



Guide de l'utilisateur

AWS Boîte à outils pour VS Code



AWS Boîte à outils pour VS Code: Guide de l'utilisateur

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

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

Table of Contents

AWS Toolkit for Visual Studio Code	1
Qu'est-ce que AWS Toolkit for Visual Studio Code	1
Informations connexes	1
Amazon Q Developer 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
Démarrer	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
Prérequis	6
Ouverture du panneau de connexion	7
Connexion AWS depuis le kit d'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
Configuration d'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
Utiliser votre chaîne d'outils	13
Authentification et accès	14
IAM Identity Center	14
Informations d'identification IAM	14
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'informations d'identification externe	18
Travailler avec AWS	19
Fonctions expérimentales	20
AWS Explorateur	20
Amazon CodeCatalyst	21
Qu'est-ce qu'Amazon CodeCatalyst ?	21
Démarrer avec CodeCatalyst	22
Utilisation des ressources CodeCatalyst	22
Travailler avec des environnements de développement	27
Résolution des problèmes	30
Amazon API Gateway	31
AWS App Runner	31
Prérequis	32
Tarification	35
Création de services App Runner	36
Gestion des services App Runner	39
AWSCompositeur d'applications	42
Utilisation d'AWSApplication Composer	42
AWS CDK	43
AWS CDKApplications	44
Piles AWS CloudFormation	46
Suppression d'unAWS CloudFormationempiler	46
Création d'un CloudFormation modèle	47
AmazonCloudWatchJournaux	49
Affichage des groupes de journaux et des flux de journaux CloudWatch Logs	50
Utilisation d' CloudWatch événements de journaux	51
Recherche de groupes de journaux	53
Amazon ECR	56
Prérequis	56
Utilisation d'images Amazon ECR avec Toolkit for VS Code	58
Amazon ECS	67
Utilisation d'IntelliSense pour les fichiers de définition de tâches	67
Amazon ECS Exec	69
Amazon EventBridge	71
Utilisation d'Amazon EventBridge Schemas	71

AWS Analyseur d'accès IAM	74
Utilisation d' AWS IAM Access Analyzer	74
AWS IoT	79
Conditions préalables requises AWS IoT	79
AWS IoTObjets	79
AWS IoTCertificats	81
Stratégies AWS IoT	84
Fonctions AWS Lambda	87
Interaction avec des fonctions Lambda distantes	88
Amazon Redshift	94
Travailler avec Amazon Redshift	94
Amazon S3	99
Utilisation des ressources S3	100
Utilisation d'objets S3	102
Application sans serveur AWS	106
Démarrage	106
Exécution et débogage de fonctions Lambda directement à partir du code	115
Exécution et débogage des ressources locales Amazon API Gateway	119
Options de configuration pour le débogage d'applications sans serveur	123
Dépannage	131
AWS Systems Manager	133
Hypothèses et prérequis	134
Autorisations IAM pour les documents d'automatisation de Systems Manager	135
Création d'un nouveau document de Systems Manager Automation de	135
Ouverture d'un document d'automatisation de Systems Manager Automation	136
Modification d'un document d'automatisation Systems Manager Automation	137
Publication d'un document d'automatisation Systems Manager Automation	138
Suppression d'un document Systems Manager Automation de	138
Exécution d'un document Systems Manager Automation de	139
Résolution des problèmes	139
AWS Step Functions	140
AWS Step Functions et VS Code	140
Compositeur de menaces	151
Utilisation de Threat Composer	151
Ressources	152
Autorisations IAM pour l'accès aux ressources	153

Ajouter des ressources existantes et interagir avec celles-ci	154
Créer et modifier des ressources	155
Sécurité	158
Protection des données	158
Historique de document	160
.....	clxiv

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

- [Premiers pas avec AWS Toolkit for Visual Studio Code](#)
- [Utilisation de AWS services et d'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/](https://code.visualstudio.com/).

Amazon Q Developer et Amazon CodeWhisperer

Depuis le 30 avril 2024, Amazon CodeWhisperer fait désormais partie d'Amazon Q Developer, ce qui inclut les suggestions de code intégrées et les scans de sécurité d'Amazon Q Developer. Téléchargez l'[extension Amazon Q Developer IDE sur VS Code Marketplace](#) pour commencer.

Pour en savoir plus 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 JetBrains](#) utilisateur.

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](#).

Premiers pas avec AWS Toolkit for Visual Studio Code

Vos AWS Toolkit for Visual Studio Code AWS services et ressources sont alors disponibles, directement à partir de 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

Pour 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 elle est nécessaire pour toutes les fonctionnalités AWS Serverless Application Model (AWS SAM), notamment [Création d'une nouvelle application sans serveur \(locale\)](#).

Pour plus d'informations, consultez [Installation de l'interface de ligne de commande AWS SAM](#) dans le [Manuel du développeur AWS Serverless Application Model](#).

- 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.

Prérequis

Si vous êtes nouveau AWS ou si vous n'avez pas créé de compte, vous devez suivre 3 étapes principales pour le connecter AWS Toolkit for Visual Studio Code à votre AWS compte :

1. Création d'un AWS compte : vous pouvez créer un AWS compte depuis le [portail d'AWS inscription](#). Pour obtenir des informations détaillées sur la configuration d'un nouveau AWS compte, consultez la rubrique [Présentation](#) du Guide de l'utilisateur de AWS configuration.
2. 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.

3. 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 panneau de connexion à la AWS boîte à outils depuis l' 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 panneau de connexion au AWS Toolkit à l'aide de 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 AWS depuis le kit d'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 à 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 Name**, **Access 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 Builder ID

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

L'AWS Toolkit for Visual Studio Code prend en charge plusieurs langues dans tous les services AWS. 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 l'AWS Toolkit for VS Code [est installé](#).
2. Installez l'[extension C#](#). Cette extension permet à VS Code de Microsoft Core.
3. Ouvrez une application AWS Serverless Application Model (AWS SAM) ou [créez-en une](#).
4. Ouvrez le dossier qui contient `template.yaml`.

Configuration d'une chaîne d'outils pour Node.js

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

Note

Lorsque vous déboguez une fonction TypeScript Lambda directement à partir du code source (c'est le cas de la configuration de lancement `"target": "code"`), le TypeScript compilateur doit être installé globalement ou dans celui de votre projet `package.json`.

Configuration d'une chaîne d'outils pour Python

1. Assurez-vous que l'AWS Toolkit for VS Code [est installé](#).
2. Installez l'[extension Python pour Visual Studio Code](#). Cette extension permet à VS Code de déboguer les applications Python.
3. Ouvrez une application AWS SAM 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 :
 - Script shell Bash : `./venv/Scripts/activate`
 - PowerShell: `./venv/Scripts/Activate.ps1`

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 application AWS SAM 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 des environnements d'exécution Go1.15+.

4. Installez les outils Go à l'aide de la [palette de commandes](#) :
 - a. Dans la palette de commandes, sélectionnez `Go: Install/Update Tools`.
 - b. Dans l'ensemble des cases à cocher, sélectionnez `dlv` et `gopls`.
5. Ouvrez une application AWS SAM ou [créez-en une](#).

6. Ouvrez le dossier qui contient `template.yaml`.

Utiliser votre chaîne d'outils

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

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 Centre d'identité IAM](#)
- [AWS Informations d'identification IAM](#)
- [AWS Builder ID pour les développeurs](#)
- [Utilisation d'un processus d'informations d'identification externe](#)

AWS Centre d'identité IAM

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 sur l'[authentification IAM Identity Center du Guide de référence des AWS SDK et des outils](#).

Pour plus de détails sur la façon d'authentifier et de connecter le AWS kit à outils avec vos informations d'identification IAM Identity Center existantes, consultez la AWS rubrique [Connect to de](#) 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 de détails sur la façon d'authentifier et de connecter le AWS kit à outils avec vos informations d'identification AWS IAM existantes, consultez la AWS rubrique [Connect to](#) du présent 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 les informations d'identification IAM via un autre AWS service (tel que le AWS CLI), ces informations d'identification sont AWS Toolkit for Visual Studio Code automatiquement détectées et mises à disposition dans VS Code.
- AWS recommande d'utiliser l'authentification IAM Identity Center. Pour plus d'informations sur les meilleures pratiques en matière d' AWS IAM, consultez la section [Bonnes pratiques de sécurité en matière d'IAM](#) du guide de l'utilisateur AWS d'Identity and Access Management.
- 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 configurer l' AWS Toolkit for Visual Studio Code authentification avec votre AWS compte, vous devez suivre 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](#) sur les AWS SDK et les outils.

Note

Étape 3 : La mise à jour du fichier d'informations d'identification partagé dans le guide de référence AWS des SDK et des outils est facultative.

Si vous terminez l'étape 3, vos informations d'identification seront AWS Toolkit for Visual Studio Code automatiquement détectées lors 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 AWS Toolkit for Visual Studio Code , vous pouvez suivre le 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 partagées 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 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 au AWS kit d'outils, choisissez IAM Credential, puis cliquez sur le bouton Continuer pour continuer.
5. Entrez le **Profile Name****Access 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.
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 que `[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 infructueuse.

```
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 indiqué 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'informations 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'accréditation 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 [Approvisionnement des informations d'identification avec un processus externe](#) dans le Guide de l'utilisateur de AWS Command Line Interface.

Utilisation de AWS services et d'outils

Elle 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 des rubriques du guide traitant de chaque service Toolkit for VS Code et de ses fonctionnalités. Choisissez un service ou un outil pour plus d'informations sur son fonctionnement, son paramétrage et son utilisation des fonctionnalités de base.

Rubriques

- [Utilisation des fonctions expérimentales](#)
- [Utilisation des AWS services dans l' AWS explorateur](#)
- [Amazon CodeCatalyst pour VS Code](#)
- [Utilisation avec Amazon API Gateway](#)
- [Utilisation AWS App Runner avec AWS Toolkit for Visual Studio Code](#)
- [AWSCompositeur d'applications](#)
- [AWS CDKpour VS Code](#)
- [Utilisation d'AWS CloudFormationpiles](#)
- [Travailler avecCloudWatchSe connecte à l'aide duAWS Toolkit for Visual Studio Code](#)
- [Fonctionnement d'Amazon Elastic Container Registry](#)
- [Utilisation d'Amazon Elastic Container Service](#)
- [Utilisation des outils Amazon EventBridge](#)
- [AWS Analyseur d'accès IAM](#)
- [Utilisation d'AWS IoT dans AWS Toolkit for Visual Studio Code](#)
- [Utilisation des fonctions AWS Lambda](#)
- [Amazon Redshift dans le kit d'outils pour VS Code](#)
- [Travailler avec Amazon S3](#)
- [Utilisation avec applications sans serveur](#)
- [Utilisation des documents d'automatisation de Systems Manager Automation](#)
- [Travailler avec AWS Step Functions](#)
- [Utilisation de Threat Composer](#)

- [Utilisation des ressources](#)

Utilisation des fonctions expérimentales

Les fonctionnalités expérimentales offrent un accès anticipé aux fonctionnalités de AWS Toolkit for Visual Studio Code avant qu'ils soient officiellement disponibles.

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 de l'AWS Toolkit for Visual Studio Code sans préavis.

Vous pouvez activer des fonctionnalités expérimentales pour des AWS services dans le `AWSToolkitSection` du `Paramètres` dans votre IDE VS Code.

1. Pour modifier des AWS paramètres dans VS Code, choisissez `Fichier, Préférences, Paramètres`.
2. Dans `Paramètres` volet, développez `Extension` et choisissez `AWSToolkit`.
3. Under `AWS : Experiments`, cochez les cases correspondant aux fonctionnalités expérimentales auxquelles vous souhaitez accéder avant la publication. Si vous souhaitez désactiver une fonctionnalité expérimentale, décochez la case correspondante.

Utilisation des AWS services dans l' AWS explorateur

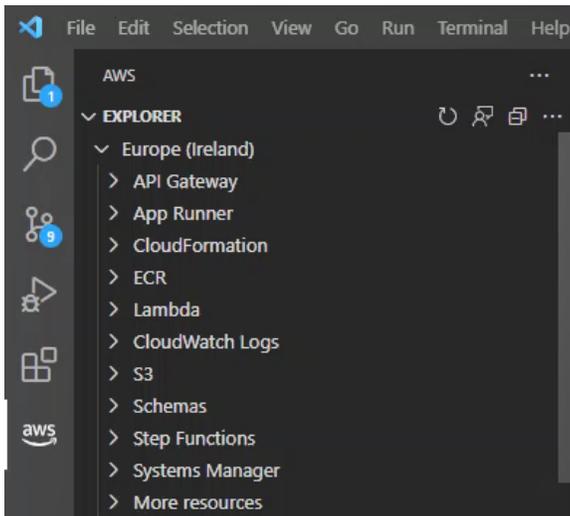
L'AWS explorateur vous donne un aperçu de certains des AWS services avec lesquels vous pouvez travailler lorsque vous utilisez le AWS Toolkit for Visual Studio Code.

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 AWS sur 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

Amazon CodeCatalyst pour VS Code

Qu'est-ce qu'Amazon CodeCatalyst ?

Amazon CodeCatalyst est un espace de collaboration basé sur le cloud destiné aux 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 le AWS kit d'outils pour lancer les 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

- [Débuter avec CodeCatalyst le Toolkit for 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](#)

Débuter avec CodeCatalyst le Toolkit for VS Code

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

Rubriques

- [Création d'un CodeCatalyst compte](#)
- [Connecter le AWS kit d'outils à CodeCatalyst](#)

Création d'un CodeCatalyst compte

Vous devez disposer d'un identifiant AWS Builder actif ou AWS IAM Identity Center d'informations d'identification pour vous connecter CodeCatalyst depuis le 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 kit d'outils à CodeCatalyst

Pour connecter le AWS kit d'outils à votre CodeCatalyst compte, consultez la CodeCatalyst section [Authentification pour Amazon](#) dans la AWS rubrique Connexion à ce guide de l'utilisateur.

Utilisation des CodeCatalyst ressources Amazon dans VS Code

Les sections suivantes fournissent un aperçu 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 dont vous pouvez 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

- [Cloner un dépôt](#)
- [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](#)

Cloner un dépôt

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 en local, 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 avec le AWS Toolkit, procédez comme suit :

Note

Si vous clonez un référentiel à partir d'un service tiers, vous serez peut-être invité à vous authentifier avec les 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 dépôt cloné, vous souhaitez ouvrir le référentiel cloné ? un message apparaît.

1. Dans le Toolkit for 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 dépôt dans votre fenêtre VS Code actuelle, choisissez Ouvrir.
 - Pour ouvrir votre dépôt dans une nouvelle fenêtre VS Code, choisissez Ouvrir dans une nouvelle fenêtre.
 - Pour terminer le processus de clonage sans ouvrir votre dépôt, 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'état CodeCatalyst. La fenêtre d'état est mise à jour lorsque le processus est terminé.

- Si l'environnement de développement ne s'ouvre pas, le statut est mis à jour avec des informations expliquant pourquoi le processus a échoué et un lien permettant d'ouvrir les journaux du processus.
- Si le processus aboutit, votre environnement de développement s'ouvre dans une nouvelle fenêtre, depuis VS Code.

1. Dans le Toolkit for 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 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 de manière permanente. 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 au bon fonctionnement de votre environnement de développement.
- 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 dossier Faites-vous confiance aux auteurs des fichiers de ce dossier ? une invite apparaît également. Acceptez les conditions générales pour continuer à travailler dans votre environnement de développement.

1. Dans le Toolkit for 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 puissance de traitement et la quantité de RAM attribuées à votre système.
 - Délai d'attente : choisissez Modifier le délai d'expiration pour modifier le temps d'inactivité du système autorisé 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 Create Dev Environment 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.

La création de liens vers un référentiel tiers en tant que source est gérée au niveau du projet dans CodeCatalyst. Pour obtenir des instructions et des informations supplémentaires sur la façon de connecter un référentiel tiers à votre environnement de développement, consultez la rubrique [Lier un référentiel source](#) dans le guide de CodeCatalyst l'utilisateur Amazon.

CodeCatalyst commandes dans VS Code

Des commandes VS Code supplémentaires sont attribuées à des fonctionnalités CodeCatalyst connexes qui ne sont pas affichées directement dans le AWS Toolkit.

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

1. Dans le Toolkit for VS Code, développez l'explorateur DEVELOPER TOOLS.
2. Choisissez Afficher CodeCatalyst les commandes pour ouvrir la palette de commandes avec une recherche préremplie pourCodeCatalyst.
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 travailler à partir d'environnements de développement à l'aide du AWS Toolkit for Visual Studio Code.

Pour des informations détaillées sur les environnements de développement, consultez la rubrique [Environnements](#) de développement dans le guide de CodeCatalyst l'utilisateur Amazon.

Configuration de votre environnement de développement avec des fichiers de développement

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 fichier de développement. 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 transférer les mises à jour vers votre référentiel source avant que les mises à jour ne prennent effet.

Pour des informations détaillées sur la configuration des environnements de développement avec des fichiers de développement, consultez la rubrique [Configuration de votre environnement](#) de développement dans le guide de l' CodeCatalystutilisateur Amazon.

La procédure suivante décrit comment modifier votre fichier de développement à partir d'une instance distante du Toolkit pendant son exécution dans un environnement de développement.

Important

Si vous modifiez le code `Devfile` depuis VS, tenez compte de ce qui suit :

- La modification du nom du fichier de développement ou du nom du composant de fichier de développement 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 pas associé à un

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 savoir comment définir et configurer votre `Devfile`, consultez la documentation relative à l'[ajout de commandes](#) sur le site Web devfile.io.

1. Dans le Toolkit for 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 fichier de développement, 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 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 avec les 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 mise à jour des informations d'identification, l'authentification et la connexion à AWS partir du kit d'outils, consultez la AWS section [Connexion à](#) de la rubrique Mise en route 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](#) du présent guide de l'utilisateur.

Utilisation du Toolkit for VS Code dans les environnements de développement

Après avoir ouvert ou créé un environnement de développement dans VS Code, vous pouvez travailler à partir du Toolkit for 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. Dans le Toolkit for VS Code, développez l'explorateur DEVELOPER TOOLS.
2. Développez CodeCatalyst et choisissez Stop Dev Environment.
3. Lorsque VS Code vous le demande, 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. Dans le Toolkit for 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 correspondant à 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 : affichez le nom et l'emplacement de votre environnement Devfile de développement. Ouvrez le votre 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 traitent des problèmes techniques potentiels liés à l'utilisation d'Amazon CodeCatalyst et de VS Code.

Rubriques

- [Version du code VS](#)
- [Autorisations pour Amazon CodeCatalyst](#)
- [Connexion à un environnement de développement depuis le Toolkit for VS Code](#)

Version du code VS

Votre version de VS Code devrait 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 à partir de VS Code Insiders. C'est parce que VS Code Insiders gère les `vscode-insiders://` URI et non les `URI. vscode://`

Autorisations pour Amazon CodeCatalyst

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

- Définissez vos propres autorisations d'accès pour votre `~/.ssh/config` fichier à `read` et `write`. Limitez `write` les autorisations pour tous les autres utilisateurs.
- Définissez vos autorisations d'accès pour les `~/.ssh/id_rsa` fichiers `~/.ssh/id_dsa` et `read` uniquement sur. `readRestriction write` et `execute` autorisations pour 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 depuis le Toolkit for VS Code

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

Vous `~/.ssh/config` avez une `aws-devenv-*` section qui n'est peut-être plus à jour.

- Choisissez la configuration ouverte. 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 `Host aws-devenv-*` au `~/ .ssh/config`. Fermez ensuite le fichier.
- Réessayez de vous connecter à un environnement de développement à partir du Toolkit for VS Code.

Utilisation avec Amazon API Gateway

Vous pouvez parcourir et exécuter des ressources API Gateway distantes dans votre appareil connecté AWS avec AWS Toolkit for Visual Studio Code.

Note

Cette fonction ne prend pas en charge le débogage.

Pour parcourir et exécuter des ressources API Gateway distantes

1. Dans 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 `Invoquer le AWS`.
3. Dans l'écran Paramètres, spécifiez les paramètres d'appel.
4. Pour exécuter la ressource API Gateway distante, choisissez `Invoquer`. Les résultats sont affichés dans le `Sortie de code VS` affichage.

Utilisation AWS App Runner avec AWS Toolkit for Visual Studio Code

[AWS App Runner](#) fournit un moyen rapide, simple et économique de déployer directement à partir du code source ou d'une image de conteneur vers une application Web évolutive et sécurisée dans le AWS cloud. En l'utilisant, vous n'avez pas besoin d'apprendre de nouvelles technologies, de

choisir le service informatique à utiliser ou de savoir comment approvisionner et configurer les AWS ressources.

Vous pouvez l'utiliser AWS App Runner pour créer et gérer des services basés sur une image source ou 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 compte. AWS
- 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 le prend en charge en [GitHub](#) tant que fournisseur de référentiel de code source.

Prérequis

Pour interagir avec App Runner à l'aide AWS Toolkit for Visual Studio Code de ce qui suit :

- 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 AWS Identity and Access Management (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 avec la palette de commandes de VS Code.

Note

Le rôle `AWSServiceRoleForAppRunner` lié au 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 Amazon Elastic Container Registry (Amazon ECR) image push.

Vous n'avez pas besoin de créer manuellement un rôle lié au service . Lorsque vous créez un rôle AWS App Runner dans AWS Management Console 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 déployez 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 pour cette AWS région.
- Cliquez avec le bouton droit de la souris sur un référentiel et choisissez Copy Repository URI (Copier l'URI du référentiel) pour copier le lien dans votre presse-papier.

Vous spécifiez l'URI du référentiel d'images lors de la configuration de votre instance de service avec 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, ce code doit être stocké dans un dépôt Git géré par un fournisseur de dépôt compatible. 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 sur GitHub), vous devez fournir à App Runner les informations de connexion.

⚠ Important

Pour créer des GitHub connexions, vous devez utiliser la console App Runner (<https://console.aws.amazon.com/apprunner>) pour créer une connexion liée GitHub à AWS. Vous pouvez sélectionner les connexions disponibles sur la page des GitHubconnexions lors de la configuration de votre instance de service avec 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 à votre code de se générer et de s'exécuter. AWS App Runner prend actuellement en charge les environnements d'exécution suivants :

- Exécution gérée par Python
- Exécution gérée par Node.js

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 Command Palette (Palette de commande) ou spécifier un [fichier de configuration App Runner](#) au format YAML. 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 [Tarification d'AWS App Runner](#).

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 en utilisant l'AWS explorateur et 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 l'Explorateur, s'il n'est pas déjà ouvert.
2. Cliquez avec le bouton droit de la souris sur le nœud App Runner et choisissez Create Service (Créer un 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 le prend en charge 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, ce qui est indiqué lorsque l'icône de l'en-tête de la palette de commandes est traversée par une ligne 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 de la 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 devez disposer du rôle d'accès ECR, AppRunnerECR AccessRole, qui permet à App Runner d'accéder aux images Amazon Elastic Container Registry (Amazon ECR) de votre compte. Cliquez sur 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 Sélectionner une connexion, choisissez une connexion 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 référentiel distant, choisissez ou entrez une URL pour le référentiel 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 Explorateur 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 l'Explorateur, s'il n'est pas déjà ouvert.
2. Développez App Runner pour afficher la liste des services.
3. Cliquez avec le bouton droit de la souris sur 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 l'Explorateur, s'il n'est pas déjà ouvert.
2. Développez App Runner pour afficher la liste des services.
3. Cliquez avec le bouton droit de la souris sur votre service et choisissez Resume (Reprendre).

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 l'Explorateur, s'il n'est pas déjà ouvert.
2. Développez App Runner pour afficher la liste des services.
3. Cliquez avec le bouton droit de la souris sur votre service et cliquez sur Start Deployment (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 qu'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. Cliquez avec le bouton droit de la souris sur un groupe de journaux et choisissez View Log Streams (Afficher les flux de journaux).
4. Dans la palette de commandes, choisissez un flux de journal dans le groupe.

L'éditeur VS Code affiche la liste des événements du 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 l'Explorateur, s'il n'est pas déjà ouvert.
2. Développez App Runner pour afficher la liste des services.

3. Cliquez avec le bouton droit de la souris sur un service et choisissez Delete Service (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.

AWSCompositeur d'applications

Vous pouvez utiliser le AWS Toolkit for Visual Studio Code pour utiliser le service AWS Application Composer. AWS Application Composer est un générateur visuel pour les AWS applications qui aide à concevoir l'architecture de votre application et à visualiser votre AWS CloudFormation infrastructure.

Pour des informations détaillées sur le service AWS Application Composer, consultez le guide de l'utilisateur d'[AWSApplication Composer](#).

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

Rubriques

- [Utilisation d'AWSApplication Composer à partir de la boîte à outils](#)

Utilisation d'AWSApplication Composer à partir de la boîte à outils

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

Pour obtenir des informations détaillées sur le service AWS Application Composer, des informations de démarrage et des didacticiels, consultez le Guide de l'utilisateur du [service AWS Application Composer](#).

Les sections suivantes décrivent comment accéder au service AWS Application Composer depuis leAWS Toolkit for Visual Studio Code.

Accès à AWS Application Composer depuis le kit d'outils

Il existe 3 manières principales d'accéder à AWS Application Composer à partir du Toolkit.

Accès à AWS Application 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 Application Composer situé dans le coin supérieur droit de la fenêtre de l'éditeur.
3. AWSApplication Composer ouvre et visualise votre fichier modèle dans la fenêtre de l'éditeur de code VS.

Accès à AWS Application 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 Application Composer.
2. Dans le menu contextuel, choisissez l'option Ouvrir avec App Composer.
3. AWSApplication Composer ouvre et visualise votre fichier modèle dans une nouvelle fenêtre de l'éditeur VS Code.

Accès à AWS Application 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 Application Composer** et choisissez AWSApplication Composer lorsqu'il renseigne les résultats.
3. Choisissez le fichier modèle que vous souhaitez ouvrir, AWS Application Composer ouvre et visualise votre fichier modèle dans une nouvelle fenêtre de l'éditeur VS Code.

AWS CDKpour VS Code

Ceci est une documentation préliminaire pour une fonctionnalité en version préliminaire. Elle est susceptible d'être modifiée.

Le AWS CDK service vous permet de travailler avec [AWS Cloud Development Kit \(AWS CDK\)](#) applications, ou applications. Vous trouverez de plus amples informations sur les AWS CDK dans le [AWS Cloud Development Kit \(AWS CDK\) Manuel du développeur](#).

Les applications AWS CDK sont composées de blocs de construction appelés [constructs](#), qui incluent des définitions pour vos piles AWS CloudFormation et les ressources AWS. À l'aide d'AWS CDK Explorer, vous pouvez visualiser les [piles](#) et les [ressources](#) qui sont définies dans les constructs AWS CDK. Cette visualisation est fournie dans une vue sur les arbres dans le volet Developer Tools de l'éditeur de code Visual Studio (VS Code).

Cette section fournit des informations sur la façon d'accéder à AWS CDK dans l'éditeur VS Code. Il suppose que ce qui suit a déjà été [installé et configuré](#) le Toolkit for VS Code pour votre IDE local.

Rubriques

- [Utilisation d'AWS CDK Applications](#)

Utilisation d'AWS CDK Applications

Ceci est une documentation préliminaire pour une fonctionnalité en version préliminaire. Elle est susceptible d'être modifiée.

Utilisation de l'AWS CDK Explorer dans le AWS Toolkit for VS Code pour visualiser et utiliser AWS CDK Applications.

Prérequis

- Assurez-vous que votre système respecte les conditions préalables spécifiées dans [Installation d'AWS Toolkit for VS Code](#).
- Installer le AWS CDK interface de ligne de commande, comme décrit dans les premières sections de [Mise en route avec les AWS CDK](#) dans le AWS Cloud Development Kit (AWS CDK) Manuel du développeur.

Important

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

Visualisation d'une consoleAWS CDKcandidature

Utilisation deAWSToolkit for VS CodeAWS CDKExplorer, vous pouvez gérer le

[piles](#)etressourcesqui sont stockés dans les constructions CDK de vos applications. LeAWS CDKL'Explorateur affiche vos ressources dans une arborescence à l'aide des informations définies dans letree.json, qui est créé lorsque vous exécutez le plugin**cdk synth**commande. Letree.jsonse trouve dans le dossier d'une applicationcdk.outrépertoire, par défaut.

Pour commencer à utiliser le ToolkitAWS CDKExplorer, vous devrez créer une application CDK.

1. Effectuez les premières étapes de la[Didacticiel Hello World](#)situé dans le plugin[AWS CDKManuel du développeur](#).

Important

Lorsque vous arrivez à l'étape du tutorielDéployer la pile, arrêtez-vous et revenez à ce guide.

Note

Vous pouvez exécuter les commandes fournies dans le didacticiel, par exemple,**mkdiretcdk init**, sur la ligne de commande du système d'exploitation ou dans unTerminaldans l'éditeur VS Code.

2. Après avoir suivi les étapes requises du didacticiel CDK, ouvrez le contenu CDK que vous avez créé dans l'éditeur VS Code.
3. À partir d'AWSAWSvolet de navigation, développez le pluginCDK (version préliminaire)rubrique. 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 VS Code Editor, vous pouvez charger plusieurs dossiers simultanément. Chaque dossier peut contenir plusieurs applications CDK, comme indiqué dans l'image précédente. AWS CDK Explorer recherche les applications dans le répertoire racine du projet et 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 cette commande à nouveau pour voir les modifications reflétées dans l'arborescence.

Effectuer d'autres opérations sur un AWS CDK appli

Vous pouvez utiliser VS Code Editor 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'un VS Code Terminal fenêtré.

Afin d'essayer ces types d'actions, utilisez VS Code Editor pour poursuivre la procédure [Didacticiel Hello World](#) dans le [AWS CDK Manuel du développeur](#). Assurez-vous d'effectuer la dernière étape, [Détruire les ressources de l'application](#), afin de ne pas engager de coûts imprévus pour votre AWS.

Utilisation d'AWS CloudFormation piles

AWS Toolkit for Visual Studio Code prend en charge les piles [AWS CloudFormation](#). À l'aide du Toolkit for VS Code, vous pouvez effectuer certaines tâches avec AWS CloudFormation piles, par exemple les supprimer.

Rubriques

- [Suppression d'un AWS CloudFormation empiler](#)
- [Créez un AWS CloudFormation modèle à l'aide du AWS Toolkit for Visual Studio Code](#)

Suppression d'un AWS CloudFormation empiler

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

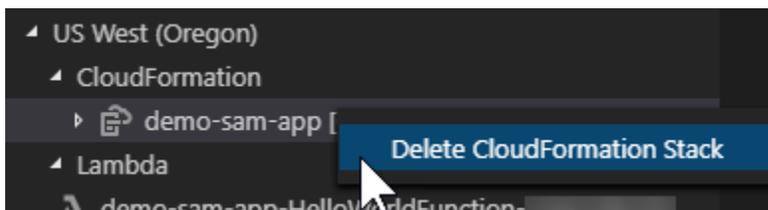
Prérequis

- Assurez-vous que votre système respecte les conditions préalables spécifiées dans [Installation d'AWS Toolkit for VS Code](#).

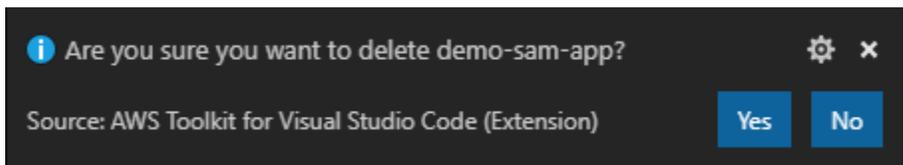
- Assurez-vous que les informations d'identification que vous avez configurées dans [Authentification et accès](#) incluent un accès en lecture/écriture approprié au service AWS CloudFormation. Si dans AWS Explorer, sous CloudFormation, un message similaire à « Erreur lors du chargement des ressources CloudFormation » apparaît, vérifiez les autorisations attachées à ces informations d'identification. Les modifications que vous apportez aux autorisations prendront quelques minutes avant d'affecter AWS Explorer dans VS Code.

Supprimer une pile CloudFormation

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



2. Choisissez Supprimer la pile CloudFormation.
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 à l'aide du 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.

Prérequis

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 un accès en lecture/écriture approprié au AWS CloudFormation service.

Note

Si le CloudFormation service 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 encaisser 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.

Travailler avec CloudWatchSe connecte à l'aide du AWS Toolkit for Visual Studio Code

Amazon CloudWatch Les journaux vous permettent de centraliser les journaux de tous vos systèmes, applications et AWS des services que vous utilisez, dans 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, voir [Qu'est-ce qu'Amazon CloudWatch Des journaux ?](#) dans le Amazon CloudWatch Guide de l'utilisateur.

Les rubriques suivantes décrivent l'utilisation du AWS Toolkit for Visual Studio Code pour travailler avec CloudWatchSe connecte à AWS compte.

Rubriques

- [Affichage des groupes CloudWatch Logs et des flux de journaux en utilisant le AWS Toolkit for Visual Studio Code](#)
- [Utilisation d' CloudWatch Événements de journaux dans les flux de journaux à l'aide de l'AWS Toolkit for Visual Studio Code](#)
- [Recherche de groupes de CloudWatch journaux](#)

Affichage des groupes CloudWatch Logs et des flux de journaux en utilisant le AWS Toolkit for Visual Studio Code

Un flux de journaux est une séquence d'événements de journaux qui partagent la même source. Chaque source différente de journaux dans CloudWatch 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, veuillez consulter [Gestion des groupes de journaux et des flux de journaux](#) dans le guide de l'utilisateur Amazon CloudWatch.

Rubriques

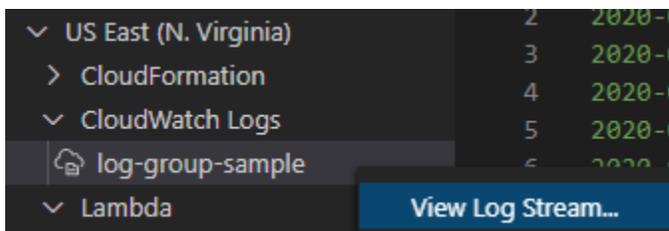
- [Affichage des groupes de journaux et des flux de journaux avec le nœud CloudWatch Logs](#)

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

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

Les groupes de journaux de la région AWS actuels figurent sous le nœud **CloudWatch Logs**.

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



4. De la **Palette de commandes**, sélectionnez un flux de journaux dans le groupe à afficher.

Note

Le **Palette de commandes** affiche un horodatage pour le dernier événement de chaque flux.

Le [Log Stream \(Flux de journaux\) éditeur](#) se lance pour afficher les événements de journal du flux.

Utilisation d' CloudWatch Événements de journaux dans les flux de journaux à l'aide de l'AWS Toolkit for Visual Studio Code

Après avoir ouvert le **Log Stream**, vous pouvez accéder aux événements de journaux dans chaque flux. 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 journaux, le **Log Stream (Flux de journaux) Éditeur** affiche la séquence des événements de journaux du flux.

1. Pour rechercher un flux de journal à afficher, ouvrez l'**Log Stream (Flux de journaux) Éditeur** (voir [Affichage des groupes de journaux et des flux de journaux CloudWatch Logs](#)).

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** : Afficher les événements de journaux les plus récents et les plus anciens en choisissant **Charger de nouveaux événements** ou **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 journaux : Cliquez avec le bouton droit l'onglet Log Stream (Flux de journaux) éditeur et choisissez Copier le nom du flux de journaux.

Note

Vous pouvez également utiliser l'Palette de commandes pour exécuter `AWSToolkit Copier Log Stream Name`.

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

Vous pouvez télécharger le contenu du CloudWatch éditeur de flux de journalisation dans un log sur votre machine locale.

Note

Avec cette option, vous enregistrez dans un fichier uniquement les événements de journaux 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 rechercher un flux de journal à copier, ouvrez l'Flux de journaux éditeur (voir [Affichage des groupes de journaux et des flux de journaux CloudWatch Logs](#)).

2. Cliquez sur l'onglet **Enregistrer** en regard de l'onglet affichant le nom du flux de journaux.

Note

Vous pouvez également utiliser l'**Palette de commandes** pour exécuter **AWSToolkit Enregistrer** le contenu du flux de journaux actuel.

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 [Utilisation des groupes de journaux et des 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](#) des 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), choisissez 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'AWSexplorateur

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

1. À partir de l'AWS Toolkit for Visual Studio Codeexplorateur, 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.

Fonctionnement d'Amazon Elastic Container Registry

Amazon Elastic Container Registry (Amazon ECR) est un service de registre de conteneur AWS géré par, sécurisé et évolutif. Plusieurs fonctions du service Amazon ECR sont accessibles à partir de la Toolkit for VS Code Explorer.

- Création d'un référentiel.
- Création d'un service AWS App Runner pour votre référentiel ou votre image balisée.
- Accès aux URI ou ARN des balises d'image et des référentiels.
- 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 l'AWS interface de ligne de commande et d'autres plateformes avec 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.

Prérequis

Vous devez effectuer ces étapes pour accéder au service Amazon ECR à partir de l'explorateur de code VS.

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'accéder AWS directement à avec des informations d'identification de votre AWS compte root. Utilisez plutôt AWS Identity and Access Management (IAM) pour créer un utilisateur IAM, puis ajoutez-le à un groupe IAM avec des autorisations administratives. Vous pouvez alors accéder à AWS à l'aide d'une URL spéciale et des informations d'identification de l'utilisateur IAM.

Si vous êtes inscrit à AWS, mais que vous n'avez pas créé d'utilisateur IAM pour vous-même, vous pouvez le faire avec la console IAM.

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

Choisissez un moyen de gérer votre administrateur	Pour	Bit	Vous pouvez également
Dans IAM Identity Center (Recommandé)	Utiliser des identifiants à court terme pour accéder à AWS. Telles sont les meilleures 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.	Suivre les instructions de la section Mise en route dans le AWS IAM Identity Center Guide de l'utilisateur.	Configuration de l'accès par programmation en Configurant le AWS CLI à utiliser AWS IAM Identity Center dans le AWS Command Line Interface Guide de l'utilisateur.
Dans IAM (Non recommandé)	Utiliser des identifiants à long terme pour accéder à AWS.	Suivre les instructions relatives à la Création de votre premier groupe utilisateur administrateur et utilisateur IAM dans le Guide de l'utilisateur IAM.	Configuration de l'accès par programmation via la 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` désigne votre numéro de compte AWS sans les traits d'union (par exemple, si votre numéro de compte AWS est `1234-5678-9012`, votre ID de compte AWS est `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 « *vousre_nom_utilisateur @ vousre_id_de_compte_aws* ».

Si vous ne voulez pas que l'URL de votre page de connexion contienne votre ID de compte AWS, vous pouvez créer un alias de compte. Dans le tableau de bord IAM, choisissez Customizer (Personnaliser) et entrez un alias de compte. Il peut s'agir du nom de votre entreprise. Pour plus d'informations, consultez [votre ID de compte AWS 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 AWS Identity and Access Management](#).

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 de Docker Engine](#) et en suivant les instructions.

Installation et configuration de laAWS CLI version 2

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

Rubriques

- [Utilisation du service Amazon Elastic Container Registry dans VS Code](#)

Utilisation du service Amazon Elastic Container Registry dans VS Code

Vous pouvez accéder au service Amazon Elastic Container Registry (Amazon ECR) directement à partir deAWSExplorer dans VS Code et utilisez-le pour envoyer une image de programme vers un référentiel Amazon ECR. Pour démarrer, procédez comme suit :

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 traitement.
3. Créez un référentiel au sein de votre instance Amazon ECR.
4. Transmettez l'image balisée à votre référentiel.

Sections

- [Prérequis](#)
- [1. Création d'un Dockerfile](#)
- [2. Créez votre image à partir de votre Dockerfile](#)
- [3. Créer un nouveau référentiel](#)
- [4. Transmettez, Extraire et Suppression d'images](#)

Prérequis

Avant de pouvoir utiliser la fonctionnalité de service Amazon ECR de Toolkit for VS Code, vous devez respecter ces [exigences](#).

1. Création d'un Dockerfile

Docker utilise un fichier appelé Dockerfile pour définir une image qui peut être poussée 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 Dockerfile

1. Utilisez l'explorateur Toolkit for VS Code pour terminer le répertoire où vous souhaitez enregistrer votre fichier Dockerfile.
2. Créez un nouveau fichier nommé Dockerfile.

Note

VS Code peut vous demander de sélectionner un type de fichier ou une extension de fichier. Si cela se produit, sélectionnez `texte brut`. 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) pour le 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 :

Exemple Modèle d'image 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 de mise à jour des caches de package. Installez les packages logiciels pour le serveur web, et écrivent ensuite le contenu « Hello World ! » à la racine du document du serveur web.

Le `EXPOSE` l'instruction expose le port 80 sur le conteneur, et `CMD` permet de démarrer le serveur web.

3. Enregistrez votre Dockerfile.

Important

Assurez-vous qu'aucune extension n'est associée au nom de votre Dockerfile. 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 envoyer 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 qui contient votre Dockerfile.
2. Exécutez `docker build` pour créer l'image définie dans votre Dockerfile.

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

3. Exécutez `docker images` pour vérifier que l'image a été créée correctement.

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

Exemple Exemple de sortie :

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

4.

Note

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

Pour exécuter la nouvelle image, utilisez la Docker commande.

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

Le -p qui est spécifiée dans l'exemple précédent met en correspondance le Port 80 sur le conteneur Port 80 du système hôte. Si vous exécutez Docker localement, accédez à <http://localhost:80> à l'aide de votre navigateur web. Si le programme s'exécutait correctement, un contenu « Hello World ! » est affichée.

Pour plus d'informations sur la Docker commande, consultez [Documentation de référence sur Docker](#) sur le site web Docker.

3. Créer un nouveau référentiel

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

Créer un nouveau référentiel Amazon ECR

1. À partir du code VS Barre d'activité, choisissez le AWS Icône Toolkit.
2. Développer la AWS Explorateur Menu.
3. Recherchez la valeur par défaut AWS Région associée à votre AWS. Ensuite, sélectionnez-la pour voir la liste des services qui se trouvent via le Toolkit for VS Code.
4. Cliquez sur l'onglet ECR + option pour commencer Créer un nouveau référentiel Processus.
5. Suivez les instructions qui s'affichent afin de finaliser le processus.
6. Une fois qu'il est terminé, vous pouvez accéder à votre nouveau référentiel à partir du ECR Section du AWS Menu Explorer.

4. Transmettez, Extraire et Suppression d'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 `AWSExplorer` avec Docker et `AWS` de commande de commande, vous pouvez effectuer les actions suivantes :

- Extrait une image de votre référentiel.
- Supprimez une image stockée dans votre référentiel.
- Supprimez votre dépôt.

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 auprès de votre registre :

1. Ouvrez un système d'exploitation en ligne de commande connecté à votre instance de `AWS` `COMMANDE`.
2. Utilisation de `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 les **region** et l'**AWS_account_id** aux informations spécifiques à votre `AWS`.

Balisez et transmettez une image à votre référentiel

Après avoir authentifié Docker auprès de votre instance de `AWS`, Transmettez une image à votre référentiel.

1. Utilisation de `Images Docker` pour afficher les images que vous avez stockées localement et identifier celle que vous souhaitez baliser.

```
docker images
```

Exemple Exemple de sortie :

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

2. Étiquetez votre image avec leBalisDockercommande.

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

3. Transmettez l'image balisée à votre référentiel avec le contenuBalisDockercommande.

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

Exemple 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 téléchargée avec succès dans votre référentiel, elle est visible dans leAWSMenu Explorer.

Extrait une image d'Amazon ECR

- Vous pouvez extraire une image vers votre instance locale deBalisDockercommande.

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

Exemple 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 laAWSExplorer.

1. À partir de laAWSExplorateur, développez leECRmenu
2. Développez le référentiel dans lequel 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. Cliquez sur l'ongletSuppression de balise...option pour supprimer toutes les images stockées associées à cette balise

Suppression d'une image en utilisant laAWSCLI

- Vous pouvez également supprimer une image de votre référentiel à l'aide de laAWSecr batch-delete-imagecommande.

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

Exemple 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 dépôt de VS Code. La première méthode consiste à utiliser la `AWSExplorer`.

1. À partir de la `AWSExplorateur`, développez le `ECRmenu`
2. Choisissez pour terminer le contenu que vous souhaitez supprimer en ouvrant le menu contextuel (clic droit)
3. Cliquez sur l'onglet `Suppression du référentiel...option` pour le référentiel choisi

Suppression d'un référentiel Amazon ECR de la `AWSCLI`

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

Note

Par défaut, vous ne pouvez pas supprimer un référentiel qui contient des images. Cependant, le `--force` flag permet cela.

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

Exemple Exemple de sortie :

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

Utilisation d'Amazon Elastic Container Service

Le [AWS Toolkit for Visual Studio Code](#) prend en charge [Amazon Elastic Container Service \(Amazon ECS\)](#). Le Toolkit for VS Code vous aide dans certaines tâches connexes à Amazon ECS, telles que la création de définitions de tâches.

Rubriques

- [Utilisation des fichiers de définition de tâches IntelliSense pour Amazon ECS](#)
- [Amazon Elastic Container Service Exec dans AWS Toolkit for Visual Studio Code](#)

Utilisation des fichiers de définition de tâches IntelliSense pour Amazon ECS

Lorsque vous travaillez avec Amazon Elastic Container Service (Amazon ECS), vous pouvez créer des définitions de tâche, comme décrit dans la section [Création d'une définition de tâche](#) à partir du [Guide d'AWS Elastic Container Service](#). Lorsque vous installez le [AWS Toolkit for Visual Studio Code](#), l'installation inclut la fonctionnalité IntelliSense pour les fichiers Amazon ECS de définition de tâche.

Prérequis

- Assurez-vous que votre système respecte les conditions préalables spécifiées dans [Installation d'AWS Toolkit for VS Code](#).

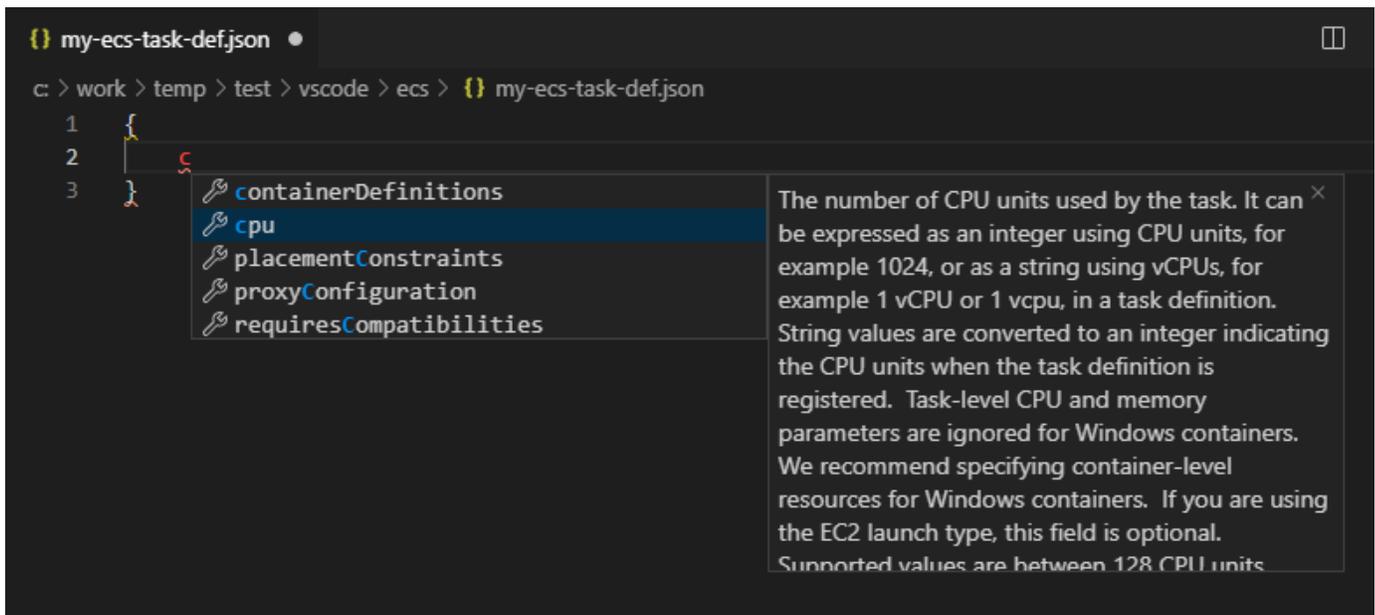
Utiliser IntelliSense dans les fichiers de définition de tâche Amazon ECS

L'exemple suivant vous montre comment tirer parti d'IntelliSense dans les fichiers de définition de tâche Amazon ECS.

- Créez un fichier JSON pour votre définition de 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`

- Ouvrez le fichier dans un éditeur de code VS et entrez les accolades initiales.
- Entrez la lettre « c » comme si vous vouliez ajouter `cpu` à la définition. Observez la boîte de dialogue IntelliSense qui s'ouvre, semblable 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 (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. Vous trouverez des informations supplémentaires dans le guide du développeur relatif aux [AMI optimisées pour Amazon ECS](#).
- 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.

Utilisation des outils Amazon EventBridge

LeAWS Toolkit for Visual Studio Code(VS Code) prend en charge les[Amazon EventBridge](#). En utilisant Toolkit for VS Code, vous pouvez utiliser certains aspects d'EventBridge, tels que les schémas.

Rubriques

- [Utilisation d'Amazon EventBridge Schemas](#)

Utilisation d'Amazon EventBridge Schemas

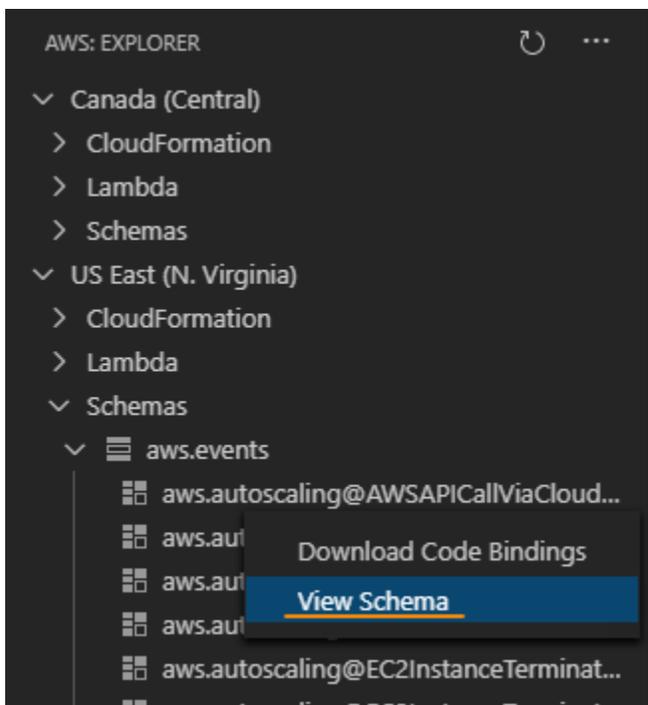
Vous pouvez utiliser le AWS Toolkit for Visual Studio Code (VS Code) pour effectuer diverses opérations sur les [EventBridge schémas Amazon](#).

Prérequis

- 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).

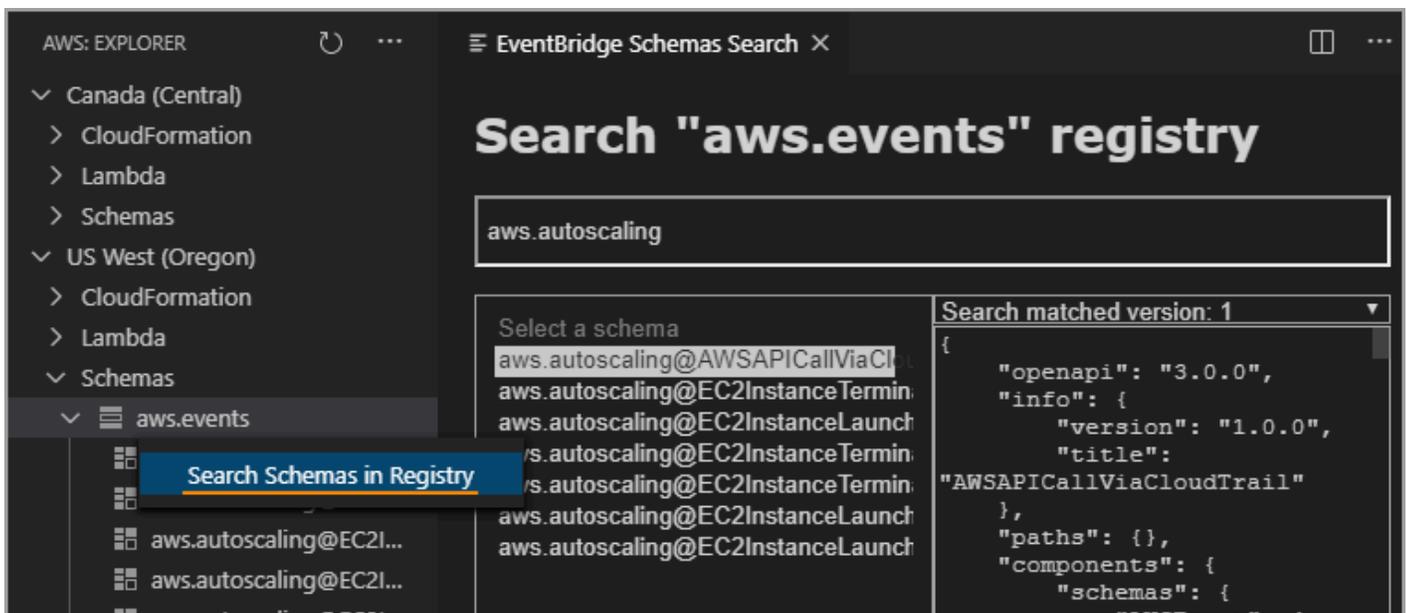


Trouver un schéma disponible

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 contrôles de politique [AWS d'Identity and Access Management \(IAM\) Access Analyzer](#) sur vos politiques IAM créées dans des AWS CloudFormation modèles, des plans Terraform et des documents de politique JSON, à l'aide de l'IAM Access Analyzer dans le. AWS Toolkit for Visual Studio Code

Les vérifications des politiques d'IAM Access Analyzer incluent la validation des politiques et des vérifications de politiques personnalisées. La validation des politiques permet de valider vos politiques IAM conformément aux normes détaillées dans les rubriques [Grammaire du langage de politique IAM JSON](#) et [Bonnes pratiques de AWS sécurité dans l'IAM](#), disponibles dans le guide de l'AWS Identity and Access Management utilisateur. Les résultats de votre validation des politiques incluent des avertissements de sécurité, des erreurs, des avertissements généraux et des suggestions de politiques.

Vous pouvez également effectuer des vérifications de politique personnalisées 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 de détails sur les vérifications des politiques d'IAM Access Analyzer, consultez la rubrique [Contrôles de validation des politiques](#) du Guide de l'AWS Identity and Access Management utilisateur.

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

Rubriques

- [Utilisation d' AWS IAM Access Analyzer](#)

Utilisation d' AWS IAM Access Analyzer

Les sections suivantes décrivent comment effectuer la validation des politiques IAM et les vérifications de politiques personnalisées dans le AWS Toolkit for Visual Studio Code. Pour plus de détails, consultez les rubriques suivantes du guide de l' AWS Identity and Access

Management utilisateur : [validation des politiques IAM Access Analyzer et vérifications des politiques personnalisées par IAM Access Analyzer](#).

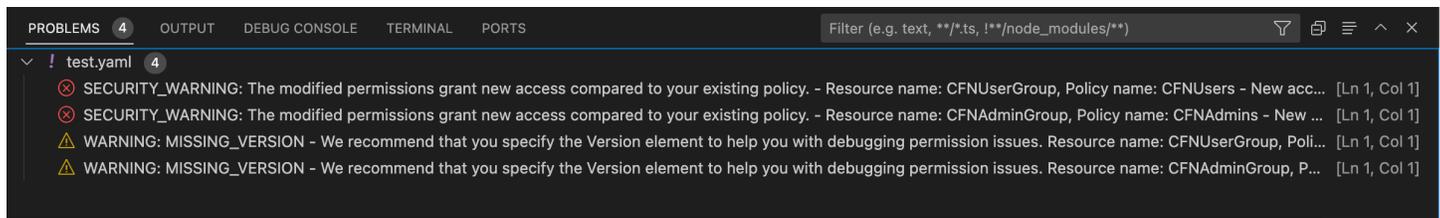
Prérequis

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

- Installez Python version 3.6 ou ultérieure.
- Installez le validateur de [politique IAM pour AWS CloudFormation ou le validateur de politique IAM pour Terraform](#) requis par les outils Python CLI 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 AWS 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 de problèmes de VS Code.



IAM Access Analyzer propose 3 types de contrôles :

- Valider la politique
- CheckAccessNotGranted
- CheckNoNewAccess

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 : AWS CloudFormation modèles, plans Terraform et documents de politique JSON

Exécution de la politique de validation

Le contrôle de validation des politiques, 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 les rubriques [Grammaire du langage de politique IAM JSON](#) et [Bonnes pratiques de AWS sécurité dans IAM](#), disponibles dans le guide de l'AWS Identity and Access Management utilisateur.

1. Dans VS Code, ouvrez un fichier pris en charge contenant des politiques AWS IAM dans l'éditeur VS Code.
2. Pour ouvrir les vérifications de politique IAM Access Analyzer, ouvrez la palette de commandes VS Code en appuyant sur **CRTL+Shift+P**, recherchez **IAM Policy Checks**, puis en cliquant 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 visant à vérifier que des actions IAM spécifiques ne sont pas autorisées par votre politique.

Note

Des frais sont associés à chaque vérification de politique personnalisée. Pour plus de détails sur les tarifs personnalisés de vérification des politiques, consultez le guide de [tarification d'AWS IAM Access Analyzer](#).

1. Dans VS Code, ouvrez un fichier pris en charge contenant des politiques AWS IAM dans l'éditeur VS Code.
2. Pour ouvrir les vérifications de politique IAM Access Analyzer, ouvrez la palette de commandes VS Code en appuyant sur **CRTL+Shift+P**, recherchez **IAM Policy Checks**, puis en cliquant 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 Custom Policy Checks, sélectionnez `CheckAccessNotGranted`.
5. Dans le champ de saisie de texte, entrez une liste d'actions séparées par des virgules. Vous pouvez également fournir un chemin de fichier pointant vers une liste d'actions.

Note

Le chemin du fichier est généralement fourni 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.

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 PASS ou un FAIL résultat.
8. Mettez à jour votre politique et répétez cette procédure en réexécutant le `CheckAccessNotGranted` chèque jusqu'à ce qu'il soit renvoyé PASS.

Courir `CheckNoNewAccess`

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

Note

Des frais sont associés à chaque vérification de politique personnalisée. Pour plus de détails sur les tarifs personnalisés de vérification des politiques, consultez le guide de [tarification d'AWS IAM Access Analyzer](#).

1. Dans VS Code, ouvrez un fichier pris en charge contenant des politiques AWS IAM dans l'éditeur VS Code.
2. Pour ouvrir les vérifications de politique IAM Access Analyzer, ouvrez la palette de commandes VS Code en appuyant sur **CRTL+Shift+P**, recherchez **IAM Policy Checks**, puis en cliquant 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 Custom Policy Checks, 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.

Note

Le chemin du fichier est généralement fourni 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.

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 PASS ou un FAIL résultat.
9. Mettez à jour votre politique et répétez cette procédure en réexécutant le CheckNoNewAccess chèque jusqu'à ce qu'il soit renvoyé PASS.

Utilisation d'AWS IoT dans AWS Toolkit for Visual Studio Code

AWS IoT dans AWS Toolkit for Visual Studio Code vous permet d'interagir avec le service AWS IoT, tout en minimisant les interruptions de votre flux de travail dans VS Code. Ce guide de l'utilisateur est destiné à vous familiariser avec les fonctionnalités AWS IoT qui sont disponibles dans le AWS Toolkit for Visual Studio Code. Pour plus d'informations sur le service AWS IoT, consultez le guide du développeur [Présentation d'AWS IoT?](#)

Conditions préalables requises AWS IoT

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

- Pour les exigences relatives au compte et les autorisations utilisateur spécifiques à AWS IoT, consultez le [Démarrer avec AWS IoT Core](#) manuel du développeur.
- Pour connaître les exigences spécifiques à l'AWS Toolkit for VS Code, consultez le [Configuration de l'AWS Toolkit for VS Code](#) Guide de l'utilisateur.

AWS IoT Objets

AWS IoT connecte des appareils à des services et ressources cloud. Vous pouvez connecter vos appareils à AWS IoT en utilisant des objets appelés choses. 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 sur AWS IoT, consultez le guide du développeur [Gestion des appareils avec AWS IoT](#).

Gestion d'AWS IoT choses

Le Toolkit for VS Code possède plusieurs fonctionnalités qui rendent votre gestion des objets AWS IoT plus efficace. Vous pouvez utiliser la boîte à outils VS Code de différentes manières pour gérer vos AWS IoT things :

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

Pour créer un objet

1. De à partir deAWSExplorateur, développez leIoT en-tête de service et sélection contextuelle (clic droit droit)Objets.
2. ChoisissezCréation d'un objetdans le menu contextuel pour ouvrir une boîte de dialogue.
3. Suivez l'invite en saisissant un nom pour votre objet IoT dans la sectionNom de l'objet.
4. Une fois l'opération terminée, unthingsuivi du nom que vous avez spécifié sera visible dans leObjetSection.

Attacher un certificat à un objet

1. De à partir deAWSExplorateur, développez leIoTsection de service.
2. SousObjets, recherchez la sous-sectionchoseoù vous joignez le certificat.
3. Sélectionnez contextuelle (clic droit) lechoseet choisissezAttacher un certificatdans le menu contextuel, pour ouvrir un sélecteur d'entrée contenant la liste de vos certificats.
4. Dans la liste, choisissez laID de certificatqui correspond au certificat que vous souhaitez joindre à votre objet.
5. Une fois cette opération terminée, votre certificat est accessible dans leAWSexplorateur, en tant qu'élément de l'objet auquel vous l'avez attaché.

Pour détacher un certificat d'un objet

1. De à partir deAWSExplorateur, développez leIoTsection de service
2. DansObjets, recherchez la sous-sectionchose dont vous voulez détacher un certificat.
3. Sélectionnez contextuelle (clic droit) lechoseet choisissezDétachez un certificatdans le menu contextuel.
4. Une fois cette opération terminée, le certificat détaché ne s'affichera plus sous cet élément dans leAWSExplorateur, mais il sera toujours accessible depuis leCertificatssous-section.

Pour supprimer un objet

1. De à partir deAWSExplorateur, développez leIoTsection de service.
2. DansObjets, recherchez la sous-sectionchose que vous souhaitez supprimer.

3. Sélectionnez contextuelle (clic droit) sur l'objet et choisissez Supprimer un objet dans le menu contextuel pour le supprimer.
4. Une fois cette opération terminée, l'élément supprimé ne sera plus disponible dans l'Objets sous-section.

Note

Remarque : Vous ne pouvez supprimer qu'un élément auquel aucun certificat n'est attaché.

AWS IoT Certificats

Les certificats sont 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 certificats, consultez le guide du développeur [Authentification \(IoT\)](#).

Gestion de certificats

La boîte à outils VS Code vous permet de gérer vos AWS IoT certificats, directement auprès du AWSExplorateur.

- [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. De à partir de AWSExplorateur, développez le IoT section de service et sélection contextuelle (clic droit) Certificats.
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 key pair RSA et votre certificat X.509.

 Note

- Les noms de fichiers par défaut contiennent l'ID du certificat en tant que préfixe.
- Seul le certificat X.509 est stocké avec votreAWScompte, par le biais duAWS IoTservice.
- Votre key pair 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 key pair ne peuvent pas être enregistrés dans votre système de fichiers pour le moment, leAWSToolkit supprime le certificat de votreAWS.

Pour modifier le statut d'un certificat

L'état d'un certificat individuel est affiché à côté de son ID dans leAWSExplorateur et peut être défini sur : actif, inactif ou révoqué.

 Note

- Votre certificat a besoin d'unactifavant de pouvoir l'utiliser pour connecter votre appareil à votreAWS IoTservice.
- UninactifLe certificat peut être activé, qu'il ait déjà été désactivé ou qu'il soit inactif par défaut.
- Un certificat qui a étérévoquéne peut pas être réactivé.

1. De à partir deAWSExplorateur, développez leIoTsection de service.
2. DansCertificats, recherchez le certificat que vous voulez modifier.
3. Sélectionnez contextuellement (clic droit) le certificat pour ouvrir un menu contextuel qui affiche les options de changement de statut disponibles pour ce certificat.

- Si un certificat possède le statutinactif, choisissezactiverpour passer l'état àactif.

- Si un certificat possède le statut **actif**, choisissez **désactiver** pour passer l'état à **inactif**.
- Si un certificat possède un statut **inactif**, choisissez **révoquer** pour passer l'état à **révoqué**.

Note

Chacune de ces actions de changement d'état est également disponible si vous sélectionnez un certificat attaché à un objet alors qu'il est affiché dans le **Objets** sous-section.

Pour associer une stratégie IoT à un certificat

1. De à partir de **AWSExplorateur**, développez le **IoT** section de service.
2. Dans **Certificats**, recherchez le certificat que vous voulez modifier.
3. Sélectionnez contextuelle (clic droit) le certificat et choisissez **Attune une politique** dans le menu contextuel, pour ouvrir un sélecteur d'entrée contenant la liste de vos politiques disponibles.
4. Choisissez la stratégie que vous voulez attacher au certificat.
5. Une fois cette opération terminée, la stratégie que vous avez sélectionnée sera ajoutée au certificat en tant qu'élément de sous-menu.

Pour détacher une stratégie IoT d'un certificat

1. De à partir de **AWSExplorateur**, développez le **IoT** section de service.
2. Dans **Certificats**, recherchez le certificat que vous voulez modifier.
3. Développez le certificat et recherchez la stratégie que vous voulez détacher.
4. Sélectionnez contextuelle (clic droit) la stratégie et choisissez **Detach** dans le menu contextuel.
5. Une fois cette opération terminée, la stratégie ne sera plus un élément accessible à partir de votre certificat, mais elle sera disponible depuis le **Stratégies** sous-section.

Pour supprimer un certificat

1. De à partir de **AWSExplorateur**, développez le **IoT** rubrique de service.
2. Dans **Certificats**, recherchez le certificat que vous voulez supprimer.

3. Sélectionnez contextuelle (clic droit) le certificat et choisissez Supprimer un certificat dans le menu contextuel.

 Note

Vous ne pouvez pas supprimer un certificat si celui-ci est attaché à un objet ou s'il a un statut actif. Vous pouvez supprimer un certificat auquel sont associées des stratégies.

Stratégies AWS IoT

AWS IoT Les principales politiques sont définies dans des documents JSON, chacun contenant une ou plusieurs déclarations de stratégie. Les politiques définissent comment AWS IoT, AWS et votre appareil peut ainsi interagir entre eux. Pour plus d'informations sur la création d'un document de stratégie, consultez le guide du développeur [Stratégies IoT](#).

 Note

Les stratégies nommées comptent plusieurs versions afin que vous puissiez les restaurer. Dans The AWS Explorer, vos stratégies IoT sont répertoriées sous le Stratégies sous-section, dans le service IoT. Vous pouvez afficher les versions des stratégies en développant une stratégie. La version par défaut est signalée par un astérisque.

Gestion des politiques

Toolkit for VS Code vous propose plusieurs moyens de gérer votre AWS IoT stratégies de service. Ces méthodes vous permettent de gérer ou de modifier vos politiques directement à partir de la AWS Explorer dans VS Code :

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

Pour créer une stratégie AWS IoT

Note

Vous pouvez créer une nouvelle stratégie à partir de la **AWSExplorer**, mais le document JSON qui définit la stratégie doit déjà exister dans votre système de fichiers.

1. De à partir de **AWSExplorateur**, développez le **IoT** section de service.
2. Sélectionnez contextuelle (clic droit) le **Stratégies** sous-section et choisissez **Créer une stratégie** à partir du document, pour ouvrir **Nom de la politique** champ de saisie.
3. Entrez 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 stratégie, la stratégie sera disponible dans le **AWSExplorateur** lorsque cette opération est terminée.

Pour télécharger un nouveau **AWS IoT** version de stratégie

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

Note

Le nouveau document JSON doit être présent sur votre système de fichiers pour créer une nouvelle version à l'aide de la **AWSExplorateur**.

1. De à partir de **AWSExplorateur**, développez le **IoT** section de service.
2. Développez **Stratégies** pour afficher vos rapports **AWS IoT** politiques
3. Sélectionnez contextuelle (clic droit) la stratégie que vous voulez mettre à jour et choisissez **Créer une 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 stratégie.
5. La nouvelle version sera accessible à partir de votre police dans le **AWSExplorateur**.

Pour modifier unAWS IoTversion de stratégie

Un document de stratégie 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 la charger dans votreAWS IoTservice duAWSExplorateur.

1. De à partir deAWSExplorateur, développez leIoTsection de service.
2. DévelopperStratégieset recherchez la stratégie que vous voulez mettre à jour.Créer une stratégie à partir du documentpour ouvrirNom de la politiquechamp de saisie.
3. Développez la stratégie que vous souhaitez mettre à jour, puis sélectionnez contextuellement (clic droit) la version de stratégie que vous souhaitez modifier.
4. ChoisissezAfficherdepuis le menu contextuel pour ouvrir la version de la stratégie dans VS Code
5. Lorsque le document de stratégie est ouvert, apportez et enregistrez les modifications souhaitées.

Note

À ce stade, les modifications que vous avez apportées à la stratégie sont uniquement enregistrées dans votre système de fichiers local. Pour mettre à jour la version et la suivre à l'aide de laAWSExplorer, répétez les étapes décrites dans la section[Upload a new policy version](#)procédure.

Pour sélectionner une nouvelle version de stratégie par défaut

1. De à partir deAWSExplorateur, développez leIoTsection de service.
2. DévelopperStratégieset recherchez la stratégie que vous voulez mettre à jour.
3. Développez la stratégie que vous souhaitez mettre à jour, puis sélectionnez contextuellement (clic droit) la version de stratégie que vous souhaitez définir et choisissezSet As Default.
4. Une fois cette opération terminée, une étoile apparaît à côté de la nouvelle version par défaut que vous avez sélectionnée.

Pour supprimer des stratégies

Note

Avant de pouvoir supprimer une stratégie ou une version de stratégie, certaines conditions doivent être remplies.

- Vous ne pouvez pas supprimer une stratégie si elle est attachée à un certificat.
- Vous ne pouvez pas supprimer une stratégie dont les versions ne sont pas celles par défaut.
- Vous ne pouvez pas supprimer la version par défaut d'une stratégie à moins qu'une nouvelle version par défaut ne soit sélectionnée ou que la stratégie dans son ensemble ne soit supprimée.
- Avant de pouvoir supprimer l'intégralité d'une stratégie, toutes les versions autres que celles par défaut de cette stratégie doivent d'abord être supprimées.

1. De à partir de **AWSExplorateur**, développez le **IoT** section de service.
2. Développez **Stratégies** et recherchez la stratégie que vous voulez mettre à jour.
3. Développez la stratégie que vous souhaitez mettre à jour, puis sélectionnez contextuellement (clic droit) la version de stratégie que vous souhaitez supprimer, puis choisissez **Supprimer**.
4. Lorsqu'une version est supprimée, elle n'apparaît plus dans l'Explorateur.
5. Lorsque la seule version restante pour une stratégie est la version par défaut, vous pouvez sélectionner en contexte (clic droit) la stratégie parent et choisir **Supprimer** pour le supprimer.

Utilisation des fonctions AWS Lambda

Le AWS Toolkit for Visual Studio Code prend en charge les fonctions [AWS Lambda](#). À l'aide de la boîte Toolkit for VS Code, vous pouvez créer du code pour les fonctions Lambda qui font partie de [applications sans serveur](#). Vous pouvez également appeler les fonctions Lambda localement ou sur AWS.

Lambda est un service de calcul entièrement géré qui exécute votre code en réponse à des événements générés par le code personnalisé ou par les divers services AWS, tels qu'Amazon Simple Storage Service (Amazon S3), Amazon DynamoDB, Amazon Kinesis, Amazon Simple Notification Service (Amazon SNS) et Amazon Cognito.

Rubriques

- [Interaction avec des fonctions Lambda distantes](#)

Interaction avec des fonctions Lambda distantes

À l'aide du Toolkit for VS Code, vous pouvez interagir avec [AWS Lambda](#) de différentes manières, comme cela est décrit dans cette rubrique.

Pour plus d'informations sur Lambda, consultez le [guide du développeur AWS Lambda](#).

Note

Si vous avez déjà créé des fonctions Lambda en utilisant la AWS Management Console ou d'une autre manière, vous pouvez les appeler à partir de la boîte à outils. Pour créer une nouvelle fonction (à l'aide de VS Code) que vous pouvez déployer vers AWS Lambda, vous devez d'abord [Créer une application sans serveur](#).

Rubriques

- [Prérequis](#)
- [Appeler une fonction Lambda](#)
- [Supprimer une fonction Lambda](#)
- [Importer une fonction Lambda](#)
- [Charger une fonction Lambda](#)

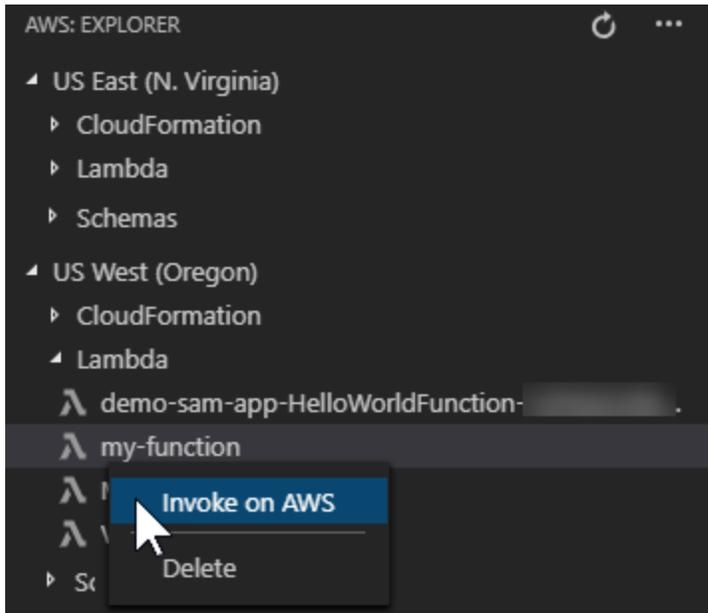
Prérequis

- Assurez-vous que votre système respecte les conditions préalables spécifiées dans [Installation d'AWS Toolkit for VS Code](#).
- Assurez-vous que les informations d'identification que vous avez configurées dans [Authentification et accès](#) incluent un accès en lecture/écriture approprié au service AWS Lambda. Si dans AWS Explorer, sous Lambda, un message similaire à « Erreur lors du chargement des ressources Lambda » apparaît, vérifiez les autorisations attachées à ces informations d'identification. Les modifications que vous apportez aux autorisations prendront quelques minutes avant d'affecter la fonction AWS Explorer dans VS Code.

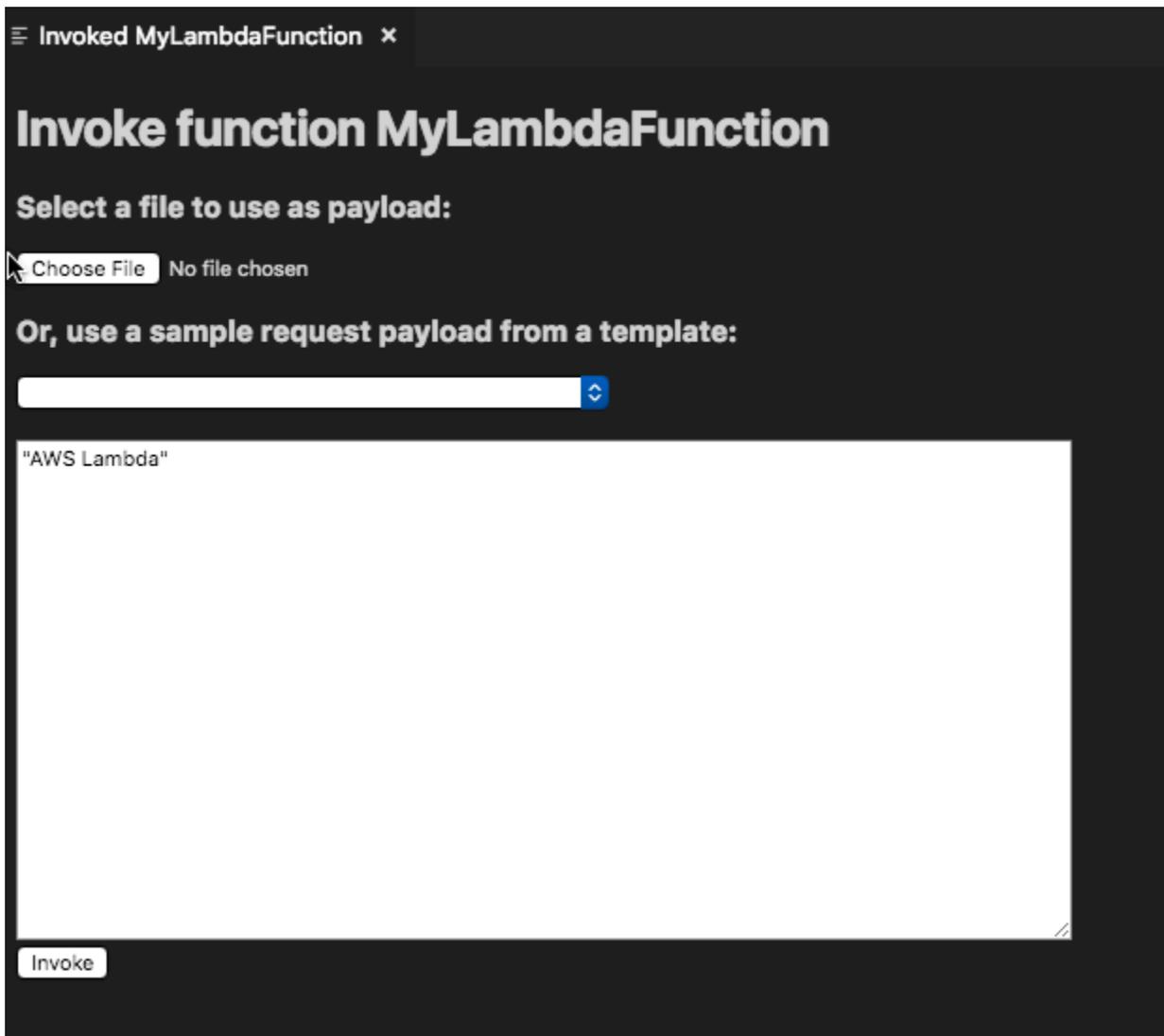
Appeler une fonction Lambda

Vous pouvez appeler une fonction Lambda sur AWS à partir de la Toolkit for VS Code.

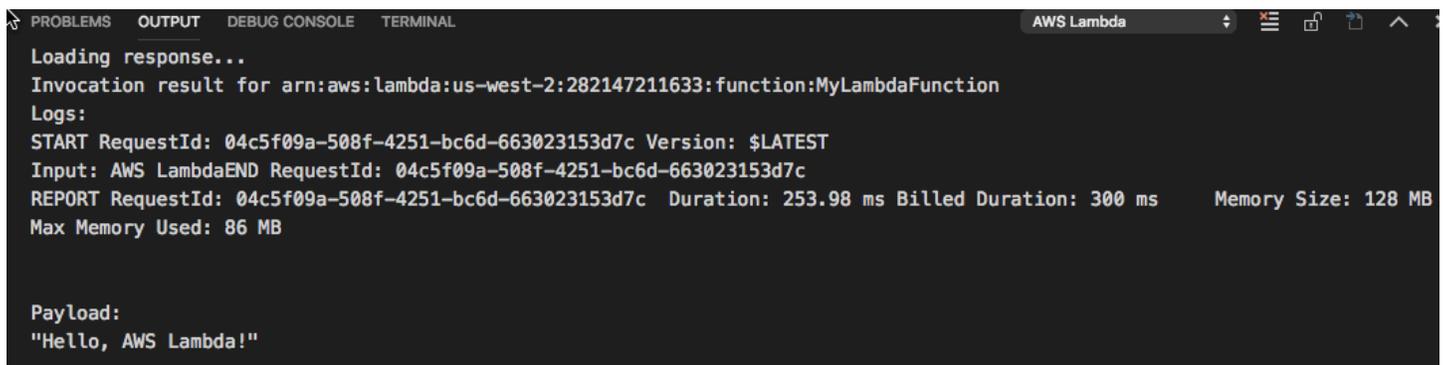
1. Dans AWS Explorer, choisissez le nom de la fonction Lambda à appeler, puis ouvrez son menu contextuel.



2. Choisissez Appeler sur AWS.
3. Dans la fenêtre d'appel qui s'ouvre, saisissez l'entrée dont votre fonction Lambda a besoin. La fonction Lambda peut, par exemple, exiger une chaîne comme entrée, comme le montre la zone de texte.



La sortie de la fonction Lambda s'affiche comme pour tout autre projet utilisant VS Code.



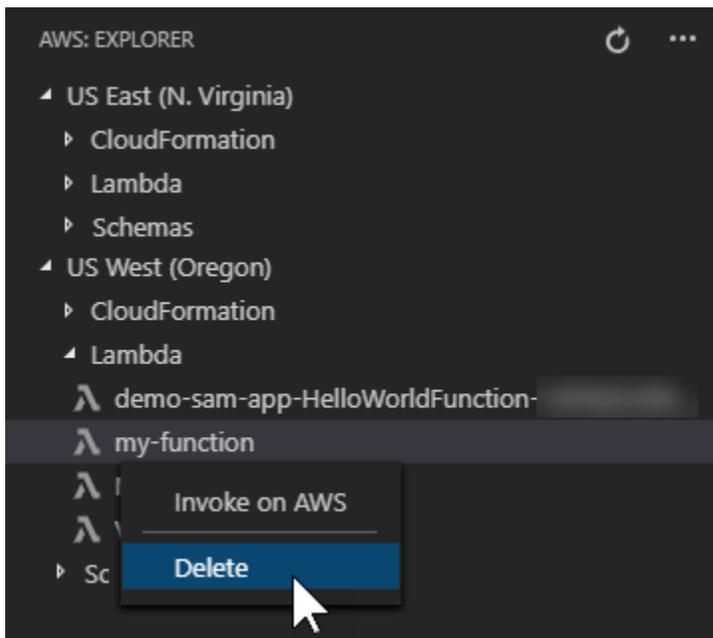
Supprimer une fonction Lambda

Vous pouvez également supprimer une fonction Lambda en utilisant le même menu contextuel.

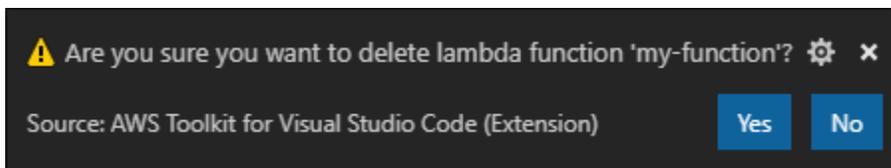
⚠ Warning

N'utilisez pas cette procédure pour supprimer des fonctions Lambda associées à [AWS CloudFormation](#) (par exemple, la fonction Lambda créée lors de la [création d'une application sans serveur](#) dans les pages précédentes de ce guide). Ces fonctions doivent être supprimées via la pile AWS CloudFormation.

1. Dans AWS Explorer, choisissez le nom de la fonction Lambda à supprimer, puis ouvrez son menu contextuel.



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



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

Importer une fonction Lambda

Vous pouvez importer du code depuis une fonction Lambda distante dans votre espace de travail VS Code à des fins de modification et de débogage.

Note

La boîte à outils prend uniquement en charge l'importation de fonctions Lambda à l'aide d'environnements d'exécution Node.js et Python pris en charge.

1. Dans **AWSExplorateur**, choisissez le nom de la fonction Lambda à importer, puis ouvrez son menu contextuel.
2. Choisissez **Importez...**
3. Choisissez un dossier dans lequel importer le code Lambda. Les dossiers situés en dehors de l'espace de travail actuel seront ajoutés à votre espace de travail actuel.
4. Après le téléchargement, le Toolkit ajoute le code à votre espace de travail et ouvre le fichier contenant le code du gestionnaire Lambda. La boîte à outils crée également une configuration de lancement, qui apparaît dans le panneau **Exécution du code VS** afin que vous puissiez exécuter et déboguer localement la fonction Lambda à l'aide de **AWS Serverless Application Model**. Pour plus d'informations sur l'utilisation de **AWS SAM**, consultez [the section called “Exécution et débogage d'une application sans serveur à partir d'un modèle \(local\)”](#).

Charger une fonction Lambda

Vous pouvez mettre à jour les fonctions Lambda existantes avec du code local. La mise à jour du code de cette manière n'utilise pas l'**AWS SAM CLI** pour le déploiement et ne crée pas de pile **AWS CloudFormation**. Cette fonctionnalité peut charger une fonction Lambda avec n'importe quel environnement d'exécution pris en charge par Lambda.

Warning

La boîte à outils ne peut pas vérifier si votre code fonctionne. Assurez-vous que le code fonctionne avant de mettre à jour les fonctions Lambda de production.

1. Dans **AWSExplorateur**, choisissez le nom de la fonction Lambda à importer, puis ouvrez son menu contextuel.
2. Choisissez **Charger une fonction Lambda...**
3. Choisissez parmi les trois options pour charger votre fonction Lambda. Les options incluent :

Chargement d'une archive .zip précréée

- Choisissez **Archive Zip** dans le menu **Quick Pick**.
- Choisissez un fichier .zip dans votre système de fichiers et confirmez le chargement avec la boîte de dialogue modale. Vous chargez ainsi le fichier .zip tel quel et la fonction Lambda est mise à jour immédiatement après le déploiement.

Chargement d'un répertoire tel quel

- Choisissez **Directory** dans le menu **Quick Pick**.
- Choisissez un répertoire dans votre système de fichiers.
- Choisissez **Non** Lorsque vous êtes invité à créer le répertoire, confirmez le chargement avec la boîte de dialogue modale. Vous chargez ainsi le répertoire tel quel, et la fonction Lambda est mise à jour immédiatement après le déploiement.

Création et chargement d'un répertoire

Note

Pour ce faire, la fonction **AWS SAMINTERFACE DE COMMANDE**.

- Choisissez **Directory** dans le menu **Quick Pick**.
- Choisissez un répertoire dans votre système de fichiers.
- Choisissez **Oui** Lorsque vous êtes invité à créer le répertoire, confirmez le chargement avec la boîte de dialogue modale. Vous générez ainsi le code dans le répertoire en utilisant la commande `sam build AWS SAM CLI`, et la fonction Lambda est mise à jour immédiatement après le déploiement.

Note

La boîte à outils vous avertira s'il ne peut pas détecter un gestionnaire correspondant avant le téléchargement. Si vous souhaitez modifier le gestionnaire lié à la fonction Lambda, vous pouvez pour ce faire utiliser la fonction AWS Management Console ou le AWS CLI.

Amazon Redshift dans le kit d'outils pour VS Code

Amazon Redshift est un service d'entreposage de 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 depuis le Toolkit for VS Code](#)

Utilisation d'Amazon Redshift depuis le 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](#).

Démarrer

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 Amazon Redshift pour démarrer.

1. [À partir de votre navigateur Internet préféré, connectez-vous à la console AWS de gestion et ouvrez la console Amazon Redshift à l'adresse https://console.aws.amazon.com/redshift/.](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 de détails 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. [À partir de votre navigateur Internet préféré, connectez-vous à la console AWS de gestion et ouvrez la console Amazon Redshift à l'adresse https://console.aws.amazon.com/redshift/.](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'Écran AWS 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 AWSemplacement 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 AWSemplacement 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 duAWS 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 leAWS 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.

Utilisation avec applications sans serveur

LeAWS Toolkit for Visual Studio Code fournit un support pour [Application sans serveur AWS](#). Les rubriques suivantes décrivent comment commencer à créer et à utiliser des applicationsAWS Serverless Application Model (AWS SAM), à partir duAWS Toolkit for Visual Studio Code.

Rubriques

- [Débuter avec les applications sans serveur](#)
- [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](#)
- [Dépannage des applications sans serveur](#)

Débuter avec les applications sans serveur

Les sections suivantes décrivent comment commencer à créer un à Application sans serveur AWS partir deAWS Toolkit for Visual Studio Code, using AWS Serverless Application Model (AWS SAM) et AWS CloudFormation stacks.

Prérequis

Avant de pouvoir créer ou utiliser unApplication 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 commande AWS SAM (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 l'utilisateur AWS Serverless Application Model.
- Dans votre fichier de configuration AWS, identifiez votre région AWS 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'interface utilisateur AWS Command Line Interface.
- Installez votre SDK de langue et configurez votre chaîne d'outils. Pour plus d'informations sur la configuration de votre chaîne d'outils, consultez la rubrique [Configuration de votre chaîne d'outils](#) dans ce guide de l'utilisateur AWS Toolkit for Visual Studio Code.
- 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 [CodeLens](#) section 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 AWS Identity and Access Management (IAM) nécessaires pour déployer et exécuter des applications sans serveur. Vous devez disposer d'un accès en lecture/écriture approprié aux services suivants : IAM, AWS CloudFormation, Lambda, Amazon API Gateway, Amazon Simple Storage Service (Amazon S3) et Amazon Elastic Container Registry (Amazon ECR).

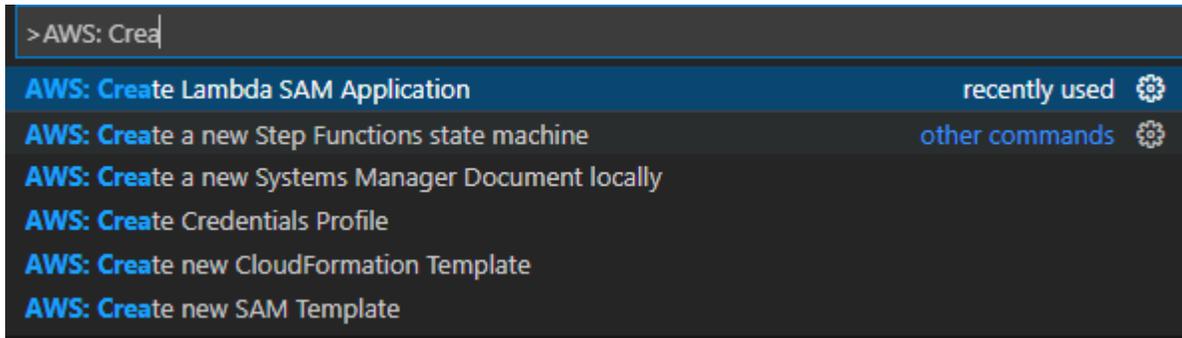
Pour plus d'informations sur la configuration de l'authentification requise pour déployer et exécuter des applications sans serveur, consultez la section [Gestion de l'accès aux ressources et des autorisations](#) dans le Guide du développeur AWS Serverless Application Model. Pour plus d'informations 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 AWSToolkit 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 basé sur ARM64 au lieu de l'environnement `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 basés sur ARM64. Pour plus d'informations, consultez [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-node.js`. 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 [CodeLens](#) fonctionnalité 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 [Prérequis](#) 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 plus d'informations, consultez [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.

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`.

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.

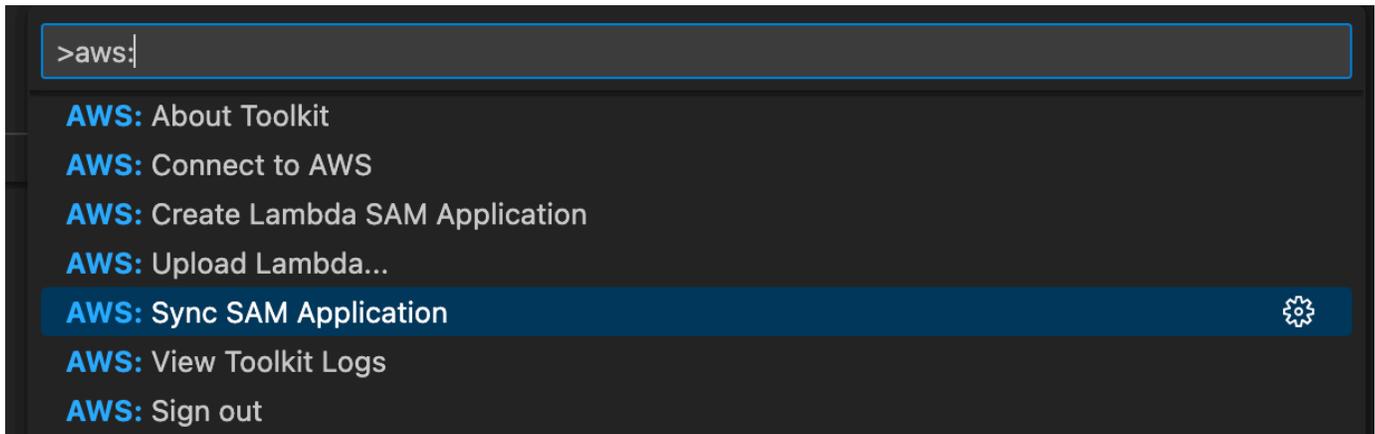
Lorsque les sessions de débogage démarrent, le panneau CONSOLE DE DÉBOGAGE affiche la sortie de débogage et toutes les valeurs renvoyées par la fonction Lambda. (Lors du débogage d'AWS SAM applications, le AWSToolkit est sélectionné comme canal de sortie dans le panneau de sortie.)

Synchronisation des applications AWS SAM

AWS Toolkit for Visual Studio Code exécute la commande AWS SAM CLI `aws 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 avec `aws sam sync` à 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 AWS et 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 `packageImage`, 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.

- Pendant le processus de synchronisation, l'état de votre déploiement est capturé dans l'onglet Terminal de VS Code. Vérifiez que votre synchronisation a bien été effectuée depuis l'onglet du terminal. Si une erreur survient, vous recevez une notification.

A dark grey notification box with a red 'X' icon on the left and the text "Failed to deploy SAM application." in white.

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 Explorer. Pour plus d'informations sur la façon d'invoquer la fonction Lambda créée dans le cadre de l'application, consultez la [Interaction avec des fonctions Lambda distantes](#) 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 pile AWS CloudFormation que vous avez précédemment déployé dans le cloud AWS. Notez que cette procédure ne supprime pas votre répertoire d'application de votre hôte local.

- Ouvrez la [AWS Explorateur](#).
- Dans la fenêtre AWSToolkit Explorer, développez la région contenant l'application déployée que vous souhaitez supprimer, puis développez-la AWS CloudFormation.
- Ouvrez le menu contextuel (clic droit) pour le nom de la AWS CloudFormation pile correspondant à l'application sans serveur que vous souhaitez supprimer, puis choisissez Supprimer la AWS CloudFormation pile.
- 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 AWS CloudFormation liste dans AWSExplorer.

Exécution et débogage de fonctions Lambda directement à partir du code

Lorsque vous testez l'AWS SAMapplication, 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 [CodeLens](#) fonctionnalité 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/Environnement d'exécution	Critères pour que les fonctions Lambda soient identifiées par des indicateurs CodeLens
C# (dotnetcore2.1, 3.1 ; .NET 5.0)	<p>La fonction de recherche dispose des fonctions suivantes :</p> <ul style="list-style-type: none"> • C'est une fonction publique d'une classe publique. • Il possède un ou deux paramètres. Avec deux paramètres, le second doit implémenter l'<code>ILambdaContext</code> interface. • Il possède un <code>*.csproj</code> fichier dans son dossier parent dans le dossier de l'espace de travail VS Code. <p>L'extension ms-dotnettools.csharp (ou toute extension) fournissant des symboles de langage pour C#) est installée et activée.</p>
JavaScript/TypeScript (Node.js 12.x, 14.x)	<p>La fonction de recherche dispose des fonctions suivantes :</p> <ul style="list-style-type: none"> • Il s'agit d'une fonction exportée avec jusqu'à trois paramètres. • Il possède un package <code>.json</code> fichier dans son dossier parent dans le dossier de l'espace de travail VS Code.

Language/Environnement d'exécution	Critères pour que les fonctions Lambda soient identifiées par des indicateurs CodeLens
Python (3,7, 3,8, 3,9, 3,10, 3,11, 3,12)	<p>La fonction de recherche possède les fonctions suivantes :</p> <ul style="list-style-type: none">• Il s'agit d'une fonction de niveau supérieur.• Il possède un <code>requirements.txt</code> fichier dans son dossier parent dans le dossier de l'espace de travail VS Code. <p>L'extension ms-python.python (ou toute extension fournissant des symboles de langage pour Python) est installée et activée.</p>

Language/Environnement d'exécution	Critères pour que les fonctions Lambda soient identifiées par des indicateurs CodeLens
Java (8, 8.al2, 11)	<p>La fonction de recherche dispose des fonctions suivantes :</p> <ul style="list-style-type: none">• C'est une fonction publique d'une classe publique non abstraite.• Il possède un, deux ou trois paramètres :<ul style="list-style-type: none">• Un paramètre : le paramètre peut être n'importe quoi.• Deux paramètres : les paramètres doivent être <code>java.io.InputStream</code> et <code>java.io.OutputStream</code> OU le dernier paramètre doit être <code>com.amazonaws.services.lambda.runtime.Context</code>.• Trois paramètres : les paramètres doivent être <code>java.io.InputStream</code>, <code>java.io.OutputStream</code> et le dernier paramètre doit être <code>com.amazonaws.services.lambda.runtime.Context</code>.• Il possède un fichier <code>build.gradle</code> (Gradle) ou <code>pom.xml</code> (Maven) dans son dossier parent dans le dossier de l'espace de travail VS Code. <p>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.</p>

Language/Environnement d'exécution	Critères pour que les fonctions Lambda soient identifiées par des indicateurs CodeLens
	Le vscjava.vscode-java-debug l'extension (ou toute extension fournissant un débogueur Java) est installée et activée.
Go (1.x)	<p>La fonction de recherche possède les fonctions suivantes :</p> <ul style="list-style-type: none"> • 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 <code>context.Context</code>. • Elle renvoie entre 0 et 2 arguments. S'il y a plus de 0 arguments, le dernier argument doit être implémenté <code>error</code>. • Il contient un <code>go.mod</code> fichier dans le dossier de l'espace de travail VS Code. <p>L'extension golang.go est installée, configurée et activée.</p>

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 `app.js` fichier.
3. Dans l' **CodeLens** indicateur qui identifie un gestionnaire de fonctions Lambda éligible, choisissez **Add Debug Configuration**.
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 "nodejs.12x".
- "payload" – Choisissez l'une des options suivantes pour définir la charge utile d'événement à fournir à votre fonction Lambda comme entrée :
 - "json" : paires clé-valeur au format JSON 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.

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, AWSToolkit 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 AWS SAM ressources locales API Gateway, spécifiées dans `template.yaml`, en exécutant une configuration de lancement VS Code de `type=aws-sam` avec `invokeTarget.target=api`.

Note

API Gateway prend en charge deux types d'API : REST et HTTP. Toutefois, la fonction API Gateway avec la commande AWS Toolkit for Visual Studio Code prend uniquement en charge les API REST. Parfois, les API HTTP sont appelées « API API Gateway V2 ».

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) dans votre AWS SAM, survolez le gestionnaire Lambda et choisissez l'option Ajouter une configuration pour le CodeLens. Ensuite, dans le menu, choisissez l'élément marqué événement API.
 - Option 2 : Modifier launch.json et créez une configuration de lancement à l'aide de la commande 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 code VSRun (Exécuter Lambda), 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. Cliquez sur `Play` (Lire) dans le `Run` (Exécuter Lambda) Panneau.
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 connexion AWS sont fournies. Pour plus d'informations, consultez le `.AWS` propriétés de connexion (« aws ») dans le tableau [Options de configuration pour le débogage d'applications sans serveur](#) Section.

sam

Manière dont la AWS SAM CLI génère l'application. Pour plus d'informations, consultez le `.AWS` Propriétés de la CLI SAM (« sam ») dans le tableau [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` pour modifier les configurations de débogage, vous pouvez utiliser le code VS [IntelliSense](#) pour afficher et compléter automatiquement les propriétés valides. Pour déclencher IntelliSense dans l'éditeur, appuyez sur `Ctrl+Barre d'espace`.

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

IntelliSense vous permet de rechercher et de définir des propriétés pour appeler les fonctions Lambda directement ou avec AWS SAM modèle. Vous pouvez également définir des propriétés pour "lambda" (comment la fonction s'exécute), "sam" (comment la AWS SAM CLI génère l'application) et "aws" (comment les informations de connexion AWS sont fournies).

AWS SAM: appel direct du gestionnaire Lambda a/appel Lambda basé sur un modèle

Propriété	Description
type	Spécifie l'extension qui gère la configuration de lancement. Toujours défini sur <code>aws-sam</code> pour utiliser la AWS SAM CLI pour générer et déboguer localement.
name	Spécifie un nom convivial de lecteur qui doit apparaître dans la liste Configuration du lancement de débogage.

Propriété	Description
<code>request</code>	Spécifie le type de configuration à exécuter par l'extension désignée (<code>aws-sam</code>). Toujours défini sur <code>direct-invoke</code> pour démarrer la fonction Lambda.

Propriété	Description
<code>invokeTarget</code>	<p>Spécifie le point d'entrée pour l'appel de la ressource.</p> <p>Pour appeler directement la fonction Lambda, définissez des valeurs pour les champs <code>invokeTarget</code> :</p> <ul style="list-style-type: none"> • <code>target</code> – Défini sur code. • <code>lambdaHandler</code> – Nom du gestionnaire de fonction Lambda à appeler. • <code>projectRoot</code> — Chemin du fichier d'application contenant le gestionnaire de fonction Lambda. • <code>architecture</code> — Architecture de processeur de l'environnement émulé dans lequel s'exécute votre application SAM Lambda locale. Pour certains environnements d'exécution, vous pouvez choisir <code>arm64</code> et non de la valeur par défaut <code>x86_64</code> architecture. Pour plus d'informations, consultez Création d'une nouvelle application sans serveur (locale). <p>Pour invoquer les ressources Lambda avec AWS SAM modèle, définissez les valeurs suivantes <code>invokeTarget</code> champs :</p> <ul style="list-style-type: none"> • <code>target</code> – Défini sur <code>template</code>. • <code>templatePath</code> — Chemin vers AWS SAM fichier de modèle. • <code>logicalId</code> – Nom de la ressource de la fonction <code>AWS::Lambda::Function</code> ou de la fonction <code>AWS::Serverless::Function</code> à appeler. Vous pouvez trouver le nom de la ressource dans le fichier au format YAML AWS SAM modèle. Remarque : AWS Toolkit reconnaît implicitement les fonctions définies avec <code>PackageType: Image</code> dans le AWS SAM modèle comme Basé sur l'image Fonctions Lambda. Pour plus d'informations, consultez Packages de déploiement M Lambda dans le Guide du développeur AWS Lambda.

Propriétés Lambda ("**lambda**")

Propriété	Description
<code>environmentVariables</code>	<p>Envoyez 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 nom du compartiment comme variable d'environnement.</p> <div data-bbox="591 569 1508 1759" style="border: 1px solid #add8e6; border-radius: 10px; padding: 15px;"><p> Note</p><p>Lorsque vous spécifiez des variables d'environnement pour une application sans serveur, vous devez ajouter des configurations à la fois auAWS SAMmodèle (<code>template.yaml</code>) et le<code>launch.json</code> dans le fichier. Exemple de mise en forme pour une variable d'environnement dans leAWS SAMmodèle :</p><pre>Resources: HelloWorldFunction: Type: AWS::Serverless::Function Properties: CodeUri: hello-world/ Handler: app.lambdaHandlerN10 Runtime: nodejs10.x Environment: Variables: SAMPLE1: Default Sample 1 Value</pre><p>Exemple de mise en forme pour une variable d'environnement dans le<code>launch.json</code> dans le fichier:</p><pre>"environmentVariables": { "SAMPLE1": "My sample 1 value" }</pre></div>
<code>payload</code>	Fournit deux options pour la charge utile d'événement que vous fournissez à votre fonction Lambda en entrée.

Propriété	Description
	<ul style="list-style-type: none">"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 l'environnement d'exécution utilisé pour 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	<p>Spécifie où se trouve le code local par rapport à l'endroit où il s'exécute dans le conteneur.</p> <p>Par défaut, le Toolkit for VS Code définit <code>localRoot</code> à la racine de code de la fonction Lambda dans l'espace de travail local, et <code>remoteRoot</code> pour <code>/var/task</code>, qui est le répertoire de travail par défaut pour le code exécuté dans Lambda. Si le répertoire de travail est modifié dans le Dockerfile ou avec le <code>workingDirectory</code> paramètre dans AWS CloudFormation fichier de modèle, au moins un <code>pathMapping</code> doit être spécifiée afin que le débogueur puisse mapper avec succès les points d'arrêt définis localement avec le code exécuté dans le conteneur Lambda.</p> <p>Exemple de mise en forme pour <code>pathMappings</code> dans le <code>launch.json</code> dans le fichier:</p> <pre data-bbox="597 982 1507 1304">"pathMappings": [{ "localRoot": " \${workspaceFolder}/sam-app/ HelloWorldFunction ", "remoteRoot": " /var/task " }]</pre> <p>Mises en garde :</p> <ul data-bbox="597 1423 1507 1654" style="list-style-type: none">• Pour les fonctions Lambda basées sur des images .NET, <code>remoteRoot</code> L'entrée doit être le répertoire de compilation.• Pour les fonctions Lambda basées sur Node.js, vous ne pouvez spécifier qu'une seule entrée de mappage de chemin.

La Toolkit for VS Code utilise la AWS SAM CLI pour créer et déboguer les applications sans serveur localement. Vous pouvez configurer le comportement des commandes AWS SAM CLI en utilisant les propriétés de la configuration "sam" dans le fichier `launch.json`.

Propriétés AWS SAM CLI ("**sam**")

Propriété	Description	Valeur par défaut
<code>buildArguments</code>	Configure la façon dont la commande <code>sam build</code> 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
<code>containerBuild</code>	Indique si la fonction doit être générée dans un conteneur Docker de type Lambda.	<code>false</code>
<code>dockerNetwork</code>	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 conteneurs Lambda se connectent uniquement au réseau Docker de pont par défaut.	Chaîne vide
<code>localArguments</code>	Spécifie des arguments d'appel local supplémentaires.	Chaîne vide
<code>skipNewImageCheck</code>	Indique si la commande doit ignorer l'extraction de la dernière image Docker pour l'exécution de Lambda.	<code>false</code>
<code>template</code>	Personnalise votreAWS SAMmodèle utilisant des	<code>"parameters":{}</code>

Propriété	Description	Valeur par défaut
	paramètres pour entrer les valeurs client. Pour plus d'informations, consultez Paramètres dans le Guide de l'utilisateur AWS CloudFormation.	

Propriétés de connexion AWS ("aws")

Propriété	Description	Valeur par défaut
<code>credentials</code>	Sélectionne un profil spécifique (par exemple, <code>profile:default</code>) à partir de votre fichier d'informations d'identification pour obtenir les informations d'identification AWS.	LeAWSinformations d'identification que votre partagéAWSfichier de configuration ou partagéAWSfichier d'informations d'identification fournir à la Toolkit for VS Code.
<code>region</code>	Définit la région AWS du service (par exemple <code>us-east-1</code>).	Valeur par défaut de la région AWS associée au profil d'informations d'identification actif.

Exemple : Configuration du lancement du modèle

Voici un exemple de fichier de configuration de lancement pourAWS SAMcible du modèle :

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

Dépannage des applications sans serveur

Cette rubrique contient des informations sur les erreurs courantes que vous pouvez rencontrer lors de la création d'applications sans serveur avec Toolkit for VS Code et comment les résoudre.

Rubriques

- [Comment puis-je utiliser un samconfig.toml avec une configuration de lancement SAM ?](#)

- [Erreur: « Erreur d'exécution : Le conteneur n'existe pas »](#)
- [Erreur : « Docker.Errors.APIError : 500 Erreur de serveur... Vous avez atteint votre limite de taux de traction. »](#)
- [Erreur: « Erreur du serveur 500 : Montage C:\Users \... »](#)
- [À l'aide de WSL, webviews \(par exemple, « Invoke onAWS« formulaire\) sont cassés](#)
- [Débogage d'une application TypeScript, mais les points d'arrêt ne fonctionnent pas](#)

Comment puis-je utiliser un samconfig.toml avec une configuration de lancement SAM ?

Spécifiez l'emplacement de votre interface de ligne de commande SAM [samconfig.toml](#) en configurant le `--config-file` dans l'argument `localArguments` 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: « Erreur d'exécution : Le conteneur n'existe pas »

Le `sam build` peut afficher cette erreur si votre système ne dispose pas d'un espace disque suffisant pour le conteneur Docker. Si votre stockage système ne dispose que de 1 à 2 Go d'espace disponible, `sam build` peut échouer pendant le traitement, même si le stockage système n'est pas complètement complet avant le début de la génération. Pour de plus amples informations, veuillez consulter [ce problème GitHub](#).

Erreur : « Docker.Errors.APIError : 500 Erreur de serveur... Vous avez atteint votre limite de taux de traction. »

Docker Hub limite les demandes que les utilisateurs anonymes peuvent effectuer. 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 votre système Docker est authentifié avec vos informations d'identification Docker Hub.

Erreur: « Erreur du serveur 500 : Montage C:\Users \... »

Les utilisateurs Windows peuvent voir cette erreur de montage Docker lors du débogage AWS SAM applications :

```
Fetching lambci/lambdajs10.x Docker container image.....
2019-07-12 13:36:58 Mounting C:\Users\\AppData\Local\Temp\ ... as /var/
task:ro,delegated inside runtime container
Traceback (most recent call last):
...
requests.exceptions.HTTPError: 500 Server Error: Internal Server Error ...
```

Essayez d'actualiser les informations d'identification de vos disques partagés (dans les paramètres Docker).

À l'aide de WSL, webviews (par exemple, « Invoke on AWS » formulaire) sont cassés

Il s'agit d'un problème connu de VS Code pour les utilisateurs de Cisco VPN. Pour de plus amples informations, veuillez consulter [ce problème GitHub](#).

Une solution de contournement est suggérée dans [ce problème de suivi WSL](#).

Débogage d'une application TypeScript, mais les points d'arrêt ne fonctionnent pas

Cela se produit s'il n'existe pas de carte source permettant de lier le fichier JavaScript compilé au fichier TypeScript source. Pour corriger cela, ouvrez votre `tsconfig.json` et assurez-vous que l'option et la valeur suivantes sont définies : `"inlineSourceMap": true`.

Utilisation des documents d'automatisation de Systems Manager Automation

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 visualiser les données

opérationnelles de plusieurs services AWS et d'automatiser les tâches opérationnelles sur l'ensemble de vos ressources AWS.

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 de 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 de Systems Manager Automation de](#)
- [Ouverture d'un document d'automatisation de Systems Manager Automation](#)
- [Modification d'un document d'automatisation Systems Manager Automation](#)
- [Publication d'un document d'automatisation Systems Manager Automation](#)
- [Suppression d'un document Systems Manager Automation de](#)
- [Exécution d'un document Systems Manager Automation de](#)
- [Dépannage des 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 de plus amples informations, consultez le [AWS Systems Manager Guide de l'utilisateur](#).
- Vous connaissez les cas d'utilisation de Systems Manager Automation de Systems Manager. 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 leAWS Identity and Access Management(IAM) nécessaires à la création, à la modification, à la publication et à la suppression de documents d'automatisation de Systems Manager Automation de Le document de politique suivant définit les autorisations IAM nécessaires pouvant être utilisées dans une politique principale :

```
{
  "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, consultez [AWS Informations d'identification IAM](#).

Création d'un nouveau document de Systems Manager Automation de

Vous pouvez créer un nouveau document Automation dansJSONouYAMLà l'aide de Visual Studio Code. Lorsque vous créez un nouveau document Automation, 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 pourAWS.

Pour créer un nouveau document d'automatisation

1. Ouvrez VS Code.
2. Dans la pageAfficher, choisissezPalette de commandespour ouvrir la palette de commandes.
3. Dans la palette de commandes, entrezAWSBoîte à outils Créer un nouveau document Systems Manager Automation en local.
4. Choisissez l'un des modèles de démarrage pour un exemple Hello World.
5. Choisissez JSON ou YAML.

Un nouveau document Automation est créé.

Note

Votre nouveau document Automation dans VS Code n'apparaît pas automatiquement dansAWS. Vous devez le publier dans AWS avant de pouvoir l'exécuter.

Ouverture d'un document d'automatisation de Systems Manager Automation

Vous utilisez le pluginAWSAutomation de Systems Manager Automation de existant de Systems Manager Automation Lorsque vous ouvrez un document d'automatisation existant, il apparaît comme un fichier sans titre dans VS Code.

Pour ouvrir votre document Automation

1. Ouvrez VS Code.
2. Dans le panneau de navigation de gauche, choisissezAWSpour ouvrirAWSexplorateur.
3. DansAWS, pourSystems Manager, choisissez l'icône de téléchargement sur le document que vous souhaitez ouvrir, puis choisissez la version du document. Le fichier s'ouvre au format correspondant à cette version. Sinon, choisissez soitTélécharger au format JSONouTélécharger 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 dans AWS avant de l'exécuter.

Modification d'un document d'automatisation Systems Manager Automation

Si vous possédez des documents Automation, ils apparaissent dans la propriété de ma catégorie de documents Systems Manager dans l'AWS Explorer. Vous pouvez posséder des documents Automation qui existent déjà dans AWS, et vous pouvez être propriétaire de documents nouveaux ou mis à jour que vous avez précédemment publiés sur AWS de VS Code.

Lorsque vous ouvrez un document Automation pour le modifier dans VS Code, vous pouvez en faire plus que dans l'AWS Management Console. Par exemple :

- Il existe une validation de schéma sur les deux JSON et YAML Formats.
- Des extraits sont disponibles dans l'éditeur de documents pour vous permettre de créer n'importe quel type d'étape d'automatisation.
- La saisie semi-automatique est prise en charge pour différentes options dans JSON et YAML.

Utilisation des versions

Les documents d'automatisation de Systems Manager Automation de Automation de Automation de Vous pouvez choisir la version par défaut d'un document Automation dans VS Code.

Pour définir une version par défaut

- Dans l'explorateur AWS, accédez au document sur lequel vous souhaitez définir la version par défaut, ouvrez le menu contextuel (clic droit) du document et choisissez Set default version (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 d'automatisation Systems Manager Automation

Une fois que vous avez modifié votre document Automation dans VS Code, vous pouvez le publier dans AWS.

Pour publier votre document d'automatisation

1. Ouvrez le document Automation que vous souhaitez publier à l'aide de la procédure décrite dans [Ouverture d'un document d'automatisation de Systems Manager Automation](#).
2. Apportez les modifications que vous souhaitez publier. Pour plus d'informations, consultez [Modification d'un document d'automatisation Systems Manager Automation](#).
3. Dans le coin supérieur droit du fichier ouvert, choisissez l'icône de chargement.
4. Dans la boîte de dialogue du flux de travail de publication, choisissez l'AWS Région dans laquelle vous voulez publier le document d'automatisation.
5. Si vous publiez un nouveau document, choisissez Création rapide. Sinon, choisissez Mise à jour rapide pour mettre à jour un document Automation existant dans 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.

Suppression d'un document Systems Manager Automation de

Vous pouvez supprimer des documents Automation 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 Automation 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 Automation

1. Ouvrez VS Code.

2. Dans le panneau de navigation de gauche, choisissez AWS pour ouvrir AWS Explorateur.
3. Dans AWS, pour Systems Manager, ouvrez le menu contextuel (clic droit) du document que vous souhaitez supprimer et choisissez Suppression du document.

Exécution d'un document Systems Manager Automation de

Une fois que votre document Automation est publié sur AWS, vous pouvez l'exécuter pour exécuter des tâches en votre nom dans votre AWS. Pour exécuter votre document d'automatisation, vous pouvez utiliser la AWS Management Console, les API de Systems Manager, la AWS CLI ou le AWS Tools for PowerShell. Pour plus d'informations sur la façon d'exécuter un document d'automatisation, consultez [Exécution d'une automatisation simple](#) dans le AWS Systems Manager Guide de l'utilisateur.

Sinon, si vous souhaitez utiliser l'un des kits SDK AWS avec les API de Systems Manager pour exécuter votre document d'automatisation, consultez les [références du kit SDK AWS](#).

Note

L'exécution d'un document d'automatisation peut créer de nouvelles ressources dans AWS et peut entraîner des frais de facturation. Nous vous recommandons fortement de comprendre ce que votre document d'automatisation va créer dans votre compte avant de le démarrer.

Dépannage des documents d'automatisation de Systems Manager dans Toolkit for VS Code

J'ai enregistré mon document Automation dans VS Code, mais je ne le vois pas dans le plugin AWS Management Console.

L'enregistrement d'un document Automation dans VS Code ne publie pas le document Automation dans AWS. Pour plus d'informations sur la publication de votre document d'automatisation, consultez [Publication d'un document d'automatisation Systems Manager Automation](#).

La publication de mon document Automation a échoué avec une erreur d'autorisation.

Assurez-vous que votre profil d'informations d'identification AWS 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 Automation sur AWS, mais je ne le vois pas dans la AWS Management Console.

Assurez-vous que vous avez publié le document dans la même région AWS que vous parcourez dans la AWS Management Console.

J'ai supprimé mon document Automation, mais je reçois toujours des factures pour les ressources qu'il a créées.

La suppression d'un document Automation ne supprime pas les ressources qu'il a créées ou modifiées. Vous pouvez identifier les ressources AWS que vous avez créées à partir de la [Console de gestion de la facturation AWS](#), examiner vos frais et choisir les ressources à supprimer à partir de là.

Travailler avec AWS Step Functions

Le AWS Toolkit for Visual Studio Code (VS Code) fournit un support pour [AWS Step Functions](#). À l'aide du Toolkit for VS Code, vous pouvez créer, mettre à jour et exécuter des machines d'état Step Functions.

Rubriques

- [Travailler avec AWS Step Functions](#)

Travailler avec AWS Step Functions

Vous pouvez utiliser le AWS Toolkit for Visual Studio Code (VS Code) pour effectuer diverses opérations avec des [machines à états](#).

Rubriques

- [Prérequis](#)
- [Travaillez avec des machines d'état dans VS Code](#)
- [Modèles de machines à états](#)
- [Visualisation de graphes de machines à états](#)
- [Extraits de code](#)
- [Complétion et validation du code](#)

Prérequis

- Assurez-vous que votre système répond aux conditions requises spécifiées dans [Installation du Toolkit for VS Code](#), puis installez le kit d'outils.
- Assurez-vous d'avoir configuré vos informations d'identification avant d'ouvrir l'AWS explorateur.

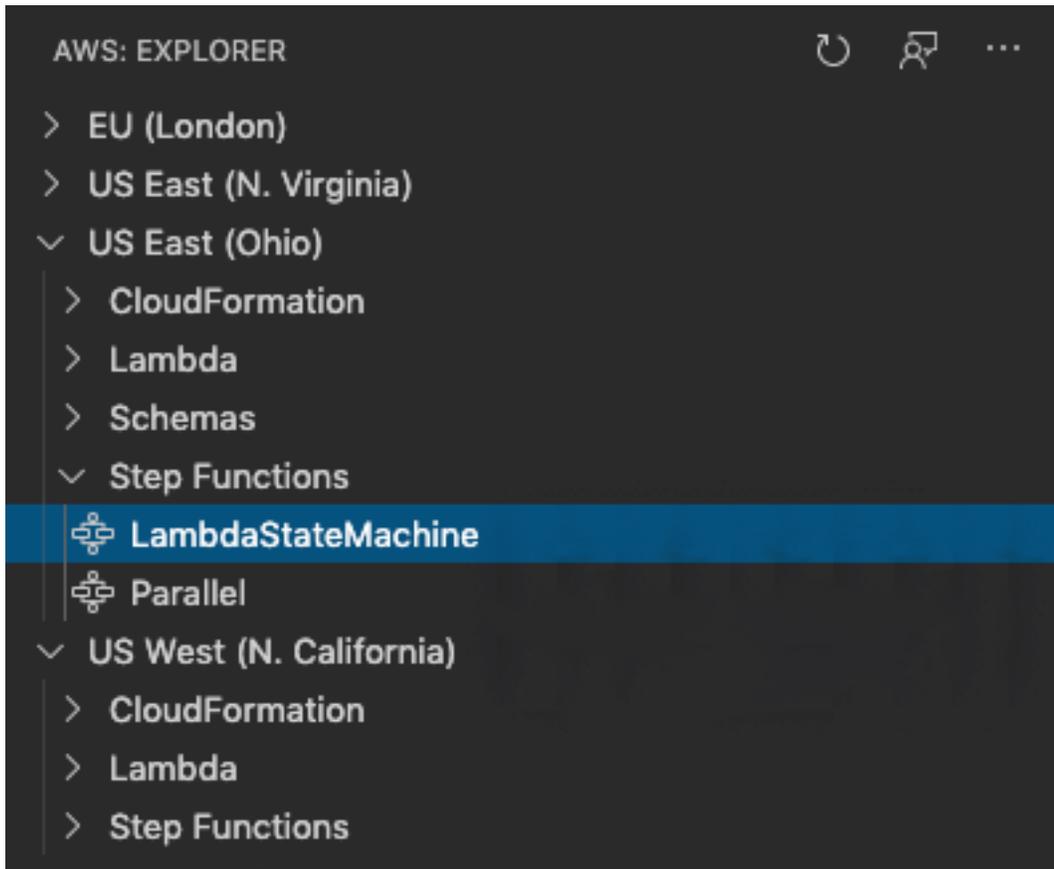
Travaillez avec des machines d'état dans VS Code

Vous pouvez utiliser VS Code pour interagir avec des machines d'état distantes et développer des machines d'état localement au format JSON ou YAML. Vous pouvez créer ou mettre à jour des machines d'état, répertorier les machines d'état existantes, les exécuter et les télécharger. VS Code vous permet également de créer de nouvelles machines d'état à partir de modèles, de voir une visualisation de votre machine d'état et d'accéder à des extraits de code, mais aussi de saisir et de valider du code.

Répertorier les machines d'état existantes

Si vous avez déjà créé des machines d'état, vous pouvez en afficher la liste :

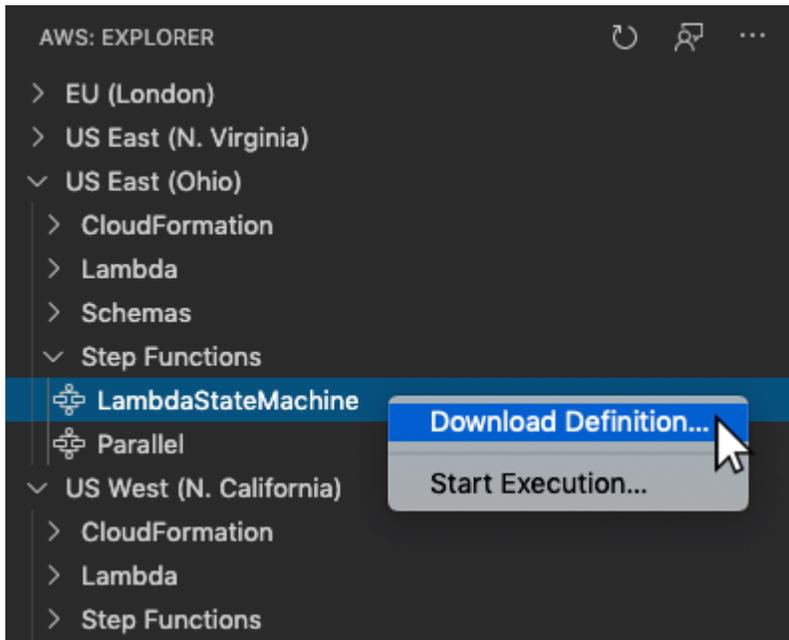
1. Ouvrez AWS Explorer.
2. Sélectionnez Step Functions
3. Vérifiez que toutes les machines d'état de votre compte sont répertoriées.



Télécharger une machine à états

Pour télécharger une machine d'état :

1. Dans l'AWS explorateur, cliquez avec le bouton droit sur la machine à états que vous souhaitez télécharger.
2. Sélectionnez Download (Télécharger), puis sélectionnez l'emplacement où vous souhaitez télécharger la machine d'état.
3. Vérifiez que le téléchargement s'est déroulé correctement.



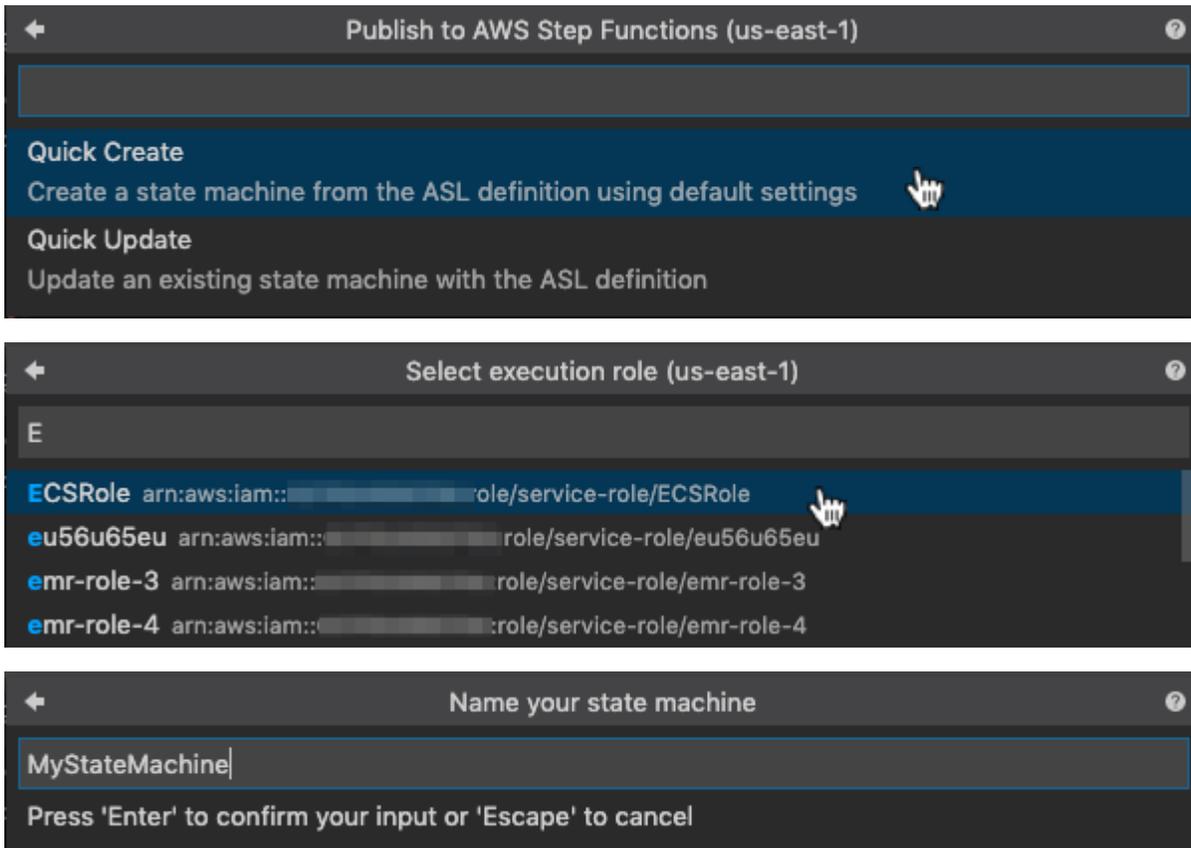
Créer une machine d'état

Vous pouvez créer vous-même une nouvelle machine d'état ou utiliser un modèle. Pour plus d'informations sur la création d'une machine d'état à partir d'un modèle, veuillez consulter [Modèles de machine d'état](#). Pour créer une nouvelle machine d'état :

1. Créez un nouveau fichier ASL ([Amazon States Language](#)) avec votre définition de machine d'état. Utilisez le menu en bas à droite pour le définir en tant que fichier Amazon States Language.
2. Sélectionnez Publish to Step Functions (Publier dans Step Functions).

```
1  Publish to Step Functions | Render graph
2  {
3    "StartAt": "FirstState",
4    "States": {
5      "FirstState": {
6        "Type": "Task",
7        "Resource": "arn:aws:lambda:us-east-1:123456789012:func:Function",
8        "Next": "ChoiceState"
9      },
10     "ChoiceState": {
11       "Type": "Choice",
```

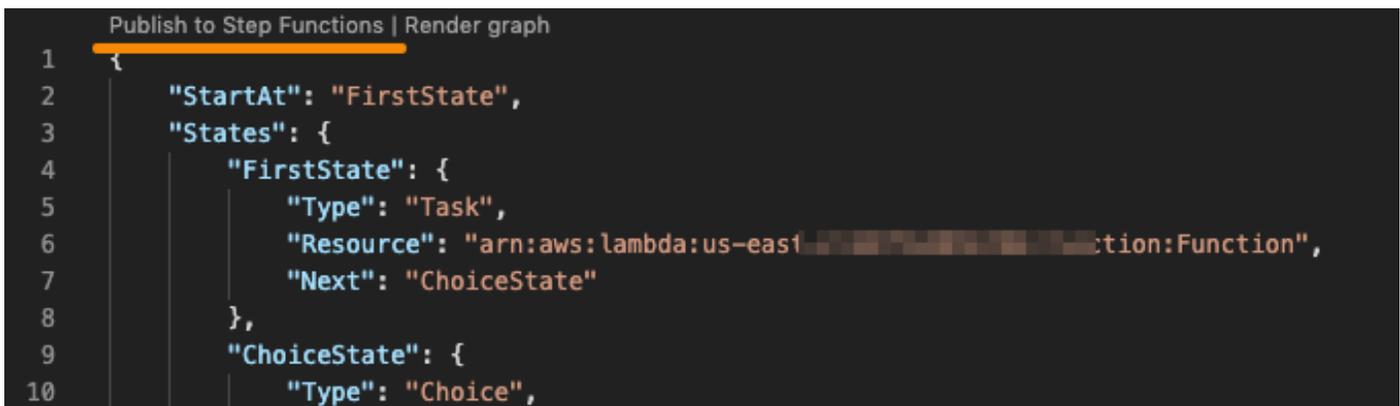
3. Sélectionnez Quick Create (Création rapide), choisissez un rôle et nommez votre machine d'état.



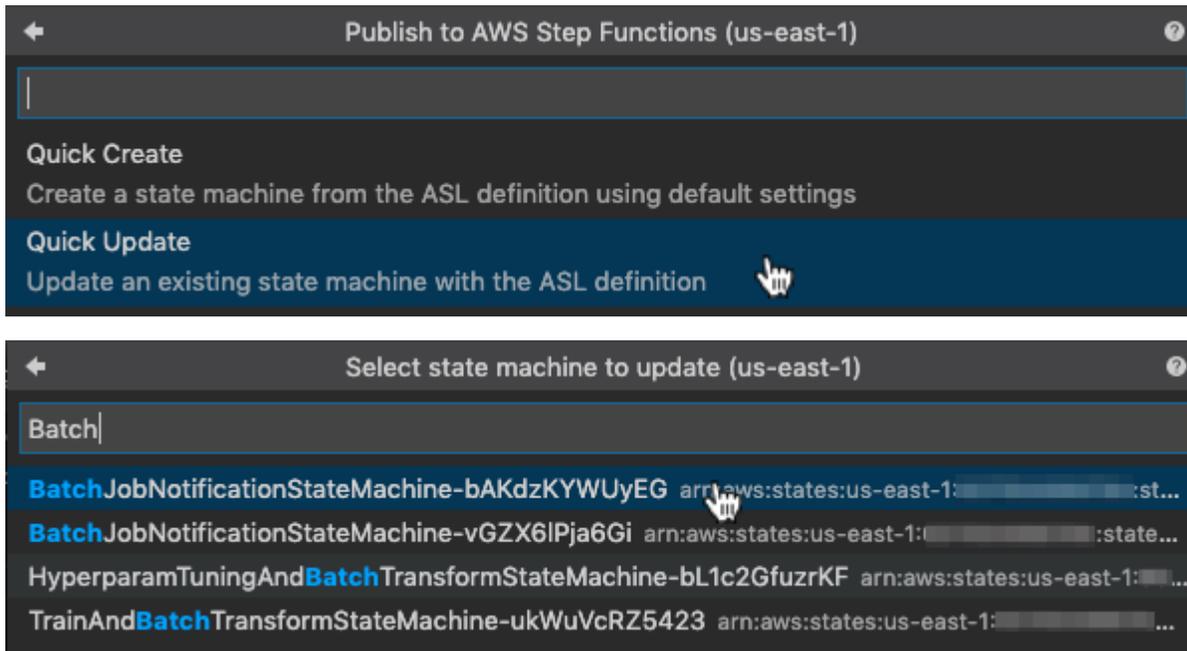
Mettre à jour une machine d'état

Pour mettre à jour une machine d'état :

1. Modifiez le fichier ASL avec votre définition de machine d'état.
2. Sélectionnez Publish to Step Functions Publier dans Step Functions).



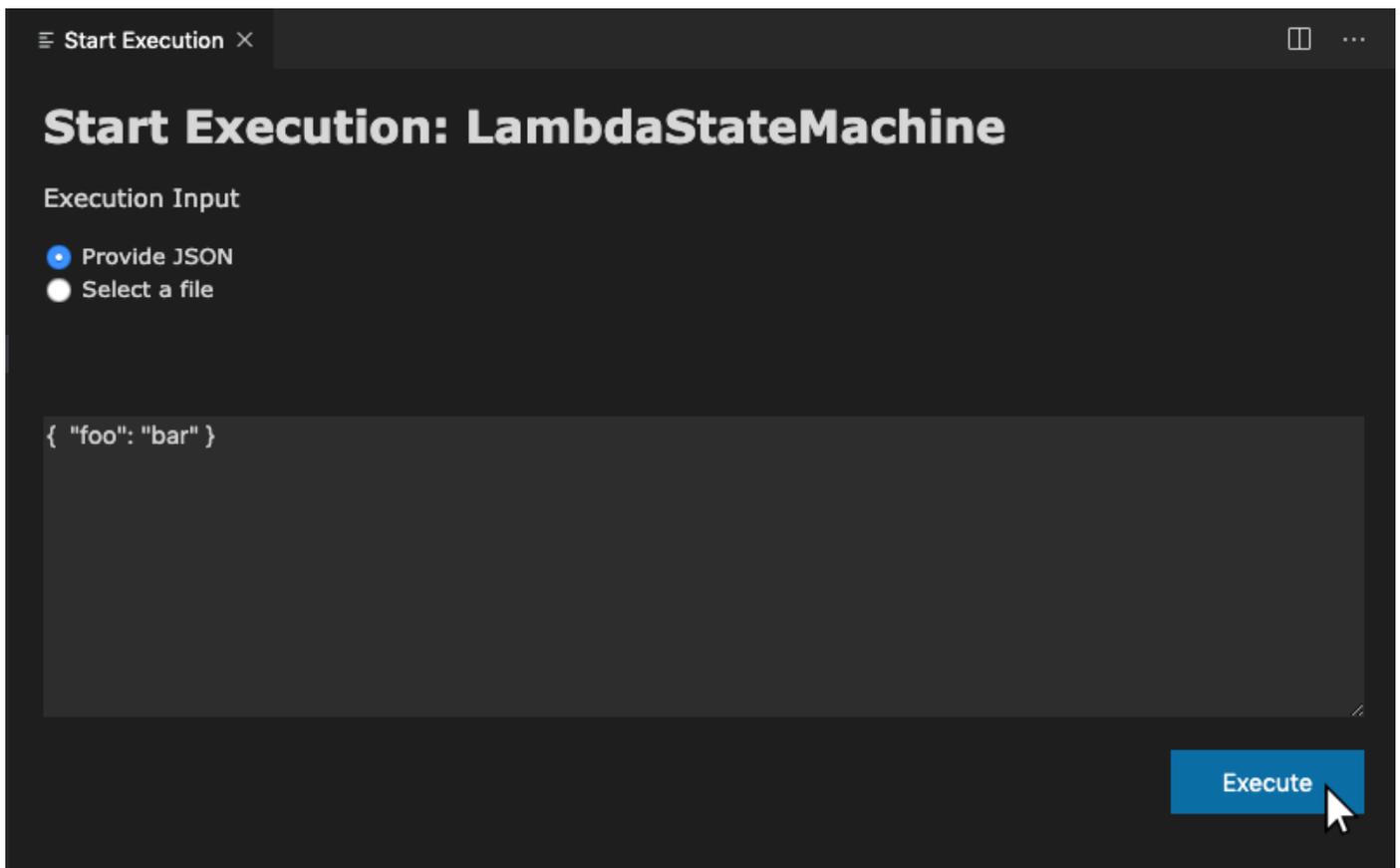
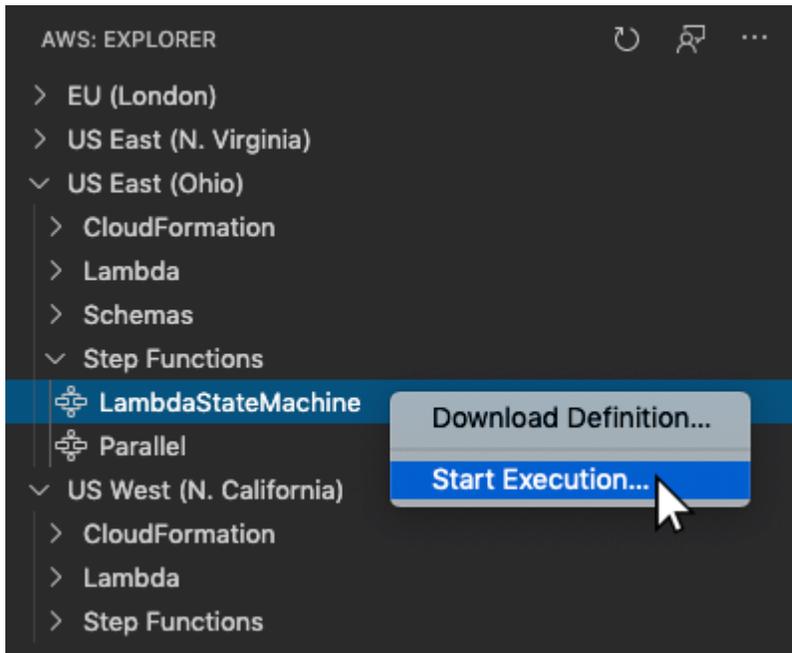
3. Sélectionnez Quick Update (Mise à jour rapide), puis sélectionnez la machine d'état à mettre à jour.



Exécuter une machine à états

Pour exécuter une machine à états :

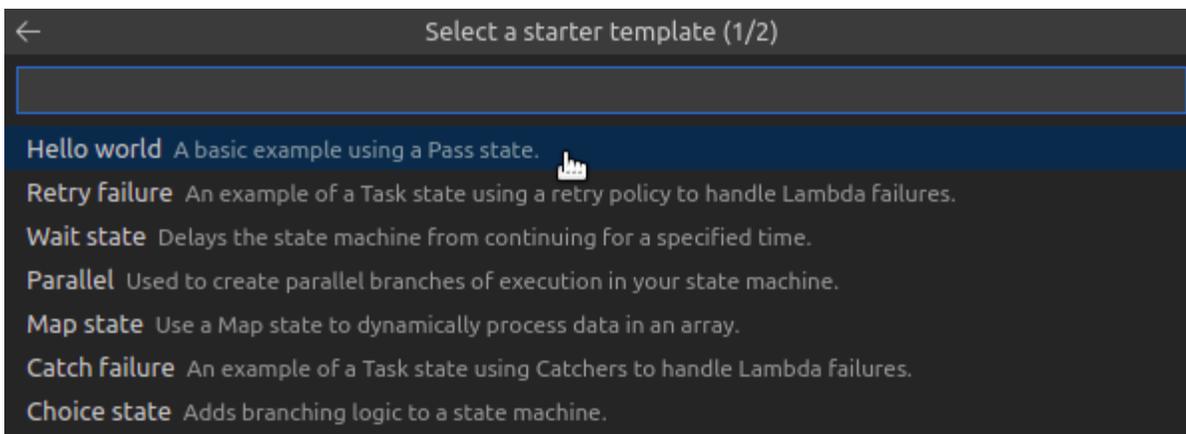
1. Dans l'AWS explorateur, cliquez avec le bouton droit sur la machine à états que vous souhaitez exécuter.
2. Fournissez une entrée pour la machine à états. Vous pouvez essayer à la fois l'entrée à partir d'un fichier et l'entrée dans une zone de texte.
3. Démarrez la machine d'état et vérifiez qu'elle fonctionne correctement.



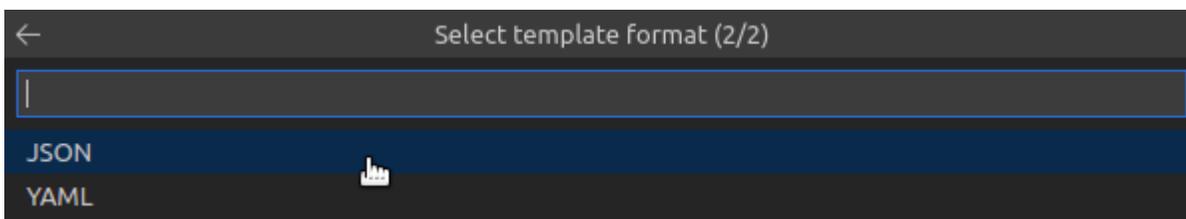
Modèles de machines à états

Lorsque vous créez une machine d'état, vous avez la possibilité de la créer à partir d'un modèle. Le modèle contient un exemple de définition de machine d'état avec plusieurs états couramment utilisés. Cela vous fournit un point de départ. Pour utiliser des modèles de machine d'état :

1. Ouvrez la palette de commandes dans VS Code.
2. Sélectionnez AWS Toolkit Create a new Step Functions state machine.
3. Choisissez le modèle que vous souhaitez utiliser.



4. Choisissez si vous souhaitez utiliser le format de modèle JSON ou YAML.



Visualisation de graphes de machines à états

Les visualisations graphiques vous permettent de voir à quoi ressemble votre machine d'état sous la forme d'un graphique. Lorsque vous créez une visualisation graphique, un autre onglet s'ouvre et affiche une visualisation de la machine à états JSON ou YAML. Vous pouvez ensuite comparer la définition de la machine d'état que vous écrivez avec sa visualisation. Lorsque vous modifiez la définition de votre machine d'état, la visualisation est mise à jour.

Note

Pour créer une visualisation d'une définition de machine d'état, la définition doit être ouverte dans l'éditeur actif. Si vous fermez ou renommez le fichier de définition, la visualisation se ferme.

Pour créer une visualisation graphique d'une machine d'état :

1. Définissez votre machine d'état.
2. Ouvrez la palette de commandes dans VS Code.
3. Pour créer une visualisation, utilisez le bouton de visualisation dans le coin supérieur droit ou choisissez **AWS Rendre le graphe**.

The screenshot displays the VS Code interface with two panels. The left panel shows the ASL JSON file 'Choice-example.asl.json' with the following content:

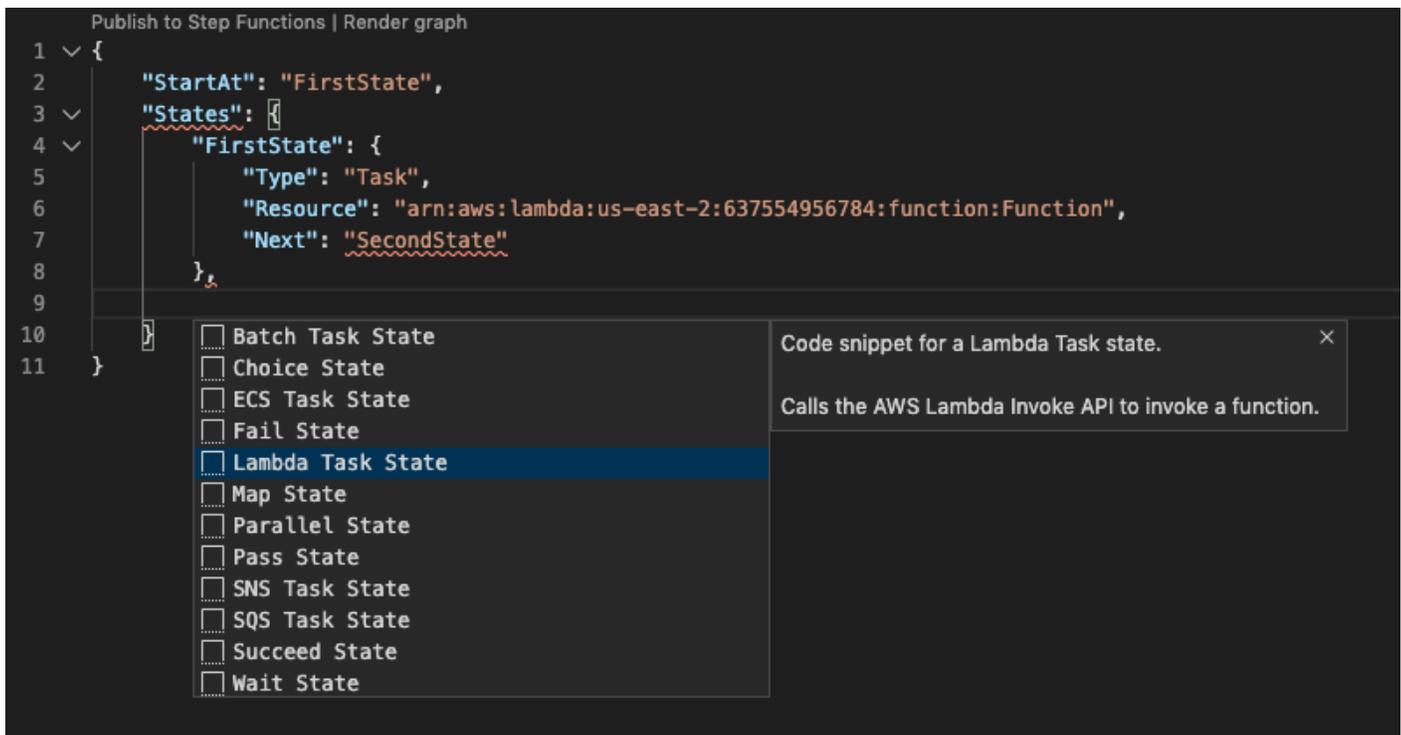
```
1  {
2    "Comment": "An example of the Amazon States Language using a choice
3    state.",
4    "StartAt": "FirstState",
5    "States": {
6      "FirstState": {
7        "Type": "Task",
8        "Resource":
9        "arn:aws:lambda:us-east-2:637554956784:function:Function",
10       "Next": "ChoiceState"
11     },
12     "ChoiceState": {
13       "Type": "Choice",
14       "Choices": [
15         {
16           "Variable": "$.foo",
17           "NumericEquals": 1,
18           "Next": "FirstMatchState"
19         },
20         {
21           "Variable": "$.foo",
22           "NumericEquals": 2,
23           "Next": "SecondMatchState"
24         }
25       ],
26       "Default": "DefaultState"
27     },
28     "FirstMatchState": {
29       "Type": "Task",
30       "Resource":
31       "arn:aws:lambda:us-east-2:637554956784:function:Function",
32       "Next": "NextState"
33     },
34     "SecondMatchState": {
35       "Type": "Task",
36       "Resource":
37       "arn:aws:lambda:us-east-2:637554956784:function:Function",
38       "Next": "NextState"
39     },
40     "NextState": {
41       "Type": "Task",
42       "Resource":
43       "arn:aws:lambda:us-east-2:637554956784:function:Function",
44       "Next": "End"
45     },
46     "DefaultState": {
47       "Type": "Task",
48       "Resource":
49       "arn:aws:lambda:us-east-2:637554956784:function:Function",
50       "Next": "End"
51     },
52     "End": {}
53   }
54 }
```

The right panel shows the rendered state machine graph. The flow starts at 'Start', goes to 'FirstState', then to 'ChoiceState'. From 'ChoiceState', the flow branches into 'FirstMatchState' and 'SecondMatchState'. Both 'FirstMatchState' and 'SecondMatchState' lead to 'NextState'. 'NextState' leads to 'End'. 'DefaultState' also leads to 'End'.

Extraits de code

Les extraits de code vous permettent d'insérer de courtes sections de code. Pour utiliser des extraits de code :

1. Ouvrez un fichier et enregistrez-le avec l'extension `.asl.json` au format JSON ou `.asl.yaml` au format YAML.
2. Créez une nouvelle machine d'état avec la propriété States (États).
3. Placez le curseur dans States (États).
4. Utilisez la combinaison de touches `Control + Space` et sélectionnez l'extrait de code de votre choix.
5. Utilisez `Tab` pour parcourir la variable et les paramètres dans l'extrait de code.
6. Testez les extraits `Retry` (Réessayer) et `Catch` (Attraper) en plaçant le curseur dans l'état associé.



Complétion et validation du code

Pour voir comment fonctionne la saisie de code :

1. Créez plusieurs états.
2. Placez le curseur après une propriété `Next` ou `Default`. `StartAt`
3. Utilisez la combinaison de touches `Control + Space` pour répertorier les saisies disponibles. Utilisez à nouveau `Control + Space` pour accéder à des propriétés supplémentaires, basées sur le `Type` de `State`.
4. Au fur et à mesure que vous travaillez, la validation de code se produit pour les éléments suivants :

- Propriétés manquantes
- Valeurs incorrectes
- Aucun état terminal
- États pointés non existants

```

"FirstMatchState": {
  "Type": "Task",
  "Resource": "arn:aws:lambda:us-east-2:637554956784:function:Function",
  "Next": ""
},
"SecondMatchState": {
  "Type": "ChoiceState",
  "Resource": "arn:aws:lambda:us-east-2:637554956784:function:Function",
  "Next": "SecondMatchState"
},
"DefaultState": {
  "Type": "Fail",
  "Error": "DefaultStateError",
  "Cause": "No Matches!"
},
"NextState": {
  "Type": "Task",
  "Resource": "arn:aws:lambda:us-east-2:637554956784:function:Function",
  "End": true
}

```

```

"FirstMatchState": {
  "Type": "Task",
  "Resource": "arn:aws:lambda:us-east-2:637554956784:function:Function",
  "Catch": [
    {
      "Error": "DefaultStateError",
      "Cause": "No Matches!"
    }
  ]
},

```

An array of objects, called Catchers, that define a fallback state. This state is executed if the state encounters runtime errors and its retry policy is exhausted or isn't defined.

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 services AWS qui sont listés par défaut dans AWS Explorer, vous pouvez également accéder à Ressources (Resources) et choisir parmi des centaines de ressources à ajouter à l'interface. Dans AWS, une ressource est une entité que vous pouvez utiliser. Parmi les ressources qui peuvent être ajoutées, citons Amazon AppFlow, Amazon Kinesis Data Streams, AWS Rôles IAM, Amazon VPC et Amazon CloudFront Distributions.

Après avoir fait votre sélection, vous pouvez vous rendre sur 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 Ressources (Resources), vous pouvez interagir avec lui et ses ressources de la manière suivante :

- Afficher une liste des ressources existantes qui sont disponibles dans la Région AWS pour ce type de ressource.
- Afficher une version en lecture seule du fichier JSON qui décrit une ressource.
- Copier l'identifiant de la ressource.
- Afficher la documentation AWS 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 modèle au format JSON conforme à un schéma.*
- Mettez à jour ou supprimez une ressource existante.*

Important

*Dans la version actuelle du AWS Toolkit for Visual Studio Code l'option pour créer, modifier et supprimer des ressources est une fonction 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 de la AWS Toolkit for Visual Studio Code sans préavis.

Pour autoriser l'utilisation de fonctionnalités expérimentales pour les ressources, ouvrez le Paramètres dans votre IDE VS Code, puis développez Extension et choisissez AWSToolkit. Under AWSToolkit, sélectionnez jsonResourceModification pour vous permettre de créer, mettre à jour et supprimer des ressources.

Pour plus d'informations, consultez [Utilisation des fonctions expérimentales](#).

Autorisations IAM pour l'accès aux ressources

Vous avez besoin d'autorisations spécifiques d'AWS Identity and Access Management pour accéder aux ressources associées aux services AWS. Par exemple, une entité IAM, telle qu'un utilisateur ou un rôle, a besoin des autorisations `Lambda::Function` pour accéder aux ressources AWS Toolkit.

En plus des autorisations pour les ressources de service, une entité IAM a besoin d'autorisations pour permettre à la boîte à outils Toolkit d'appeler des opérations de l'API Cloud Control pour son compte. 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.

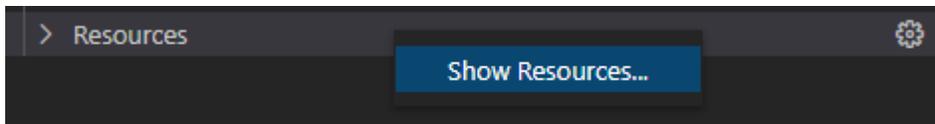
Le moyen le plus simple d'accorder des autorisations est de joindre la [politique gérée PowerUserAccess](#) à l'entité IAM qui appelle ces opérations d'API à l'aide de l'interface Toolkit. Cette [politique gérée](#) accorde une série d'autorisations pour effectuer des tâches de développement d'applications, 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](#).

Ajouter des ressources existantes et interagir avec celles-ci

1. Dans **AWSExplorateur**, clic droit **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 ressource à ajouter à la liste **AWSExplorateur** et appuyez sur **Retour** ou choisissez **OK** pour confirmer.

Les types de ressource que vous avez sélectionnés sont répertoriés sous **Ressources**.

Note

Si vous avez déjà ajouté un type de ressource au **AWSExplorateur** puis décochez la case correspondant à ce type, il n'est plus répertorié sous **Ressources** après avoir choisi **OK**. Seuls les types de ressources actuellement sélectionnés sont visibles dans le **AWSExplorateur**.

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 ressource : Copiez l'identifiant de la ressource spécifique dans le presse-papier. (Par exemple, la ressource `AWS Toolkit:DynamoDB::Table` peut être identifiée à l'aide de la propriété `TableName`.)
- Version préliminaire : Afficher une version en lecture seule du modèle au format JSON qui décrit la ressource.

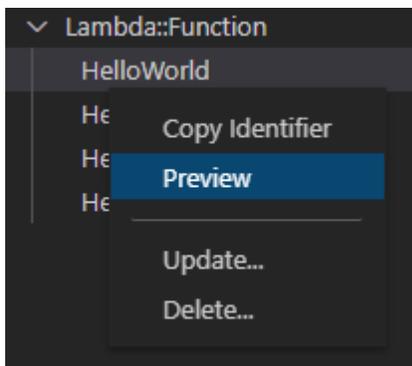
Une fois le modèle de ressources affiché, vous pouvez le modifier en choisissant l'option **Mise à jour** à droite de l'onglet de l'éditeur. Pour mettre à jour une ressource, vous devez disposer des éléments requis **???** activé.

- Mise à jour : Modifiez le modèle au format JSON pour la ressource dans un éditeur VS Code. Pour plus d'informations, consultez [Créer et modifier des ressources](#).

- Supprimer : Supprimez la ressource en confirmant la suppression dans une boîte de dialogue qui s'affiche. (La suppression de ressources est actuellement une [??? dans cette version de AWS Toolkit for Visual Studio Code.](#))

Warning

Si vous supprimez une ressource, n'importe quel type AWS CloudFormation La 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 AWS CloudFormation modèle. Pour plus d'informations, veuillez consulter ce [du centre de connaissances](#).



Créer et modifier des ressources

Important

La création et la mise à jour des ressources sont actuellement [??? dans cette version de AWS Toolkit for Visual Studio Code.](#)

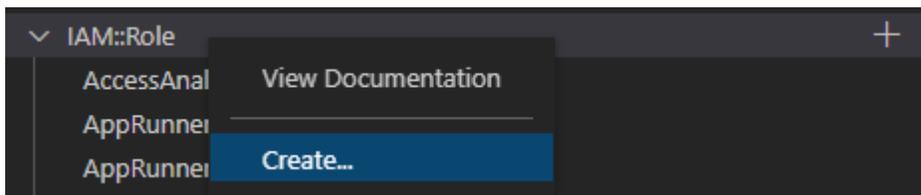
La création d'une nouvelle ressource implique l'ajout d'un type de ressource au `Ressources` puis en éditant un modèle au format JSON qui définit la ressource, ses propriétés et ses attributs.

Par exemple, une ressource qui appartient au `AWS Toolkit:SageMaker::UserProfile` le type de ressource est défini à l'aide d'un modèle qui crée un profil utilisateur pour Amazon SageMaker Studio. Le modèle qui définit cette ressource de profil utilisateur doit être conforme au schéma de type de ressource pour `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.

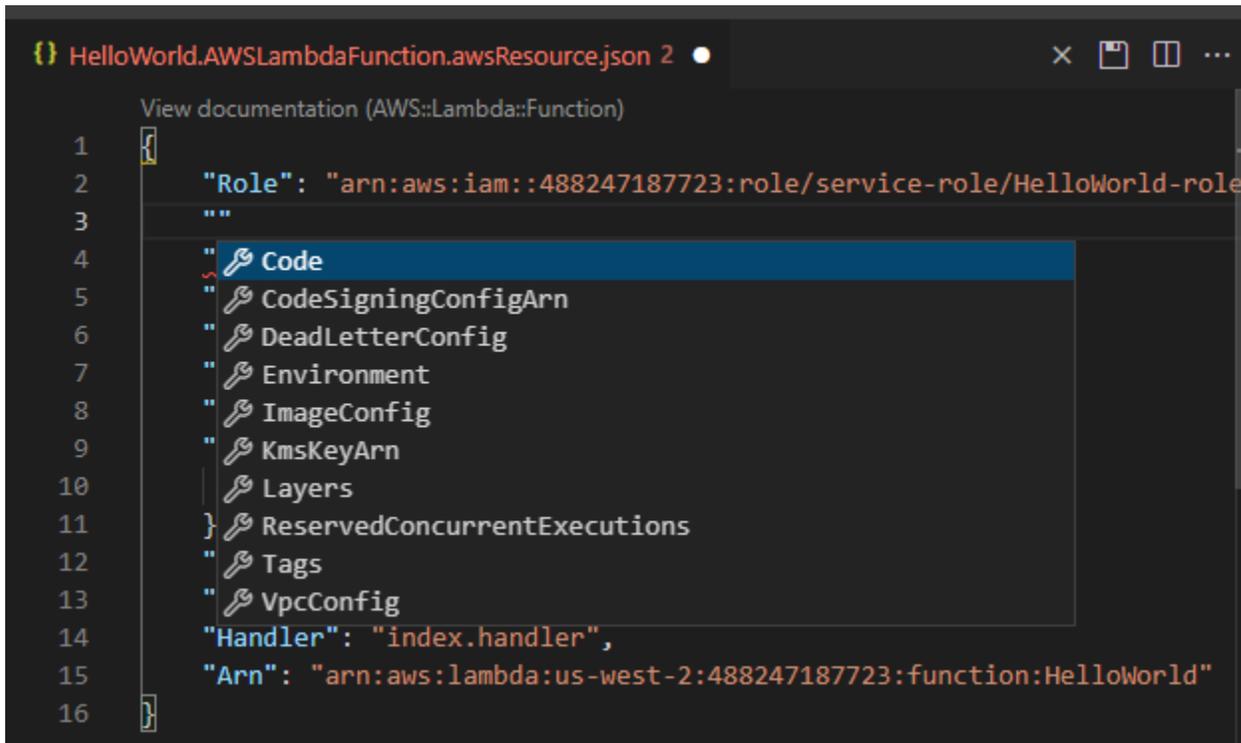
1. Ajoutez le type de ressource pour la ressource que vous souhaitez créer en cliquant avec le bouton droit sur `Ressources` et en choisissant `Afficher les ressources`.
2. Une fois le type de ressource ajouté sous `Ressources`, choisissez l'icône plus (« + ») pour ouvrir le fichier modèle dans un nouvel éditeur.

Vous pouvez également cliquer avec le bouton droit sur le nom du type de ressource et choisir `Créer`. Vous pouvez également accéder aux informations sur la façon de modéliser la ressource en choisissant `Voir la documentation`.



3. Dans l'éditeur, commencez à définir les propriétés qui constituent le modèle de ressource. La fonction de saisie semi-automatique suggère des noms de propriétés conformes au schéma de votre modèle. Lorsque vous survolez un type de propriété, un volet affiche une description de son utilisation. Pour plus d'informations sur le schéma, choisissez `Voir la documentation`.

Tout texte qui n'est pas conforme au schéma de ressources est indiqué par un soulignement rouge ondulé.



```
1 {
2   "Role": "arn:aws:iam::488247187723:role/service-role/HelloWorld-role",
3   ""
4   "Code"
5   "CodeSigningConfigArn"
6   "DeadLetterConfig"
7   "Environment"
8   "ImageConfig"
9   "KmsKeyArn"
10  "Layers"
11  "ReservedConcurrentExecutions"
12  "Tags"
13  "VpcConfig"
14  "Handler": "index.handler",
15  "Arn": "arn:aws:lambda:us-west-2:488247187723:function:HelloWorld"
16 }
```

4. Une fois que vous avez fini de déclarer votre ressource, choisissez l'option **Enregistrer** pour valider votre modèle et enregistrer la ressource sur la télécommande **AWS Cloud**.

Si votre modèle définit la ressource conformément à son schéma, un message s'affiche pour confirmer que la ressource a été créée. (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.

Sécurité de AWS Toolkit for Visual Studio Code

Rubriques

- [Protection des données dans AWS Toolkit for Visual Studio Code](#)

Protection des données dans AWS Toolkit for Visual Studio Code

Le [modèle de responsabilité AWS partagée](#) s'applique à la protection des données dans AWS Toolkit for Visual Studio Code. Comme décrit dans ce modèle, AWS est responsable de la protection de l'infrastructure globale sur laquelle l'ensemble d'AWS Cloud s'exécute. 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 pour les Services AWS que vous utilisez. Pour en savoir plus sur la confidentialité des données, consultez [Questions fréquentes \(FAQ\) sur la confidentialité des données](#). Pour en savoir plus sur la protection des données en Europe, consultez le billet de blog Modèle de responsabilité partagée [AWS et RGPD \(Règlement général sur la protection des données\)](#) sur le AWSBlog de sécurité.

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

- Utilisez l'authentification multifactorielle (MFA) avec chaque compte.
- Utilisez les certificats SSL/TLS pour communiquer avec les ressources AWS. Nous exigeons TLS 1.2 et recommandons TLS 1.3.
- Configurez une API (Interface de programmation) et le journal de l'activité des utilisateurs avec AWS CloudTrail.
- Utilisez des solutions de chiffrement AWS, ainsi que tous les contrôles de sécurité par défaut au sein des 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 FIPS (Federal Information Processing Standard) 140-2 lorsque vous accédez à AWS via une CLI (Interface de ligne de commande) ou une API (Interface de programmation), utilisez un point de terminaison FIPS (Federal Information

Processing Standard). Pour en savoir plus sur les points de terminaison FIPS (Federal Information Processing Standard) disponibles, consultez [Federal Information Processing Standard \(FIPS\) 140-2](#) (Normes de traitement de l'information fédérale).

Nous vous recommandons fortement de ne jamais placer d'informations confidentielles ou sensibles, telles que les adresses e-mail de vos clients, dans des balises ou des champs de texte libre tels que le champ Name (Nom). Cela inclut lorsque vous travaillez avec AWS Toolkit for Visual Studio Code ou autre outil Services AWS à l'aide de la console, de l'API ou AWS des SDK. AWS CLI Toutes les données que vous saisissez 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 Guide de l'utilisateur AWS Toolkit for Visual Studio Code

Dernière mise à jour majeure de la documentation : 14 octobre 2021

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
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 services AWS.	14 octobre 2021

<u>Support pour AWS les ressources</u>	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
<u>Présentation du service Amazon ECR pour AWS Toolkit for Visual Studio Code</u>	Ajout d'une présentation et d'une présentation des fonctionnalités du service Amazon ECR accessibles dans VS Code	14 octobre 2021
<u>Support pour les environnements ARM64</u>	Vous pouvez désormais exécuter des applications sans serveur dans des environnements émulés basés sur ARM64 ainsi que dans des environnements basés sur x86_64.	1er octobre 2021
<u>Application sans serveur AWS</u>	Ajout du support pour l'exécution d'AWS SAM applications sur la plate-forme ARM64	30 septembre 2021
<u>Mise à jour du format de la section Node.js</u>	Selon les commentaires des clients, mise à jour du formatage pour Node.js/TypeScript.	12 août 2021
<u>Prise en charge d'App Runner</u>	Ajout de la prise en charge de AWS App Runner pour AWS Toolkit for Visual Studio Code.	11 août 2021
<u>Fonctions de Debugging Go</u>	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 Prise en charge de	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 du support pour Amazon S3.	30 juillet 2020
AWS Step Functions Prise en charge de	Ajout de la prise en charge de AWS Step Functions.	31 mars 2020
Contenu relatif à la sécurité	Ajout du contenu de sécurité.	6 février 2020
Utilisation d'Amazon EventBridge Schemas	Ajout du support pour 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.