serveur AWS	Ajout de la prise en charge des images de conteneur Lambda avec des applications sans serveur.	1er décembre 2020
AWS Systems Manager soutien	Ajout du support pour les documents d'automatisation de Systems Manager.	30 septembre 2020
CloudWatch Journaux	Ajout du support pour CloudWatch les journaux.	24 août 2020
Amazon S3	Ajout de la prise en charge d'Amazon S3.	30 juillet 2020
AWS Step Functions soutien	Ajout du support pour AWS Step Functions.	31 mars 2020
Utilisation d'Amazon EventBridge Schemas	Ajout de la prise en charge d'Amazon EventBridge Schemas	1 décembre 2019
AWS CDK	Version préliminaire du AWS CDK service.	25 novembre 2019

Utilisation d'un processus d'accréditation externe	Ajout d'informations sur l'utilisation d'un processus d'identification externe pour obtenir des AWS informations d'identification.	25 septembre 2019
Utilisation IntelliSense pour les fichiers de définition de tâches	IntelliSense un support a été ajouté pour travailler avec les fichiers de définition de tâches Amazon ECS.	24 septembre 2019
Guide de l'utilisateur pour AWS Toolkit for Visual Studio Code	Version pour la disponibilité générale	11 juillet 2019
Guide de l'utilisateur pour AWS Toolkit for Visual Studio Code	Mise à jour de la structure du document pour une clarté et une facilité d'utilisation accrues.	3 juillet 2019
Installation du AWS Toolkit for Visual Studio Code	Ajout d'informations sur l'installation de kits SDK de langue pour prendre en charge diverses chaînes d'outils.	12 juin 2019
Configurez votre chaîne d'outils	Ajout d'informations sur la configuration de diverses chaînes d'outils.	12 juin 2019
Publication initiale	Version initiale du Guide de l'utilisateur AWS Toolkit for Visual Studio Code.	28 mars 2019

Les traductions sont fournies par des outils de traduction automatique. En cas de conflit entre le contenu d'une traduction et celui de la version originale en anglais, la version anglaise prévaudra.