



Panduan Pengguna

# AWS Toolkit for VS Code



# AWS Toolkit for VS Code: Panduan Pengguna

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

Merek dagang dan tampilan dagang Amazon tidak boleh digunakan sehubungan dengan produk atau layanan apa pun yang bukan milik Amazon, dengan cara apa pun yang dapat menyebabkan kebingungan di antara pelanggan, atau dengan cara apa pun yang menghina atau mendiskreditkan Amazon. Semua merek dagang lain yang tidak dimiliki oleh Amazon merupakan kekayaan masing-masing pemiliknya, yang mungkin atau mungkin tidak berafiliasi, terkait dengan, atau disponsori oleh Amazon.

---

# Table of Contents

AWS Toolkit for Visual Studio Code .....	1
Apa itu AWS Toolkit for Visual Studio Code .....	1
Informasi terkait .....	1
Pengembang Amazon Q dan Amazon CodeWhisperer .....	2
Unduh Toolkit .....	3
Mengunduh Toolkit dari VS Code Marketplace .....	3
Toolkit IDE tambahan dari AWS .....	3
Memulai .....	4
Menginstal Toolkit for VS Code .....	4
Prasyarat .....	4
Mengunduh dan menginstal AWS Toolkit for Visual Studio Code .....	4
Prasyarat opsional .....	5
Menghubungkan ke AWS .....	6
Prasyarat .....	6
Membuka panel Masuk .....	7
Menghubungkan ke AWS dari Toolkit .....	7
Otentikasi untuk Amazon CodeCatalyst .....	8
Mengubah AWS Wilayah .....	9
Menambahkan Wilayah ke AWS Explorer .....	9
Sembunyikan Wilayah dari AWS Explorer .....	10
Mengkonfigurasi toolchain Anda .....	10
Mengkonfigurasi toolchain untuk .NET .....	10
Mengkonfigurasi toolchain for Node.js .....	10
Mengkonfigurasi toolchain untuk Python .....	11
Mengkonfigurasi toolchain untuk Java .....	11
Mengkonfigurasi toolchain untuk Go .....	12
Menggunakan Toolchain Anda .....	12
Otentikasi dan akses .....	13
Pusat Identitas IAM .....	13
IAMkredensialnya .....	13
Membuat IAM pengguna .....	14
Membuat file kredensial bersama dari AWS Toolkit for Visual Studio Code .....	15
Tambahkan profil kredensi tambahan .....	16
AWS ID Pembangun .....	17

Menggunakan proses proses kredensi kredensi .....	17
Bekerja dengan AWS .....	18
Fitur eksperimental .....	19
AWS Penjelajah .....	19
Amazon CodeCatalyst .....	20
Apa itu Amazon CodeCatalyst? .....	20
Memulai dengan CodeCatalyst .....	21
Bekerja dengan CodeCatalyst sumber daya .....	21
Bekerja dengan Lingkungan Dev .....	25
Pemecahan Masalah .....	28
Amazon API Gateway .....	29
AWS App Runner .....	30
Prasyarat .....	30
Harga .....	34
Membuat layanan App Runner .....	34
Mengelola layanan App Runner .....	37
AWSKomposer Aplikasi .....	39
Bekerja dengan Komposer AWS Aplikasi .....	40
AWS CDK .....	41
AWS CDK aplikasi .....	42
Tumpukan AWS CloudFormation .....	44
MenghapusAWS CloudFormationtumpukan .....	44
Buat CloudFormation template .....	45
Amazon CloudWatch Logs .....	46
Melihat grup log CloudWatch dan aliran log .....	47
Bekerja dengan CloudWatch peristiwa Log .....	48
Mencari grup log .....	50
Amazon ECR .....	52
Prasyarat .....	53
Menggunakan Amazon ECR dengan Toolkit for VS Code .....	55
Amazon ECS .....	64
Menggunakan IntelliSense untuk file task-definition .....	64
Amazon ECS Exec .....	65
Amazon EventBridge .....	68
Bekerja dengan Amazon EventBridge Schemas .....	68
AWS IAMPenganalisis Akses .....	70

Bekerja dengan AWS IAM Access Analyzer .....	71
AWS IoT .....	75
Prasyarat AWS IoT .....	75
AWS IoT Hal .....	76
Sertifikat AWS IoT .....	77
Kebijakan AWS IoT .....	80
AWS Lambda Fungsi .....	83
Berinteraksi dengan Fungsi Lambda Jarak Jauh .....	84
Amazon Redshift .....	90
Bekerja dengan Amazon Redshift .....	90
Amazon S3 .....	95
Bekerja dengan sumber daya S3 .....	95
Bekerja dengan objek S3 .....	97
AWS Serverless Application .....	101
Memulai .....	101
Menjalankan dan men-debug fungsi Lambda sambil AWS SAM mengecualikan sumber daya template .....	109
Menjalankan dan men-debug sumber daya Amazon API Gateway lokal .....	113
Opsi konfigurasi untuk men-debug aplikasi nirserver .....	117
Pemecahan Masalah .....	124
AWS Systems Manager .....	126
Asumsi dan prasyarat .....	127
Izin IAM untuk dokumen otomatisasi Systems Manager .....	127
Membuat dokumen otomatisasi Systems Manager .....	128
Membuka dokumen otomatisasi Systems Manager .....	129
Mengedit dokumen otomatisasi Systems Manager .....	129
Menerbitkan dokumen otomatisasi Systems Manager .....	130
Menghapus dokumen otomatisasi Systems Manager .....	131
Melaksanakan dokumen otomatisasi Systems Manager .....	131
Pemecahan Masalah .....	132
AWS Step Functions .....	132
AWS Step Functions dan Kode VS .....	133
Komposer Ancaman .....	142
Bekerja dengan Threat Composer .....	143
Sumber daya .....	144
IAM izin untuk mengakses sumber daya .....	145

---

Menambah dan berinteraksi dengan sumber daya yang ada .....	145
Membuat dan mengedit sumber daya .....	147
Keamanan .....	150
Perlindungan data .....	150
Riwayat dokumen .....	152
.....	clviii

# AWS Toolkit for Visual Studio Code

Ini adalah panduan pengguna untuk AWS Toolkit for VS Code. Jika Anda mencari AWS Toolkit for Visual Studio, lihat [Panduan Pengguna untuk AWS Toolkit for Visual Studio](#).

## Apa itu AWS Toolkit for Visual Studio Code

Toolkit for VS Code adalah ekstensi open-source untuk editor Visual Studio Code (VS Code). Ekstensi ini memudahkan pengembang untuk mengembangkan, men-debug secara lokal, dan menyebarkan aplikasi tanpa server yang menggunakan Amazon Web Services (AWS).

Topik

- [Memulai dengan AWS Toolkit for Visual Studio Code](#)
- [Bekerja dengan AWS layanan dan alat](#)

## Informasi terkait

Gunakan sumber daya berikut untuk mengakses kode sumber untuk toolkit atau melihat masalah yang saat ini terbuka.

- [Kode Sumber](#)
- [Pelacak Masalah](#)

Untuk mempelajari lebih lanjut tentang editor Visual Studio Code, kunjungi <https://code.visualstudio.com/>.

# Pengembang Amazon Q dan Amazon CodeWhisperer

Mulai 30 April 2024, Amazon sekarang menjadi bagian dari CodeWhisperer Pengembang Amazon Q, ini termasuk saran kode sebaris dan pemindaian keamanan Pengembang Amazon Q. Unduh [IDEekstensi Pengembang Amazon Q dari VS Code Marketplace](#) untuk memulai.

Untuk detail tentang layanan Pengembang Amazon Q, lihat Panduan Pengguna [Pengembang Amazon Q](#). Untuk informasi terperinci tentang paket dan harga Amazon Q, lihat panduan [harga Amazon Q](#).

# Mengunduh Toolkit for VS Code

Anda dapat mengunduh, menginstal, dan mengatur AWS Toolkit for Visual Studio Code melalui VS Code Marketplace di IDE Anda. Untuk petunjuk terperinci, lihat bagian [Unduh dan instal](#) di topik Memulai Panduan Pengguna ini.

## Mengunduh Toolkit dari VS Code Marketplace

Atau, Anda dapat mengunduh file AWS Toolkit for Visual Studio Code instalasi dengan menavigasi ke [VS Code Marketplace](#) dari browser web Anda.

## Toolkit IDE tambahan dari AWS

Selain itu AWS Toolkit for Visual Studio Code, AWS juga menawarkan IDE Toolkit untuk JetBrains dan Visual Studio.

### AWS Toolkit for JetBrainslink

- Ikuti tautan ini untuk [mengunduh AWS Toolkit for JetBrains dari](#) JetBrains Marketplace.
- Untuk mempelajari selengkapnya AWS Toolkit for JetBrains, lihat Panduan [AWS Toolkit for JetBrains](#) Pengguna.

### Tautan Toolkit for Visual Studio

- Ikuti tautan ini untuk [Mengunduh Toolkit for Visual](#) Studio dari Marketplace Visual Studio.
- Untuk mempelajari lebih lanjut tentang Toolkit for Visual Studio, lihat Panduan Pengguna [Toolkit for Visual Studio](#).

# Memulai dengan AWS Toolkit for Visual Studio Code

AWS Toolkit for Visual Studio Code membuat AWS layanan dan sumber daya Anda tersedia, langsung dari lingkungan pengembangan terintegrasi VS Code (IDE) Anda.

Untuk membantu Anda memulai, topik berikut menjelaskan cara mengatur, menginstal, dan mengkonfigurasi AWS Toolkit for Visual Studio Code.

## Topik

- [Instalasi AWS Toolkit for Visual Studio Code](#)
- [Menghubungkan ke AWS](#)
- [Mengubah AWS Wilayah](#)
- [Mengkonfigurasi toolchain Anda](#)

## Instalasi AWS Toolkit for Visual Studio Code

### Prasyarat

Untuk mulai bekerja dengan AWS Toolkit for Visual Studio Code dari VS Code, prasyarat berikut harus dipenuhi. Untuk mempelajari lebih lanjut tentang mengakses semua AWS layanan dan sumber daya yang tersedia dari AWS Toolkit for Visual Studio Code, lihat [the section called “Prasyarat opsional”](#) bagian panduan ini.

- VS Code membutuhkan sistem operasi Windows, macOS, atau Linux.
- Ini AWS Toolkit for Visual Studio Code mengharuskan Anda untuk bekerja dari VS Code versi 1.73.0 atau versi yang lebih baru.

Untuk informasi tambahan tentang VS Code atau untuk mengunduh versi terbaru VS Code, lihat situs web [unduhannya VS Code](#).

## Mengunduh dan menginstal AWS Toolkit for Visual Studio Code

Anda dapat mengunduh, menginstal, dan mengatur AWS Toolkit for Visual Studio Code melalui VS Code Marketplace di situs Anda IDE. Atau, Anda dapat mengunduh file AWS Toolkit for Visual Studio Code instalasi dengan menavigasi ke [VS Code Marketplace](#) dari browser web Anda.

## Menginstal AWS Toolkit for Visual Studio Code dari VS Code IDE Marketplace

1. Buka AWS Toolkit for Visual Studio Code ekstensi di VS Code Anda IDE dengan tautan berikut: [Buka VS Code Marketplace](#).

### Note

Jika VS Code belum berjalan di mesin Anda, operasi ini mungkin memakan waktu beberapa saat saat VS Code sedang dimuat.

2. Dari AWS Toolkit for Visual Studio Code ekstensi di VS Code Marketplace, pilih Install untuk memulai proses instalasi.
3. Saat diminta, pilih untuk memulai ulang VS Code untuk menyelesaikan proses instalasi.

## Prasyarat opsional

Sebelum Anda dapat menggunakan fitur tertentu dari AWS Toolkit for Visual Studio Code, Anda harus memiliki yang berikut:

- Akun Amazon Web Services (AWS): AWS Akun bukanlah persyaratan untuk menggunakan akun AWS Toolkit for Visual Studio Code, tetapi fungsionalitas sangat terbatas tanpanya. Untuk mendapatkan AWS akun, buka [halaman AWS beranda](#). Pilih Buat AWS Akun, atau Selesaikan Daftar (jika Anda pernah mengunjungi situs sebelumnya).
- Pengembangan Kode — Relevan SDK untuk bahasa yang ingin Anda gunakan. Anda dapat mengunduh dari tautan berikut, atau menggunakan pengelola paket favorit Anda:
  - .NET SDK: <https://dotnet.microsoft.com/download>
  - Node.js SDK: <https://nodejs.org/en/download>
  - Python SDK: <https://www.python.org/downloads>
  - Java SDK: <https://aws.amazon.com/corretto/>
  - Pergi SDK: <https://golang.org/doc/install>
- AWS SAM CLI— Ini adalah AWS CLI alat yang membantu Anda mengembangkan, menguji, dan menganalisis aplikasi tanpa server Anda secara lokal. Ini tidak diperlukan untuk menginstal toolkit. Namun, kami menyarankan Anda menginstalnya (dan Docker, dijelaskan selanjutnya) karena diperlukan untuk fungsionalitas AWS Serverless Application Model (AWS SAM) apa pun, seperti [Membuat aplikasi tanpa server baru \(lokal\)](#).

Untuk informasi selengkapnya, lihat [Menginstal AWS SAM CLI](#) di [Panduan AWS Serverless Application Model Pengembang](#).

- Docker — AWS SAM CLI Membutuhkan platform wadah perangkat lunak sumber terbuka ini. Untuk informasi selengkapnya dan petunjuk unduhan, lihat [Docker](#).
- Package Manager — Manajer paket sehingga Anda dapat mengunduh dan berbagi kode aplikasi.
  - .NET: [NuGet](#)
  - Node.js: [npm](#)
  - Python: [pip](#)
  - Java: [Gradle atau Maven](#)

## Menghubungkan ke AWS

Sebagian besar sumber daya Amazon Web Services (AWS) dikelola melalui AWS akun. AWS Akun tidak diperlukan untuk menggunakan AWS Toolkit for Visual Studio Code, namun fungsi Toolkit terbatas tanpa koneksi.

Jika sebelumnya Anda telah menyiapkan AWS akun dan otentikasi melalui AWS layanan lain (seperti AWS Command Line Interface), maka AWS Toolkit for Visual Studio Code secara otomatis mendeteksi kredensialnya.

## Prasyarat

Jika Anda baru AWS atau belum membuat akun, maka ada 3 langkah utama untuk menghubungkannya AWS Toolkit for Visual Studio Code dengan AWS akun Anda:

1. Mendaftar untuk AWS akun: Anda dapat mendaftar AWS akun dari [portal AWS pendaftaran](#). Untuk informasi rinci tentang menyiapkan AWS akun baru, lihat topik [Ringkasan](#) di Panduan Pengguna AWS Pengaturan.
2. Menyiapkan otentikasi: Ada 3 metode utama untuk mengautentikasi dengan AWS akun Anda dari AWS Toolkit for Visual Studio Code Untuk mempelajari lebih lanjut tentang masing-masing metode ini, lihat topik [Otentikasi dan Akses](#) di Panduan Pengguna ini.
3. Mengautentikasi dengan AWS dari Toolkit: Anda dapat terhubung dengan AWS akun Anda dari Toolkit dengan menyelesaikan prosedur di bagian berikut dari Panduan Pengguna ini.

## Membuka panel Masuk

Selesaikan salah satu prosedur berikut untuk membuka panel Masuk AWS Toolkit.

Untuk membuka panel Masuk AWS Toolkit dari AWS Explorer:

1. Dari AWS Toolkit for Visual Studio Code, perluas EXPLORER.
2. Perluas Lebih Banyak Tindakan... Menu dengan memilih... ikon.
3. Dari Lebih Banyak Tindakan... menu, pilih Connect AWS to untuk membuka panel Masuk AWS Toolkit.

Untuk membuka panel Masuk AWS Toolkit menggunakan palet perintah VS Code:

1. Buka palet perintah dengan menekan **Shift+Command+P (Ctrl+Shift+P)** (Windows).
2. Masuk **AWS: Add a New Connection** ke bidang pencarian.
3. Pilih **AWS: Add a New Connection** untuk membuka panel Masuk AWS Toolkit.

## Menghubungkan ke AWS dari Toolkit

### Mengautentikasi dan terhubung dengan SSO

Untuk mengautentikasi dan terhubung dengan AWS menggunakan AWS IAM Identity Center, selesaikan prosedur berikut.

#### Note

Otentikasi dengan AWS Builder ID atau IAM Identity Center meluncurkan portal AWS otorisasi di browser web default Anda. Setiap kali kredensi Anda kedaluwarsa, proses ini harus diulang untuk memperbarui koneksi antara akun Anda AWS dan akun. AWS Toolkit for Visual Studio Code

### Mengautentikasi dan terhubung dengan Pusat AWS IAM Identitas

1. Dari panel Masuk AWS Toolkit, pilih tab Workforce, lalu pilih tombol Lanjutkan untuk melanjutkan.

2. Dari panel Masuk dengan Pusat IAM Identitas, masukkan Mulai URL untuk organisasi Anda. URL ini diberikan kepada Anda oleh admin atau help desk di perusahaan Anda.
3. Pilih AWS Wilayah Anda dari menu tarik-turun. Ini adalah AWS wilayah yang menghosting direktori identitas Anda.
4. Pilih tombol Lanjutkan dan konfirmasi bahwa Anda ingin membuka situs web permintaan AWS Otorisasi di browser web default Anda.
5. Ikuti petunjuk di browser web default Anda, Anda diberi tahu ketika proses otorisasi selesai, aman untuk menutup browser Anda, dan kembali ke Kode VS.

## Mengautentikasi dan terhubung dengan IAM Kredensial

Untuk mengautentikasi dan terhubung dengan AWS menggunakan IAM Credentials, selesaikan prosedur berikut.

### Mengautentikasi dan terhubung dengan IAM Kredensial

1. Dari panel Masuk AWS Toolkit, pilih IAM Credential, lalu pilih tombol Continue untuk melanjutkan.
2. Masukkan **Profile Name**, **Access Key**, dan **Secret Key** AWS akun Anda di bidang yang disediakan, lalu pilih tombol Lanjutkan untuk menambahkan profil ke file konfigurasi Anda dan menghubungkan Toolkit dengan akun Anda AWS .
3. Toolkit AWS Explorer diperbarui untuk menampilkan AWS layanan dan sumber daya Anda saat otentikasi selesai dan koneksi telah dibuat.

## Otentikasi untuk Amazon CodeCatalyst

Untuk mulai bekerja dengan CodeCatalyst dari Toolkit, autentikasi dan sambungkan dengan kredensial AWS Builder ID atau IAM Identity Center Anda.

Prosedur berikut menjelaskan cara mengautentikasi dan menghubungkan Toolkit dengan akun Anda AWS .

### Mengautentikasi dan terhubung dengan AWS Builder ID

1. Dari panel Masuk AWS Toolkit, pilih tab Workforce, lalu pilih tombol Lanjutkan untuk melanjutkan.
2. Di bagian atas SSO panel Masuk dengan, pilih tautan Lewati ke masuk.

- Ikuti petunjuk di browser web default Anda, Anda diberi tahu ketika proses otorisasi selesai, aman untuk menutup browser Anda, dan kembali ke Kode VS.

### Mengautentikasi dan terhubung dengan Pusat IAM Identitas

- Dari panel Masuk AWS Toolkit, pilih tab Workforce, lalu pilih tombol Lanjutkan untuk melanjutkan.
- Dari panel Masuk dengan Pusat IAM Identitas, masukkan Mulai URL untuk organisasi Anda. URL ini diberikan kepada Anda oleh admin atau help desk di perusahaan Anda.
- Pilih AWS Wilayah Anda dari menu tarik-turun. Ini adalah AWS wilayah yang menghosting direktori identitas Anda.
- Pilih tombol Lanjutkan dan konfirmasi bahwa Anda ingin membuka situs web permintaan AWS Otorisasi di browser web default Anda.
- Ikuti petunjuk di browser web default Anda, Anda diberi tahu ketika proses otorisasi selesai, aman untuk menutup browser Anda, dan kembali ke Kode VS.

## Mengubah AWS Wilayah

AWS Wilayah menentukan di mana AWS sumber daya Anda dikelola. AWS Wilayah default Anda terdeteksi ketika Anda terhubung ke AWS akun Anda dari AWS Toolkit for Visual Studio Code, secara otomatis ditampilkan di AWS Explorer.

Bagian berikut menjelaskan cara menambahkan atau menyembunyikan Region dari AWS Explorer.

### Menambahkan Wilayah ke AWS Explorer

Selesaikan prosedur berikut untuk menambahkan Wilayah ke AWS Explorer.

- Dari VS Code, buka Command Palette dengan memperluas View pada menu utama dan memilih Command Palette. Atau gunakan tombol pintas berikut:
  - Windows dan Linux — Tekan **Ctrl+Shift+P**.
  - macOS — Tekan **Shift+Command+P**
- Dari Palet Perintah, cari **AWS: Show or Hide Regions** dan pilih AWS: Tampilkan atau Sembunyikan Wilayah untuk menampilkan daftar Wilayah yang tersedia.
- Dari daftar, pilih AWS Wilayah yang ingin Anda tambahkan ke AWS Explorer.

4. Pilih tombol OK untuk mengonfirmasi pilihan Anda dan memperbarui AWS Explorer.

## Sembunyikan Wilayah dari AWS Explorer

Untuk menyembunyikan Wilayah dari tampilan AWS Explorer, selesaikan prosedur berikut.

1. Dari AWS Explorer, cari AWS Wilayah yang ingin Anda sembunyikan.
2. Buka menu konteks untuk (klik kanan) Wilayah yang ingin Anda sembunyikan.
3. Pilih Tampilkan atau Sembunyikan Wilayah untuk membuka opsi AWS: Tampilkan atau Sembunyikan Wilayah di VS Code.
4. Hapus pilihan Wilayah yang ingin Anda sembunyikan dalam tampilan AWS Explorer.

## Mengkonfigurasi toolchain Anda

AWS Toolkit for Visual Studio Code mendukung berbagai bahasa di semua AWS layanan. Bagian berikut menjelaskan cara mengonfigurasi toolchain Anda untuk berbagai bahasa.

### Mengkonfigurasi toolchain untuk .NET

1. Pastikan bahwa Anda memiliki AWS Toolkit for VS Code [diinstal](#).
2. Instal [ekstensi C #](#). Ekstensi ini memungkinkan VS Code untuk men-debug aplikasi .NET Core.
3. Buka aplikasi AWS Serverless Application Model (AWS SAM), atau [buat satu](#).
4. Buka folder yang berisi `template.yaml`.

### Mengkonfigurasi toolchain for Node.js

1. Pastikan bahwa Anda memiliki AWS Toolkit for VS Code [diinstal](#).
2. Buka AWS SAM aplikasi, atau [buat satu](#).
3. Buka folder yang berisi `template.yaml`.

**Note**

Ketika men-debug fungsi TypeScript Lambda langsung dari kode sumber (konfigurasi peluncuran memiliki "target": "code"), TypeScript compiler harus diinstal baik secara global atau di proyek `Andapackage.json`.

## Mengkonfigurasi toolchain untuk Python

1. Pastikan bahwa Anda memiliki AWS Toolkit for VS Code [diinstal](#).
2. Instal [ekstensi Python untuk Visual Studio Code](#). Ekstensi ini memungkinkan VS Code untuk men-debug aplikasi Python.
3. Buka AWS SAM aplikasi, atau [buat satu](#).
4. Buka folder yang berisi `template.yaml`.
5. Buka terminal di root aplikasi Anda, dan konfigurasi `virtualenv` dengan menjalankan `python -m venv ./venv`.

**Note**

Anda hanya perlu mengkonfigurasi `virtualenv` sekali per sistem.

6. Aktifkan `virtualenv` dengan menjalankan salah satu berikut:
  - Bash shell: `./venv/Scripts/activate`
  - PowerShell: `./venv/Scripts/Activate.ps1`

## Mengkonfigurasi toolchain untuk Java

1. Pastikan bahwa Anda memiliki AWS Toolkit for VS Code [diinstal](#).
2. Instal [ekstensi Java dan Java 11](#). Ekstensi ini memungkinkan VS Code untuk mengenali fungsi Java.
3. Instal [ekstensi debugger Java](#). Ekstensi ini memungkinkan VS Code untuk men-debug aplikasi Java.
4. Buka AWS SAM aplikasi, atau [buat satu](#).

5. Buka folder yang berisitemplate.yaml.

## Mengkonfigurasi toolchain untuk Go

1. Pastikan bahwa Anda memilikiAWS Toolkit for VS Code [diinstal](#).
2. Go 1.14 atau lebih tinggi diperlukan untuk men-debug fungsi Go Lambda.
3. Instal [ekstensi Go](#).

### Note

Versi 0.25.0 atau lebih tinggi diperlukan untuk debugging Go1.15+runtimes

4. Instal alat Go menggunakan [palet perintah](#):
  - a. Dari palet perintah, pilihGo: Install/Update Tools.
  - b. Dari set kotak centang, pilihdlv dangopls.
5. BukaAWS SAM aplikasi, atau [buat satu](#).
6. Buka folder yang berisitemplate.yaml.

## Menggunakan Toolchain Anda

Setelah Anda mengatur toolchain Anda, Anda dapat menggunakannya untuk [menjalankan atau men-debug](#)AWS SAM aplikasi.

# Otentikasi dan akses untuk AWS Toolkit for Visual Studio Code

Anda tidak perlu mengotentikasi dengan AWS untuk mulai bekerja dengan. AWS Toolkit for Visual Studio Code Namun, sebagian besar AWS sumber daya dikelola melalui AWS akun. Untuk mengakses semua AWS Toolkit for Visual Studio Code layanan dan fitur, Anda harus mengotentikasi dengan AWS IAM Identity Center kredensi AWS Builder ID atau IAM.

Topik berikut berisi detail tambahan tentang setiap jenis kredensi.

Untuk detail tentang cara menyambung ke AWS dalam AWS Toolkit for Visual Studio Code dengan kredensial Anda yang ada, lihat AWS topik [Menyambung ke](#) dalam Panduan Pengguna ini.

Topik

- [AWS Pusat Identitas IAM](#)
- [AWS IAMkredensialnya](#)
- [AWS Builder ID untuk pengembang](#)
- [Menggunakan proses proses kredensi kredensi](#)

## AWS Pusat Identitas IAM

AWS IAM Identity Center adalah praktik terbaik yang disarankan untuk mengelola otentikasi AWS akun Anda.

Untuk petunjuk terperinci tentang cara menyiapkan Pusat Identitas IAM untuk Kit Pengembangan Perangkat Lunak (SDK), lihat bagian [otentikasi Pusat Identitas IAM](#) pada AWS SDK dan Panduan Referensi Alat.

Untuk detail tentang cara mengotentikasi dan menghubungkan AWS toolkit dengan kredensial Pusat Identitas IAM yang ada, lihat [Connect to AWS](#) topic di Panduan Pengguna ini.

## AWS IAMkredensialnya

AWS IAMotentikasi kredensial dengan AWS akun Anda melalui kunci akses yang disimpan secara lokal.

Untuk detail tentang cara mengautentikasi dan menghubungkan AWS toolkit dengan AWS IAM kredensial yang ada, lihat AWS topik [Connect to](#) di Panduan Pengguna ini.

Bagian berikut menjelaskan cara mengatur IAM kredensial untuk mengautentikasi dengan AWS akun Anda dari AWS Toolkit for Visual Studio Code

### Important

Sebelum menyiapkan IAM kredensial untuk mengautentikasi dengan AWS akun Anda, perhatikan bahwa:

- Jika Anda telah menyetel IAM kredensial melalui AWS layanan lain (seperti AWS CLI), maka AWS Toolkit for Visual Studio Code secara otomatis mendeteksi kredensial tersebut dan membuatnya tersedia di VS Code.
- AWS merekomendasikan menggunakan otentikasi Pusat IAM Identitas. Untuk informasi tambahan tentang praktik AWS IAM terbaik, lihat [Praktik terbaik keamanan di IAM](#) bagian Panduan Pengguna AWS Identity and Access Management.
- Untuk menghindari risiko keamanan, jangan gunakan IAM pengguna untuk otentikasi saat mengembangkan perangkat lunak yang dibuat khusus atau bekerja dengan data nyata. Sebagai gantinya, gunakan federasi dengan penyedia identitas seperti [Apa itu Pusat IAM Identitas?](#) dalam AWS IAM Identity Center User Guide.

## Membuat IAM pengguna

Sebelum Anda dapat mengatur AWS Toolkit for Visual Studio Code untuk mengautentikasi dengan AWS akun Anda, Anda harus menyelesaikan Langkah 1: Buat IAM pengguna Anda dan Langkah 2: Dapatkan kunci akses Anda di topik [Autentikasi menggunakan kredensial jangka panjang di Panduan Referensi](#) Alat AWS SDK dan Alat.

### Note

Langkah 3: Perbarui file kredensial bersama di Panduan Referensi Alat AWS SDKs dan adalah opsional.

Jika Anda menyelesaikan Langkah 3, AWS Toolkit for Visual Studio Code secara otomatis mendeteksi kredensial Anda selama berada di [the section called “Membuat file kredensial bersama dari AWS Toolkit for Visual Studio Code”](#) bawah ini.

Jika Anda belum menyelesaikan Langkah 3, AWS Toolkit for Visual Studio Code memandu Anda melalui proses pembuatan `credentials` file seperti yang dijelaskan di [the section called “Membuat file kredensial bersama dari AWS Toolkit for Visual Studio Code”](#) bawah ini.

## Membuat file kredensial bersama dari AWS Toolkit for Visual Studio Code

File konfigurasi bersama dan file kredensial bersama menyimpan konfigurasi dan informasi kredensi untuk akun Anda. AWS Untuk informasi selengkapnya tentang konfigurasi dan kredensial bersama, lihat Di [mana pengaturan konfigurasi disimpan?](#) bagian dalam Panduan AWS Command Line Interface Pengguna.

Membuat file kredensial bersama melalui AWS Toolkit for Visual Studio Code

1. Buka palet perintah dengan menekan **Shift+Command+P** (**Ctrl+Shift+P** Windows).
2. Masuk **AWS: Add a New Connection** ke bidang pencarian.
3. Pilih **AWS: Add a New Connection** untuk membuka panel Masuk AWS Toolkit.
4. Dari panel Masuk AWS Toolkit, pilih IAMCredential, lalu pilih tombol Continue untuk melanjutkan.
5. Masukkan **Profile Name**, **Access Key**, dan **Secret Key** AWS akun Anda di bidang yang disediakan, lalu pilih tombol Lanjutkan untuk menambahkan profil ke file konfigurasi Anda dan menghubungkan Toolkit dengan akun Anda AWS .
6. Toolkit AWS Explorer diperbarui untuk menampilkan AWS layanan dan sumber daya Anda saat otentikasi selesai dan koneksi telah dibuat.

### Note

Dalam contoh ini, asumsikan bahwa `[Profile_Name]` berisi kesalahan sintaks dan menyebabkan otentikasi gagal.

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

Berikut ini adalah contoh pesan log yang dihasilkan sebagai respons terhadap upaya otentikasi yang gagal.

```
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.
```

## Tambahkan profil kredensi tambahan

Anda dapat menambahkan beberapa kredensi ke file konfigurasi Anda. Untuk melakukannya, buka Command Palette dan pilih AWS Toolkit Create Credentials Profile. Ini akan membuka file kredensial. Pada halaman ini, Anda dapat menambahkan profil baru di bawah profil pertama Anda, seperti yang ditunjukkan pada contoh berikut:

```
# 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 untuk pengembang

AWS Builder ID adalah AWS akun tambahan yang bersifat opsional atau diperlukan untuk AWS layanan tertentu. Untuk informasi lebih lanjut tentang metode autentikasi AWS Builder ID, lihat topik [Masuk dengan AWS Builder ID](#) di Panduan Pengguna AWS Masuk.

Untuk detail tentang cara mengautentikasi dan menghubungkan AWS toolkit dengan AWS Builder ID yang ada, lihat AWS topik [Connect to](#) di Panduan Pengguna ini.

## Menggunakan proses proses kredensi kredensi

Anda dapat mengkonfigurasi AWS Toolkit for Visual Studio Code untuk proses kredensi yang tidak didukung secara langsung oleh AWS, dengan memodifikasi `shared config file`.

Memodifikasi `shared config file` untuk proses kredensi adalah sama untuk kedua AWS Toolkit for Visual Studio Code dan AWS Command Line Interface. Untuk informasi mendetail tentang cara menyiapkan kredensial eksternal, lihat [kredensi Sumber dengan topik proses eksternal](#) di Panduan AWS Command Line Interface Pengguna.

# Bekerja dengan AWS layanan dan alat

AWS Toolkit for Visual Studio Code Membuat AWS layanan, alat, dan sumber daya tersedia untuk Anda, langsung di VS Code. Berikut ini adalah daftar topik panduan yang mencakup setiap layanan Toolkit for VS Code dan fitur-fiturnya. Pilih layanan atau alat untuk informasi lebih lanjut tentang apa yang dilakukannya, cara mengaturnya, dan bekerja dengan fitur-fitur dasar.

## Topik

- [Bekerja dengan fitur eksperimental](#)
- [Bekerja dengan AWS Layanan di AWS Explorer](#)
- [Amazon CodeCatalyst untuk Kode VS](#)
- [Bekerja dengan Amazon API Gateway](#)
- [Menggunakan AWS App Runner dengan AWS Toolkit for Visual Studio Code](#)
- [AWSKomposer Aplikasi](#)
- [AWS CDKuntuk VS Code](#)
- [Bekerja denganAWS CloudFormationtumpukan](#)
- [Bekerja denganCloudWatchLog dengan menggunakanAWS Toolkit for Visual Studio Code](#)
- [Bekerja dengan Amazon Elastic Container Registry](#)
- [Bekerja dengan Layanan Wadah Amazon Elastic](#)
- [Bekerja dengan Amazon EventBridge](#)
- [AWS IAMPenganalisis Akses](#)
- [Bekerja denganAWS IoTdiAWS Toolkit for Visual Studio Code](#)
- [Bekerja denganAWS LambdaFungsi](#)
- [Amazon Redshift di Toolkit untuk VS Code](#)
- [Bekerja dengan Amazon S3](#)
- [Bekerja dengan irserver](#)
- [Menggunakan dokumen otomatisasi Systems Manager](#)
- [Bekerja dengan AWS Step Functions](#)
- [Bekerja dengan Threat Composer](#)
- [Bekerja dengan sumber daya](#)

## Bekerja dengan fitur eksperimental

Fitur eksperimental menawarkan akses awal ke fitur di AWS Toolkit for Visual Studio Code sebelum resmi dirilis.

### Warning

Karena fitur eksperimental terus diuji dan diperbarui, mereka mungkin memiliki masalah kegunaan. Dan fitur eksperimental dapat dihapus dari AWS Toolkit for Visual Studio Code tanpa pemberitahuan.

Anda dapat mengaktifkan fitur eksperimental untuk spesifik AWS services AWS Toolkit bagian Pengaturan panel di IDE Kode VS Anda.

1. Untuk mengedit AWS pengaturan di VS Code, pilih Berkas, Preferensi, Pengaturan.
2. Di Pengaturan panel, memperluas Ekstensi dan pilihlah AWS Toolkit.
3. Di bawah AWS: percobaan, pilih kotak centang untuk fitur eksperimental yang ingin Anda akses sebelum rilis. Jika Anda ingin mematikan fitur eksperimental, kosongkan kotak centang yang relevan.

## Bekerja dengan AWS Layanan di AWS Explorer

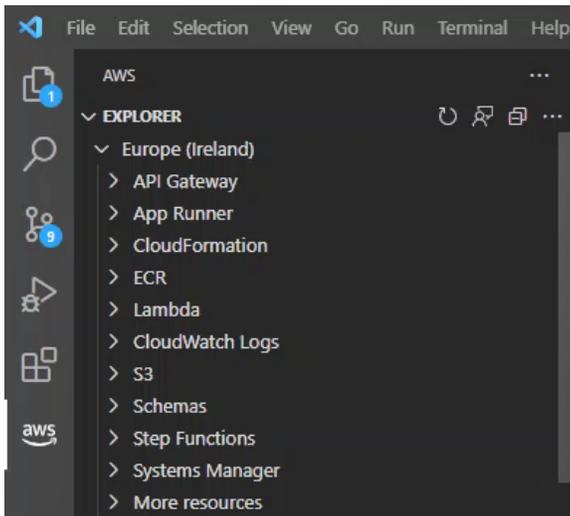
AWS Explorer memberi Anda pandangan tentang beberapa AWS layanan yang dapat Anda gunakan saat menggunakan AWS Toolkit for Visual Studio Code.

Bagian ini memberikan informasi tentang cara mengakses dan menggunakan AWS Explorer di VS Code. Ini mengasumsikan bahwa Anda telah [menginstal dan mengonfigurasi](#) Toolkit for VS Code di sistem Anda.

Beberapa poin penting:

- Jika toolkit diinstal dan dikonfigurasi dengan benar, Anda akan melihat item di Explorer. AWS Untuk melihat AWS Explorer, pilih AWS ikon di bilah Aktivitas.

Sebagai contoh:



- Fitur tertentu memerlukan AWS izin tertentu. Misalnya, untuk melihat AWS Lambda fungsi di AWS akun Anda, kredensial yang Anda konfigurasi [Otentikasi dan akses](#) harus menyertakan setidaknya izin Lambda hanya-baca. Lihat topik berikut untuk informasi selengkapnya tentang izin yang dibutuhkan setiap fitur.
- Jika Anda ingin berinteraksi dengan AWS layanan yang tidak langsung terlihat di AWS Explorer, Anda dapat membuka Lebih banyak sumber daya dan memilih dari ratusan sumber daya yang dapat ditambahkan ke antarmuka.

Misalnya, Anda dapat memilih `AWS Toolkit:CodeArtifact: :Repository` dari pemilihan jenis sumber daya yang tersedia. Setelah jenis sumber daya ini ditambahkan ke Sumber daya lainnya, Anda dapat memperluas entri untuk melihat daftar sumber daya yang membuat CodeArtifact repositori berbeda dengan properti dan atributnya sendiri. Selain itu, Anda dapat menjelaskan properti dan atribut sumber daya dalam template berformat JSON, yang dapat disimpan untuk membuat sumber daya baru di Cloud. AWS

## Amazon CodeCatalyst untuk Kode VS

### Apa itu Amazon CodeCatalyst?

Amazon CodeCatalyst adalah ruang kolaborasi berbasis cloud untuk tim pengembangan perangkat lunak. Melalui AWS Toolkit for Visual Studio Code, Anda dapat melihat dan mengelola CodeCatalyst sumber daya Anda langsung dari VS Code. Anda juga dapat bekerja langsung di cloud dengan meluncurkan CodeCatalyst Dev Environment. Untuk informasi selengkapnya tentang CodeCatalyst layanan ini, lihat Panduan CodeCatalyst Pengguna [Amazon](#).

Topik berikut menjelaskan cara menghubungkan VS Code dengan CodeCatalyst, dan cara bekerja dengan CodeCatalyst dari Toolkit for VS Code.

Topik

- [Memulai CodeCatalyst dan Toolkit for VS Code](#)
- [Bekerja dengan CodeCatalyst sumber daya Amazon di VS Code](#)
- [Bekerja dengan Toolkit di Lingkungan Pengembang](#)
- [Memecahkan masalah Amazon CodeCatalyst dan VS Code](#)

## Memulai CodeCatalyst dan Toolkit for VS Code

Untuk mulai bekerja dengan CodeCatalyst di VS Code, ikuti prosedur ini.

Topik

- [Membuat CodeCatalyst akun](#)
- [Menghubungkan AWS Toolkit dengan CodeCatalyst](#)

## Membuat CodeCatalyst akun

Anda harus memiliki ID AWS Builder aktif atau AWS IAM Identity Center kredensial untuk terhubung CodeCatalyst dari Toolkit for VS Code. Untuk mempelajari lebih lanjut tentang AWS Builder ID, IAM Identity Center, dan CodeCatalyst kredensialnya, lihat CodeCatalyst bagian [Menyiapkan dengan](#) di CodeCatalystPanduan Pengguna.

## Menghubungkan AWS Toolkit dengan CodeCatalyst

Untuk menghubungkan AWS Toolkit dengan CodeCatalyst akun Anda, lihat CodeCatalyst bagian [Otentikasi untuk Amazon](#) di AWS topik Menghubungkan ke Panduan Pengguna ini.

## Bekerja dengan CodeCatalyst sumber daya Amazon di VS Code

Bagian berikut memberikan gambaran umum tentang fitur pengelolaan CodeCatalyst sumber daya Amazon yang tersedia dari Toolkit for VS Code.

Untuk informasi selengkapnya tentang Lingkungan Pengembang dan cara mengaksesnya CodeCatalyst, lihat bagian [Lingkungan Pengembang](#) di CodeCatalyst Panduan Pengguna Amazon.

Bagian berikut menjelaskan cara membuat, membuka, dan bekerja dengan Dev Environments dari VS Code.

## Topik

- [Mengkloning repositori](#)
- [Membuka Lingkungan Pengembang](#)
- [Membuat Lingkungan CodeCatalyst Dev](#)
- [Membuat Lingkungan Pengembang dari repositori pihak ketiga](#)
- [CodeCatalyst perintah dalam VS Code](#)

## Mengkloning repositori

CodeCatalyst adalah layanan berbasis cloud yang mengharuskan Anda terhubung ke cloud untuk mengerjakan CodeCatalyst proyek. Jika Anda lebih suka mengerjakan proyek secara lokal, Anda dapat mengkloning CodeCatalyst repositori Anda ke mesin lokal Anda dan menyinkronkannya dengan CodeCatalyst proyek Anda secara online, saat berikutnya Anda terhubung ke cloud.

Untuk mengkloning repositori dari CodeCatalyst akun Anda ke VS Code dengan AWS Toolkit, selesaikan langkah-langkah berikut:

### Note

Jika Anda mengkloning repositori dari layanan pihak ke-3, Anda mungkin diminta untuk mengautentikasi dengan kredensi layanan tersebut.

Saat repositori sedang dikloning, VS Code menampilkan kemajuan di jendela status Cloning Repository. Setelah repositori dikloning, Apakah Anda ingin membuka repositori kloning? pesan muncul.

1. Dari Toolkit for VS Code, perluas explorer DEVELOPERTOOLS.
2. Perluas CodeCatalyst, pilih Clone Repository.
3. Dari dialog Select a CodeCatalyst Repository, cari repositori yang ingin Anda kloning, lalu pilih untuk membuka dialog Choose a folder to clone.
4. Pilih Lokasi Repositori untuk menutup prompt dan mulai mengkloning repositori.
5. Dari jendela dialog, pilih salah satu dari berikut ini untuk menyelesaikan proses kloning:

- Untuk membuka repositori Anda di jendela VS Code Anda saat ini, pilih Buka.
- Untuk membuka repositori Anda di jendela VS Code baru, pilih Buka di jendela baru.
- Untuk menyelesaikan proses kloning tanpa membuka repositori Anda, tutup jendela dialog.

## Membuka Lingkungan Pengembang

Untuk membuka Lingkungan Dev yang ada di VS Code, selesaikan langkah-langkah berikut.

### Note

Memilih Lingkungan Dev memulai proses untuk menghubungkan VS Code CodeCatalyst dengan membuka Dev Environment Anda. Selama proses ini, VS Code menampilkan pembaruan kemajuan di jendela CodeCatalyst status. Jendela status diperbarui saat proses selesai.

- Jika Lingkungan Pengembang gagal dibuka, status akan diperbarui dengan informasi tentang mengapa proses gagal dan tautan untuk membuka log proses.
- Jika proses berhasil, Lingkungan Dev Anda terbuka di jendela baru, dari VS Code.

1. Dari Toolkit for VS Code, perluas explorer DEVELOPERTOOLS.
2. Perluas CodeCatalyst dan pilih Open Dev Environment untuk membuka dialog Select a CodeCatalyst Dev Environment di VS Code.
3. Dari dialog Select a CodeCatalyst Dev Environment, pilih Dev Environment yang ingin Anda buka.

## Membuat Lingkungan CodeCatalyst Dev

Untuk membuat Lingkungan Dev baru, selesaikan langkah-langkah berikut:

### Note

Saat membuat Lingkungan Dev baru, perhatikan hal berikut:

- AWS merekomendasikan agar Anda menentukan alias karena menyederhanakan organisasi dan meningkatkan kemampuan pencarian untuk Lingkungan Pengembang.

- Dev Environments menyimpan pekerjaan Anda terus-menerus. Ini berarti bahwa Lingkungan Pengembang Anda dapat dihentikan tanpa kehilangan pekerjaan Anda. Menghentikan Lingkungan Pengembang Anda mengurangi biaya yang diperlukan untuk membuat Lingkungan Dev Anda tetap aktif dan berjalan.
- Penyimpanan adalah satu-satunya pengaturan yang tidak dapat diubah setelah Lingkungan Dev Anda dibuat.
- VS Code menampilkan kemajuan Lingkungan Pengembang Anda yang dibuat di jendela status. Setelah Dev Environment dibuat, VS Code membuka Dev Environment di jendela baru dan Apakah Anda mempercayai penulis file di folder ini? prompt juga muncul. Setujui syarat dan ketentuan untuk terus bekerja di Lingkungan Pengembang Anda.

1. Dari Toolkit for VS Code, perluas explorer DEVELOPERTOOLS.
2. Perluas CodeCatalyst, dan pilih opsi Create Dev Environment untuk membuka menu Create a CodeCatalyst Dev Environment di VS Code.
3. Dari bagian Kode Sumber, pilih salah satu opsi berikut:
  - Gunakan CodeCatalyst Repositori yang ada: Membuat Lingkungan Dev dari repositori yang ada. CodeCatalyst Anda harus memilih CodeCatalyst Proyek dan Cabang.
  - Buat Lingkungan Dev kosong: Membuat Lingkungan Dev kosong.
4. (Opsional) Dari bagian Alias, masukkan nama alternatif untuk Lingkungan Pengembang Anda.
5. (Opsional) Dari bagian Konfigurasi Lingkungan Dev, ubah pengaturan berikut untuk memenuhi kebutuhan spesifik Anda.
  - Compute: Pilih Edit Compute untuk mengubah jumlah daya pemrosesan dan RAM yang ditetapkan ke sistem Anda.
  - Timeout: Pilih Edit Timeout untuk mengubah jumlah waktu idle sistem yang diizinkan sebelum Lingkungan Dev Anda dihentikan.
  - Penyimpanan: Pilih Edit Ukuran Penyimpanan untuk mengubah jumlah ruang penyimpanan yang ditetapkan ke sistem Anda.
6. Pilih Create Dev Environment untuk membuat lingkungan pengembangan cloud baru Anda.

## Membuat Lingkungan Pengembang dari repositori pihak ketiga

Anda dapat membuat Lingkungan Dev dari repositori pihak ketiga dengan menautkan ke repositori sebagai sumber.

Menautkan ke repositori pihak ketiga sebagai sumber ditangani di tingkat proyek di CodeCatalyst. Untuk petunjuk dan detail tambahan tentang cara menghubungkan repositori pihak ketiga ke Lingkungan Pengembang Anda, lihat topik [Menautkan repositori sumber](#) di Panduan Pengguna Amazon CodeCatalyst.

## CodeCatalyst perintah dalam VS Code

Ada perintah VS Code tambahan yang ditetapkan ke fitur CodeCatalyst terkait yang tidak ditampilkan secara langsung di AWS Toolkit.

Untuk melihat daftar perintah yang ditetapkan CodeCatalyst dari palet perintah, selesaikan langkah-langkah berikut:

1. Dari Toolkit for VS Code, perluas explorer DEVELOPERTOOLS.
2. Pilih Tampilkan CodeCatalyst Perintah untuk membuka Palet Perintah dengan pencarian yang telah diisi sebelumnya. CodeCatalyst
3. Pilih CodeCatalyst perintah dari daftar untuk mengaktifkannya.

## Bekerja dengan Toolkit di Lingkungan Pengembang

Dev Environments adalah lingkungan komputasi virtual untuk Amazon CodeCatalyst. Bagian berikut menjelaskan cara membuat, meluncurkan, dan bekerja dari Dev Environments menggunakan AWS Toolkit for Visual Studio Code.

Untuk informasi mendetail tentang Lingkungan Pengembang, lihat topik [Lingkungan Pengembang](#) di CodeCatalyst Panduan Pengguna Amazon.

## Mengonfigurasi Lingkungan Dev Anda dengan devfiles

devfileSpesifikasi adalah format standar terbuka YAML yang dapat digunakan untuk menentukan konfigurasi untuk Lingkungan Dev. Setiap Lingkungan Dev memiliki devfile. Jika Anda membuat Lingkungan Dev tanpa repositori atau dari repositori yang tidak berisi devfile, default diterapkan ke sumber secara otomatis. Devfiles dapat diperbarui dari CodeCatalyst atau AndaiDE. Proses untuk memperbarui devfile dalam instance VS Code lokal atau jarak jauh identik, tetapi jika Anda

memperbarui devfile secara lokal, Anda harus mendorong pembaruan ke repositori sumber Anda sebelum pembaruan diterapkan.

Untuk informasi mendetail tentang mengonfigurasi Lingkungan Pengembang dengan devfiles, lihat [topik Mengonfigurasi Lingkungan Pengembang Anda di Panduan Pengguna](#) Amazon. CodeCatalyst

Prosedur berikut menjelaskan cara mengedit devfile Anda dari instance Toolkit jarak jauh saat sedang berjalan di Lingkungan Dev.

#### Important

Jika Anda mengedit Devfile dari VS Code, perhatikan hal berikut:

- Mengubah nama devfile atau nama komponen devfile menggantikan isi direktori root Anda. Semua konten sebelumnya hilang dan tidak dapat dipulihkan.
- Jika Anda membuat Lingkungan Dev tanpa devfile di folder root atau Lingkungan Dev yang tidak terkait dengan repositori sumber, devfile dengan pengaturan konfigurasi default dibuat untuk Lingkungan Dev Anda saat Anda membuatnya.
- Untuk petunjuk tentang cara menentukan dan mengonfigurasi AndaDevfile, lihat dokumentasi [Menambahkan Perintah](#) di situs web [devfile.io](#).

1. Dari Toolkit for VS Code, perluas explorer DEVELOPERTOOLS.
2. Perluas CodeCatalyst dan pilih Buka Devfile untuk membuka `devfile.yaml` di jendela editor baru, dalam Lingkungan Dev Anda saat ini.
3. Dari editor VS Code, perbarui devfile Anda, lalu simpan perubahan Anda.
4. Lain kali Anda meluncurkan Lingkungan Dev Anda, konfigurasi diperbarui agar sesuai dengan spesifikasi yang ditentukan dalam AndaDevfile.

## Mengautentikasi dan menghubungkan ke AWS dari Lingkungan Dev Anda

Untuk mengakses semua AWS sumber daya Anda dari Lingkungan Pengembang, Anda harus mengautentikasi dan menghubungkan instance Toolkit jarak jauh Anda dengan akun Anda. AWS Instance remote Toolkit secara otomatis mengautentikasi dengan kredensi yang diwarisi dari instance lokal Toolkit Anda saat Dev Environment Anda diluncurkan.

Prosedur untuk memperbarui kredensial Anda untuk instance Toolkit jarak jauh identik dengan pengalaman otentikasi di instance lokal Toolkit Anda. Untuk petunjuk terperinci tentang cara

memperbarui kredensi, mengautentikasi, dan menghubungkan AWS dari Toolkit, lihat AWS bagian [Menghubungkan ke](#) dalam topik Memulai Panduan Pengguna ini.

Untuk informasi tambahan tentang masing-masing metode AWS otentikasi yang kompatibel dengan AWS Toolkit for Visual Studio Code, lihat topik [Otentikasi dan akses](#) di Panduan Pengguna ini.

## Bekerja dengan Toolkit for VS Code di Lingkungan Dev

Setelah Anda membuka atau membuat Lingkungan Pengembang di VS Code, Anda dapat bekerja dari Toolkit for VS Code, mirip dengan yang Anda bisa dari instance lokal VS Code. Lingkungan Dev yang menjalankan VS Code dikonfigurasi untuk menginstal AWS Toolkit secara otomatis dan terhubung dengan AWS Builder ID Anda.

### Menghentikan Lingkungan Pengembang

Untuk menghentikan Lingkungan Dev Anda saat ini:

1. Dari Toolkit for VS Code, perluas explorer DEVELOPERTOOLS.
2. Perluas CodeCatalyst dan pilih Stop Dev Environment.
3. Saat diminta oleh VS Code, konfirmasi bahwa Anda ingin menghentikan Lingkungan Pengembang Anda.
4. Lingkungan Pengembang Anda telah berhasil dihentikan ketika VS Code menutup koneksi jarak jauh dan kembali ke instance pengembangan lokal.

### Membuka pengaturan Lingkungan Dev

Untuk membuka pengaturan untuk Lingkungan Dev Anda saat ini, selesaikan langkah-langkah berikut:

#### Note

Anda tidak dapat mengubah jumlah ruang penyimpanan yang ditetapkan ke Lingkungan Pengembang Anda setelah dibuat.

1. Dari Toolkit for VS Code, perluas explorer DEVELOPERTOOLS.
2. Perluas CodeCatalyst dan pilih Buka Pengaturan untuk membuka tampilan Pengaturan Lingkungan Dev, untuk Lingkungan Dev Anda saat ini.

3. Dari tampilan Pengaturan Lingkungan Dev, bagian berikut berisi opsi untuk Lingkungan Pengembang Anda:
  - Alias: Lihat dan ubah Alias yang ditetapkan ke Lingkungan Pengembang Anda.
  - Status: Lihat status Lingkungan Pengembang Anda saat ini, proyek yang ditetapkan, dan hentikan lingkungan Anda.
  - Devfile: Lihat nama dan lokasi Devfile untuk Lingkungan Dev Anda. Buka tombol Anda Devfile dengan memilih tombol Buka di Editor.
  - Pengaturan Komputasi: Ubah ukuran dan Panjang Batas Waktu default untuk Lingkungan Pengembang Anda.

## Memecahkan masalah Amazon CodeCatalyst dan VS Code

Topik berikut membahas potensi masalah teknis saat bekerja dengan Amazon CodeCatalyst dan VS Code.

Topik

- [Versi Kode VS](#)
- [Izin untuk Amazon CodeCatalyst](#)
- [Menghubungkan ke Lingkungan Pengembang dari Toolkit for VS Code](#)

### Versi Kode VS

Versi VS Code Anda diharapkan dapat menyiapkan handler untuk `vscode://` URIs sistem Anda. Tanpa handler ini, Anda tidak dapat mengakses semua CodeCatalyst fitur dari AWS Toolkit. Misalnya, Anda mengalami kesalahan saat meluncurkan Lingkungan Pengembang dari VS Code Insiders. Ini karena VS Code Insiders menangani `vscode-insiders://` URIs dan tidak menangani `vscode://`URIs.

### Izin untuk Amazon CodeCatalyst

Berikut ini adalah persyaratan izin file untuk bekerja dengan CodeCatalyst dari AWS Toolkit for Visual Studio Code:

- Tetapkan izin akses Anda sendiri untuk `~/ .ssh/config` file Anda ke `read` dan `write`. Batasi `write` izin untuk semua pengguna lain.

- Tetapkan izin akses Anda untuk `~/.ssh/id_rsa` file `~/.ssh/id_dsa` dan `read` hanya untuk file. Batasi `read`, `write` dan `execute` izin untuk semua pengguna lain.
- `globals.context.globalStorageUri.fsPathFile` Anda harus berada di lokasi yang dapat ditulis.

## Menghubungkan ke Lingkungan Pengembang dari Toolkit for VS Code

Jika Anda menerima kesalahan berikut saat mencoba menyambung ke Lingkungan Pengembang dari: AWS Toolkit for Visual Studio Code

Anda `~/.ssh/config` memiliki `aws-devenv-*` bagian yang mungkin kedaluwarsa.

- Pilih konfigurasi Buka.. tombol untuk membuka `~/.ssh/config` file Anda di VS Code Editor.
- Dari Editor, pilih dan hapus konten Host `aws-devenv-*` bagian.
- Simpan perubahan yang Anda buat ke Host `aws-devenv-*` dari `~/.ssh/config`. Kemudian, tutup file.
- Mencoba kembali untuk terhubung ke Lingkungan Pengembang dari Toolkit for VS Code.

## Bekerja dengan Amazon API Gateway

Anda dapat menelusuri dan menjalankan sumber daya API Gateway jarak jauh di terhubung AWS menggunakan akun AWS Toolkit for Visual Studio Code.

### Note

Fitur ini tidak mendukung debugging.

Untuk menelusuri dan menjalankan sumber daya API Gateway jarak jauh

1. Di AWS Penjelajah, pilih API Gateway untuk memperluas menu. Sumber daya API Gateway jarak jauh terdaftar.
2. Temukan sumber daya API Gateway yang ingin Anda invokasi, buka menu konteks (klik kanan), lalu pilih Memanggil AWS.
3. Dalam bentuk parameter, tentukan parameter memanggil.

4. Untuk menjalankan sumber daya API Gateway jarak jauh, pilih Aktifkan. Hasilnya dideploy di VS Kode Output lihat.

## Menggunakan AWS App Runner dengan AWS Toolkit for Visual Studio Code

[AWS App Runner](#) menyediakan cara yang cepat, sederhana, dan hemat biaya untuk menyebarkan dari kode sumber atau gambar kontainer langsung ke aplikasi web yang dapat diskalakan dan aman di Cloud. AWS Dengan menggunakannya, Anda tidak perlu mempelajari teknologi baru, memutuskan layanan komputasi mana yang akan digunakan, atau mengetahui cara menyediakan dan mengonfigurasi AWS sumber daya.

Anda dapat menggunakan AWS App Runner untuk membuat dan mengelola layanan berdasarkan gambar sumber atau kode sumber. Jika Anda menggunakan gambar sumber, Anda dapat memilih gambar kontainer publik atau pribadi yang disimpan dalam repositori gambar. App Runner mendukung penyedia repositori gambar berikut:

- Amazon Elastic Container Registry (Amazon ECR): Menyimpan gambar pribadi di akun Anda. AWS
- Amazon Elastic Container Registry Public (Amazon ECR Public): Menyimpan gambar yang dapat dibaca publik.

Jika Anda memilih opsi kode sumber, Anda dapat menerapkan dari repositori kode sumber yang dikelola oleh penyedia repositori yang didukung. Saat ini, App Runner mendukung [GitHub](#) sebagai penyedia repositori kode sumber.

### Prasyarat

Untuk berinteraksi dengan App Runner menggunakan hal AWS Toolkit for Visual Studio Code berikut ini:

- AWS Akun
- Versi fitur AWS Toolkit for Visual Studio Code itu AWS App Runner

Selain persyaratan inti tersebut, pastikan bahwa semua pengguna IAM yang relevan memiliki izin untuk berinteraksi dengan layanan App Runner. Anda juga perlu mendapatkan informasi spesifik

tentang sumber layanan Anda seperti URI gambar kontainer atau koneksi ke GitHub repositori. Anda memerlukan informasi ini saat membuat layanan App Runner.

### Mengkonfigurasi izin IAM untuk App Runner

Cara termudah untuk memberikan izin yang diperlukan untuk App Runner adalah dengan melampirkan kebijakan AWS terkelola yang ada ke entitas yang relevan AWS Identity and Access Management (IAM), khususnya pengguna atau grup. App Runner menyediakan dua kebijakan terkelola yang dapat Anda lampirkan ke pengguna IAM Anda:

- `AWSAppRunnerFullAccess`: Memungkinkan pengguna untuk melakukan semua tindakan App Runner.
- `AWSAppRunnerReadOnlyAccess`: Izinkan pengguna untuk membuat daftar dan melihat detail tentang sumber daya Pelari Aplikasi.

Selain itu, jika Anda memilih repositori pribadi dari Amazon Elastic Container Registry (Amazon Elastic Container Registry) sebagai sumber layanan, Anda harus membuat peran akses berikut untuk layanan App Runner Anda:

- `AWSAppRunnerServicePolicyForECRAccess`: Memungkinkan App Runner mengakses gambar Amazon Elastic Container Registry (Amazon ECR) di akun Anda.

Anda dapat membuat peran ini secara otomatis saat mengonfigurasi instance layanan Anda dengan Palet Perintah VS Code.

#### Note

Peran `AWSAppRunnerServiceRoleForAppRunner` terkait layanan memungkinkan AWS App Runner untuk menyelesaikan tugas-tugas berikut:

- Dorong log ke grup CloudWatch log Amazon Logs.
- Buat aturan Amazon CloudWatch Events untuk berlangganan push image Amazon Elastic Container Registry (Amazon ECR).

Anda tidak perlu membuat peran tertaut layanan secara manual. Saat Anda membuat AWS App Runner di dalam AWS Management Console atau dengan menggunakan operasi API

yang dipanggil oleh AWS Toolkit for Visual Studio Code, AWS App Runner buat peran terkait layanan ini untuk Anda.

Untuk informasi selengkapnya, lihat [Identitas dan manajemen akses untuk Pelari Aplikasi](#) di Panduan AWS App Runner Pengembang.

Memperoleh sumber layanan untuk App Runner

Anda dapat menggunakan AWS App Runner untuk menyebarkan layanan dari gambar sumber atau kode sumber.

Source image

Jika Anda menerapkan dari gambar sumber, Anda dapat memperoleh tautan ke repositori untuk gambar itu dari registri gambar pribadi atau publik AWS .

- [Registri pribadi Amazon ECR: Salin URI untuk repositori pribadi yang menggunakan konsol Amazon ECR di https://console.aws.amazon.com/ecr/repositories.](https://console.aws.amazon.com/ecr/repositories)
- [Registri publik Amazon ECR: Salin URI untuk repositori publik yang menggunakan Galeri Publik Amazon ECR di https://gallery.ecr.aws/.](https://gallery.ecr.aws/)

#### Note

Anda juga dapat memperoleh URI untuk repositori Amazon ECR pribadi langsung dari AWS Explorer di Toolkit for VS Code:

- Buka AWS Explorer dan perluas node ECR untuk melihat daftar repositori untuk Wilayah itu. AWS
- Klik kanan repositori dan pilih Salin Repositori URI untuk menyalin tautan ke clipboard Anda.

Anda menentukan URI untuk repositori gambar saat mengonfigurasi instance layanan Anda dengan Palet Perintah VS Code

Untuk informasi selengkapnya, lihat [Layanan Pelari Aplikasi berdasarkan gambar sumber](#) di Panduan AWS App Runner Pengembang.

## Source code

Agar kode sumber Anda dapat diterapkan ke AWS App Runner layanan, kode tersebut harus disimpan dalam repositori Git yang dikelola oleh penyedia repositori yang didukung. App Runner mendukung satu penyedia repositori kode sumber: [GitHub](#)

Untuk informasi tentang menyiapkan GitHub repositori, lihat [dokumentasi Memulai](#). GitHub

Untuk menerapkan kode sumber Anda ke layanan App Runner dari GitHub repositori, App Runner membuat sambungan ke GitHub. Jika repositori Anda bersifat pribadi (artinya, repositori tidak dapat diakses publik GitHub), Anda harus memberikan detail koneksi kepada App Runner.

### Important

Untuk membuat GitHub koneksi, Anda harus menggunakan konsol App Runner (<https://console.aws.amazon.com/apprunner>) untuk membuat koneksi yang GitHub ditautkan AWS. Anda dapat memilih koneksi yang tersedia di halaman GitHub koneksi saat mengonfigurasi instance layanan Anda dengan Palet Perintah VS Code.

Untuk informasi selengkapnya, lihat [Mengelola koneksi App Runner](#) di Panduan AWS App Runner Pengembang.

Instans layanan App Runner menyediakan runtime terkelola yang memungkinkan kode Anda dibuat dan dijalankan. AWS App Runner saat ini mendukung runtime berikut:

- Runtime terkelola Python
- Node.js mengelola runtime

Sebagai bagian dari konfigurasi layanan, Anda memberikan informasi tentang cara layanan App Runner membangun dan memulai layanan Anda. Anda dapat memasukkan informasi ini menggunakan Command Palette atau menentukan file konfigurasi [App Runner](#) yang diformat YAML. Nilai dalam file ini menginstruksikan App Runner cara membuat dan memulai layanan Anda, serta menyediakan konteks runtime. Ini termasuk pengaturan jaringan yang relevan dan variabel lingkungan. File konfigurasi diberi nama `apprunner.yaml`. Ini secara otomatis ditambahkan ke direktori root repositori aplikasi Anda.

## Harga

Anda dikenakan biaya untuk sumber daya komputasi dan memori yang digunakan aplikasi Anda. Selain itu, jika Anda mengotomatiskan penerapan, Anda juga membayar biaya bulanan yang ditetapkan untuk setiap aplikasi yang mencakup semua penerapan otomatis untuk bulan itu. Jika Anda memilih untuk menerapkan dari kode sumber, Anda juga membayar biaya pembuatan untuk jumlah waktu yang diperlukan App Runner untuk membuat container dari kode sumber Anda.

Untuk informasi selengkapnya, silakan lihat [Harga AWS App Runner](#).

### Topik

- [Membuat layanan App Runner](#)
- [Mengelola layanan App Runner](#)

## Membuat layanan App Runner

Anda dapat membuat layanan App Runner di Toolkit for VS Code dengan menggunakan Palet AWS Perintah Explorer dan VS Code. Setelah Anda memilih untuk membuat layanan di AWS Wilayah tertentu, langkah-langkah bernomor yang disediakan oleh Command Palette memandu Anda melalui proses mengonfigurasi instance layanan tempat aplikasi Anda berjalan.

Sebelum membuat layanan App Runner, pastikan Anda telah menyelesaikan [prasyarat](#). Ini termasuk memberikan izin IAM yang relevan dan mengonfirmasi repositori sumber tertentu yang ingin Anda terapkan.

Untuk membuat layanan App Runner

1. Buka AWS Explorer, jika belum terbuka.
2. Klik kanan node App Runner dan pilih Create Service.

Palet Perintah ditampilkan.

3. Untuk Pilih jenis lokasi kode sumber, pilih ECR atau Repositori.

Jika Anda memilih ECR, Anda menentukan gambar kontainer dalam repositori yang dikelola oleh Amazon Elastic Container Registry. Jika Anda memilih Repositori, Anda menentukan repositori kode sumber yang dikelola oleh penyedia repositori yang didukung. Saat ini, App Runner mendukung [GitHub](#) sebagai penyedia repositori kode sumber.

## Menerapkan dari ECR

1. Untuk Pilih atau masukkan repositori gambar, pilih atau masukkan URL repositori gambar yang dikelola oleh registri pribadi Amazon ECR Anda atau Galeri Publik Amazon ECR.

### Note

Jika Anda menentukan repositori dari Galeri Publik Amazon ECR, pastikan penerapan otomatis dimatikan karena App Runner tidak mendukung penerapan otomatis untuk gambar di repositori Publik ECR.

Penerapan otomatis dimatikan secara default, dan ini ditunjukkan ketika ikon pada header Command Palette menampilkan garis diagonal melaluinya. Jika Anda memilih untuk mengaktifkan penerapan otomatis, sebuah pesan memberi tahu Anda bahwa opsi ini dapat menimbulkan biaya tambahan.

2. Jika langkah Command Palette melaporkan bahwa Tidak ada tag yang ditemukan, Anda harus kembali ke langkah untuk memilih repositori yang berisi gambar kontainer yang ditandai.
3. Jika Anda menggunakan registri pribadi Amazon ECR, Anda memerlukan peran akses ECR, AppRunnerECR AccessRole, yang memungkinkan App Runner mengakses gambar Amazon Elastic Container Registry (Amazon ECR) di akun Anda. Pilih ikon “+” pada header Command Palette untuk membuat peran ini secara otomatis. (Peran akses tidak diperlukan jika gambar Anda disimpan di Amazon ECR Public, tempat gambar tersedia untuk umum.)
4. Untuk Port, masukkan port IP yang digunakan oleh layanan (Port8000, misalnya).
5. Untuk Konfigurasi variabel lingkungan, Anda dapat menentukan file yang berisi variabel lingkungan yang digunakan untuk menyesuaikan perilaku dalam instance layanan Anda. Atau Anda dapat melewati langkah ini.
6. Untuk Nama layanan Anda, masukkan nama unik tanpa spasi dan tekan Enter.
7. Untuk konfigurasi instans Pilih, pilih kombinasi unit CPU dan memori dalam GB untuk instance layanan Anda.

Ketika layanan Anda sedang dibuat, statusnya berubah dari Creating to Running.

8. Setelah layanan Anda mulai berjalan, klik kanan dan pilih Salin URL Layanan.
9. Untuk mengakses aplikasi yang Anda gunakan, rekatkan URL yang disalin ke bilah alamat browser web Anda.

## Menerapkan dari repositori jarak jauh

1. Untuk Pilih koneksi, pilih koneksi yang terhubung GitHub ke AWS. Koneksi yang tersedia untuk dipilih tercantum di halaman GitHub koneksi di konsol App Runner.
2. Untuk Pilih GitHub repositori jarak jauh, pilih atau masukkan URL untuk repositori jarak jauh.

Repositori jarak jauh yang sudah dikonfigurasi dengan manajemen kontrol sumber (SCM) Visual Studio Code tersedia untuk dipilih. Anda juga dapat menempelkan tautan ke repositori jika tidak terdaftar.

3. Untuk Pilih cabang, pilih cabang Git mana dari kode sumber Anda yang ingin Anda gunakan.
4. Untuk Pilih sumber konfigurasi, tentukan bagaimana Anda ingin menentukan konfigurasi runtime Anda.

Jika Anda memilih Gunakan file konfigurasi, instance layanan Anda dikonfigurasi oleh pengaturan yang ditentukan oleh file `apprunner.yaml` konfigurasi. File ini ada di direktori root repositori aplikasi Anda.

Jika Anda memilih Konfigurasi semua pengaturan di sini, gunakan palet Command untuk menentukan hal berikut:

- Runtime: Pilih Python 3 atau Nodejs 12.
  - Perintah build: Masukkan perintah untuk membangun aplikasi Anda di lingkungan runtime instance layanan Anda.
  - Mulai perintah: Masukkan perintah untuk memulai aplikasi Anda di lingkungan runtime instance layanan Anda.
5. Untuk Port, masukkan port IP yang digunakan oleh layanan (Port8000, misalnya).
  6. Untuk Konfigurasi variabel lingkungan, Anda dapat menentukan file yang berisi variabel lingkungan yang digunakan untuk menyesuaikan perilaku dalam instance layanan Anda. Atau Anda dapat melewati langkah ini.
  7. Untuk Nama layanan Anda, masukkan nama unik tanpa spasi dan tekan Enter.
  8. Untuk konfigurasi instans Pilih, pilih kombinasi unit CPU dan memori dalam GB untuk instance layanan Anda.

Ketika layanan Anda sedang dibuat, statusnya berubah dari Creating to Running.

9. Setelah layanan Anda mulai berjalan, klik kanan dan pilih Salin URL Layanan.

10. Untuk mengakses aplikasi yang Anda gunakan, rekatkan URL yang disalin ke bilah alamat browser web Anda.

#### Note

Jika upaya Anda untuk membuat layanan App Runner gagal, layanan akan menampilkan status Buat gagal di AWS Explorer. Untuk tips pemecahan masalah, lihat [Saat pembuatan layanan gagal](#) di Panduan Pengembang App Runner.

## Mengelola layanan App Runner

Setelah membuat layanan App Runner, Anda dapat mengelolanya dengan menggunakan panel AWS Explorer untuk melakukan aktivitas berikut:

- [Menjeda dan melanjutkan layanan App Runner](#)
- [Menyebarkan layanan App Runner](#)
- [Melihat aliran log untuk App Runner](#)
- [Menghapus layanan App Runner](#)

### Menjeda dan melanjutkan layanan App Runner

Jika Anda perlu menonaktifkan aplikasi web untuk sementara dan menghentikan kode agar tidak berjalan, Anda dapat menjeda layanan AWS App Runner. App Runner mengurangi kapasitas komputasi untuk layanan menjadi nol. Ketika Anda siap untuk menjalankan aplikasi Anda lagi, lanjutkan layanan App Runner Anda. App Runner menyediakan kapasitas komputasi baru, menyebarkan aplikasi Anda ke dalamnya, dan menjalankan aplikasi.

#### Important

Anda ditagih untuk App Runner hanya saat sedang berjalan. Oleh karena itu, Anda dapat menjeda dan melanjutkan aplikasi Anda sesuai kebutuhan untuk mengelola biaya. Ini sangat membantu dalam skenario pengembangan dan pengujian.

## Untuk menjeda layanan App Runner

1. Buka AWS Explorer, jika belum terbuka.
2. Perluas App Runner untuk melihat daftar layanan.
3. Klik kanan layanan Anda dan pilih Jeda.
4. Di kotak dialog yang ditampilkan, pilih Konfirmasi.

Saat layanan berhenti, status layanan berubah dari Running ke Pusing dan kemudian ke Pause.

## Untuk melanjutkan layanan App Runner

1. Buka AWS Explorer, jika belum terbuka.
2. Perluas App Runner untuk melihat daftar layanan.
3. Klik kanan layanan Anda dan pilih Lanjutkan.

Saat layanan dilanjutkan, status layanan berubah dari Melanjutkan ke Berjalan.

## Menyebarkan layanan App Runner

Jika Anda memilih opsi penerapan manual untuk layanan Anda, Anda perlu secara eksplisit memulai setiap penerapan ke layanan Anda.

1. Buka AWS Explorer, jika belum terbuka.
2. Perluas App Runner untuk melihat daftar layanan.
3. Klik kanan layanan Anda dan pilih Mulai Deployment.
4. Saat aplikasi Anda sedang di-deploy, status layanan berubah dari Deploying ke Running.
5. Untuk mengonfirmasi bahwa aplikasi Anda berhasil digunakan, klik kanan layanan yang sama dan pilih Salin URL Layanan.
6. Untuk mengakses aplikasi web yang Anda gunakan, tempel URL yang disalin ke bilah alamat browser web Anda.

## Melihat aliran log untuk App Runner

Gunakan CloudWatch Log untuk memantau, menyimpan, dan mengakses aliran log Anda untuk layanan seperti App Runner. Pengaliran log adalah urutan log acara yang berbagi sumber yang sama.

1. Perluas App Runner untuk melihat daftar instance layanan.
2. Perluas instance layanan tertentu untuk melihat daftar grup log. (Grup log adalah grup aliran log yang berbagi pengaturan retensi, pemantauan, dan kontrol akses yang sama.)
3. Klik kanan grup log dan pilih Lihat Aliran Log.
4. Dari Command Palette, pilih aliran log dari grup.

Editor VS Code menampilkan daftar peristiwa log yang membentuk aliran. Anda dapat memilih untuk memuat acara lama atau yang lebih baru ke editor.

## Menghapus layanan App Runner

### Important

Jika Anda menghapus layanan App Runner, layanan tersebut akan dihapus secara permanen dan data yang disimpan akan dihapus. Jika Anda perlu membuat ulang layanan, App Runner perlu mengambil sumber Anda lagi dan membangunnya jika itu adalah repositori kode. Aplikasi web Anda mendapatkan domain App Runner baru.

1. Buka AWS Explorer, jika belum terbuka.
2. Perluas App Runner untuk melihat daftar layanan.
3. Klik kanan layanan dan pilih Hapus Layanan.
4. Di Palet Perintah, masukkan hapus dan kemudian tekan Enter untuk mengonfirmasi.

Layanan yang dihapus menampilkan status Menghapus, dan kemudian layanan menghilang dari daftar.

## AWSKomposer Aplikasi

Anda dapat menggunakan AWS Toolkit for Visual Studio Code untuk bekerja dengan layanan AWS Application Composer. AWS Application Composer adalah pembangun visual untuk AWS aplikasi yang membantu merancang arsitektur aplikasi Anda dan memvisualisasikan infrastruktur Anda. AWS CloudFormation

Untuk informasi rinci tentang layanan AWS Application Composer, lihat Panduan Pengguna [AWSApplication Composer](#).

Topik berikut menjelaskan cara bekerja dengan AWS Application Composer dari AWS Toolkit for Visual Studio Code

Topik

- [Bekerja dengan AWS Application Composer dari Toolkit](#)

## Bekerja dengan AWS Application Composer dari Toolkit

AWS Aplikasi Komposer untuk AWS Toolkit for Visual Studio Code memungkinkan Anda untuk merancang aplikasi secara visual melalui kanvas interaktif. Anda juga dapat menggunakan Application Composer untuk memvisualisasikan dan memodifikasi AWS CloudFormation dan AWS Serverless Application Model (AWS SAM) template. Saat bekerja dengan Application Composer, perubahan Anda disimpan terus-menerus memungkinkan Anda untuk beralih mulus antara mengedit file secara langsung di editor VS Code atau menggunakan kanvas interaktif.

Untuk informasi rinci tentang layanan AWS Application Composer, memulai informasi, dan tutorial, lihat Panduan Pengguna [layanan Komposer AWS Aplikasi](#).

Bagian berikut menjelaskan cara mengakses layanan AWS Application Composer dari AWS Toolkit for Visual Studio Code

### Mengakses AWS Application Composer dari Toolkit

Ada 3 cara utama Anda dapat mengakses AWS Application Composer dari Toolkit.

Mengakses AWS Application Composer dari template yang ada

1. Dari VS Code, buka file template yang ada di editor VS Code.
2. Dari jendela editor, klik tombol AWS Application Composer yang terletak di sudut kanan atas jendela editor.
3. AWS Application Composer membuka dan memvisualisasikan file template Anda di jendela editor VS Code.

Mengakses AWS Application Composer dari menu konteks (klik kanan)

1. Dari VS Code klik kanan file template yang ingin Anda buka dengan AWS Application Composer.
2. Di menu konteks, pilih opsi Buka dengan App Composer.

3. AWS Application Composer membuka dan memvisualisasikan file template Anda di jendela editor VS Code baru.

### Mengakses AWS Application Composer dari Command Palette

1. Dari VS Code buka Command Palette dengan menekan **Cmd + Shift + P** atau **Ctrl + Shift + P** (Windows)
2. Di bidang pencarian, masukkan **AWS Application Composer** dan pilih AWS Application Composer saat mengisi hasil.
3. Pilih file template yang ingin Anda buka, AWS Application Composer membuka dan memvisualisasikan file template Anda di jendela editor VS Code baru.

## AWS CDK untuk VS Code

Ini adalah dokumentasi prarilis untuk fitur dalam rilis pratinjau. Hal ini dapat berubah.

Parameter AWS CDK layanan memungkinkan Anda untuk bekerja dengan [AWS Cloud Development Kit \(AWS CDK\)](#) aplikasi, atau aplikasi. Anda dapat menemukan informasi rinci tentang AWS CDK di dalam [AWS Cloud Development Kit \(AWS CDK\) Panduan Pengembang](#).

AWS CDK aplikasi terdiri dari blok bangunan yang dikenal sebagai [membangun](#), yang mencakup definisi untuk AWS CloudFormation tumpukan dan AWS sumber daya di dalamnya. Menggunakan AWS CDK Penjelajah, Anda dapat memvisualisasikan [tumpukan](#) dan [sumber daya](#) yang didefinisikan dalam AWS CDK konstruksi. Visualisasi ini disediakan dalam tampilan pohon di panel Developer Tools dalam editor Visual Studio Code (VS Code).

Bagian ini menyediakan informasi tentang cara mengakses dan menggunakan AWS CDK di editor Kode VS. Ini mengasumsikan bahwa Anda sudah [diinstal dan konfigurasi](#) Toolkit for VS Code untuk IDE lokal Anda.

### Topik

- [Bekerja dengan AWS CDK aplikasi](#)

## Bekerja dengan AWS CDK aplikasi

Ini adalah dokumentasi prarilis untuk fitur dalam rilis pratinjau. Dokumentasi dapat berubah.

Gunakan AWS CDK Explorer di AWS Toolkit for VS Code untuk memvisualisasikan dan bekerja AWS CDK dengan aplikasi.

### Prasyarat

- Pastikan sistem Anda memenuhi prasyarat yang ditentukan dalam [Menginstal Toolkit for VS Code](#).
- Instal antarmuka baris AWS CDK perintah, seperti yang dijelaskan di beberapa bagian pertama [Memulai dengan](#) Panduan AWS Cloud Development Kit (AWS CDK) Pengembang. AWS CDK

#### Important

AWS CDK Versi harus 1.17.0 atau yang lebih baru. Gunakan **cdk --version** pada baris perintah untuk melihat versi apa yang Anda jalankan.

### Visualisasikan aplikasi AWS CDK

[Menggunakan AWS Toolkit for VS AWS CDK Code Explorer, Anda dapat mengelola tumpukan dan sumber daya yang disimpan dalam konstruksi CDK aplikasi Anda.](#) AWS CDK Explorer menampilkan sumber daya Anda dalam tampilan pohon menggunakan informasi yang ditentukan dalam `tree.json` file, yang dibuat saat Anda menjalankan **cdk synth** perintah. `tree.json` file ini terletak di `cdk.out` direktori aplikasi, secara default.

Untuk mulai menggunakan Toolkit AWS CDK Explorer, Anda harus membuat CDK aplikasi.

1. Selesaikan beberapa langkah pertama dari [Hello World Tutorial](#) yang terletak di [Panduan AWS CDK Pengembang](#).

#### Important

Ketika Anda tiba di langkah tutorial Menyebarkan Stack, berhenti dan kembali ke panduan ini.

**Note**

Anda dapat menjalankan perintah yang disediakan dalam tutorial, misalnya, **mkdir** dan **cdk init**, pada baris perintah sistem operasi atau di jendela Terminal di dalam editor Kode VS.

2. Setelah Anda menyelesaikan langkah-langkah CDK tutorial yang diperlukan, buka CDK konten yang Anda buat di editor VS Code.
3. Dari panel AWS navigasi, perluas judul CDK(Pratinjau). CDKAplikasi Anda dan sumber daya yang terkait sekarang ditampilkan di tampilan pohon CDK Explorer.

**Catatan penting**

- Saat memuat CDK aplikasi ke editor VS Code, Anda dapat memuat beberapa folder sekaligus. Setiap folder dapat berisi beberapa CDK aplikasi, seperti yang ditunjukkan pada gambar sebelumnya. AWS CDK Explorer menemukan aplikasi di direktori root proyek dan subdirektori langsungnya.
- Ketika Anda melakukan beberapa langkah pertama dari tutorial, Anda mungkin memperhatikan bahwa perintah terakhir yang Anda jalankan adalah **cdk synth**, yang menghasilkan `tree.json` file. Jika Anda mengubah aspek CDK aplikasi, misalnya, menambahkan lebih banyak sumber daya, Anda perlu menjalankan perintah itu lagi untuk melihat perubahan yang tercermin dalam tampilan pohon.

**Melakukan operasi lain di AWS CDK aplikasi**

Anda dapat menggunakan editor Kode VS untuk melakukan operasi lain pada CDK aplikasi, seperti yang Anda lakukan dengan menggunakan baris perintah sistem operasi Anda atau alat lainnya. Misalnya, Anda dapat memperbarui file kode di editor dan menerapkan aplikasi dengan menggunakan jendela VS Code Terminal.

Untuk mencoba jenis tindakan ini, gunakan editor VS Code untuk melanjutkan [Hello World Tutorial](#) di Panduan AWS CDK Pengembang. Pastikan untuk melakukan langkah terakhir, Menghancurkan Sumber Daya Aplikasi, sehingga Anda tidak dikenakan biaya tak terduga ke akun Anda AWS .

# Bekerja dengan AWS CloudFormation tumpukan

Parameter AWS Toolkit for Visual Studio Code menyediakan dukungan untuk [AWS CloudFormation](#) tumpukan. Menggunakan Toolkit for VS Code, Anda dapat melakukan tugas-tugas tertentu dengan AWS CloudFormation tumpukan, seperti menghapusnya.

## Topik

- [Menghapus AWS CloudFormation tumpukan](#)
- [Buat AWS CloudFormation template menggunakan AWS Toolkit for Visual Studio Code](#)

## Menghapus AWS CloudFormation tumpukan

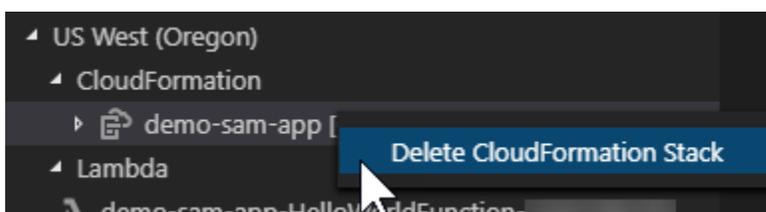
Anda dapat menggunakan AWS Toolkit for Visual Studio Code untuk menghapus AWS CloudFormation tumpukan.

## Prasyarat

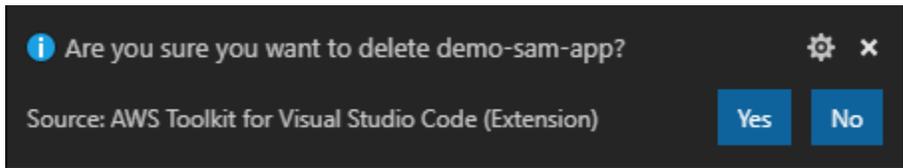
- Pastikan sistem Anda memenuhi prasyarat yang ditentukan dalam [Instalasi Toolkit for VS Code](#).
- Pastikan kredensi yang Anda konfigurasi termasuk [Otentikasi dan akses](#) termasuk akses baca/tulis yang sesuai ke AWS CloudFormation layanan. Jika di AWS Explorer, di bawah CloudFormation, Anda melihat pesan yang mirip dengan “Kesalahan memuat sumber daya CloudFormation”, periksa izin yang dilampirkan pada kredensial tersebut. Perubahan yang Anda buat pada izin akan memakan waktu beberapa menit untuk memengaruhi AWS Penjelajah di VS Code.

## Hapus tumpukan CloudFormation

1. Di AWS Explorer, buka menu konteks tumpukan AWS CloudFormation yang ingin Anda hapus.



2. Pilih Hapus Tumpukan CloudFormation.
3. Pada pesan yang muncul, pilih Ya untuk mengonfirmasi penghapusan.



Setelah tumpukan dihapus, itu tidak lagi tercantum di AWS Explorer.

## Buat AWS CloudFormation template menggunakan AWS Toolkit for Visual Studio Code

AWS Toolkit for Visual Studio Code Dapat membantu Anda dalam menulis AWS CloudFormation dan template SAM.

### Prasyarat

Toolkit for VS Code dan prasyarat kredensial

- Sebelum Anda dapat mengakses CloudFormation layanan dari Toolkit for VS Code, Anda harus memenuhi persyaratan yang diuraikan dalam [panduan pengguna Menginstal Toolkit for VS Code](#).
- Kredensial yang Anda buat [Otentikasi dan akses](#) harus menyertakan akses baca/tulis yang sesuai ke layanan. AWS CloudFormation

#### Note

Jika CloudFormation layanan menampilkan pesan Error loading CloudFormation resources, periksa izin yang Anda lampirkan ke kredensial tersebut. Perhatikan juga bahwa Perubahan yang dilakukan pada izin mungkin memerlukan beberapa menit untuk diperbarui di AWS Explorer.

### CloudFormation prasyarat template

- Instal dan aktifkan ekstensi [Redhat Developer YAMM VS Code](#).
- Anda harus terhubung ke internet saat menggunakan ekstensi Redhat Developer YAMM VS Code karena digunakan untuk mengunduh dan mengunggah skema JSON di komputer Anda.

## Menulis CloudFormation template dengan YAMM Schema Support

Toolkit menggunakan dukungan bahasa YAMAL dan skema JSON untuk merampingkan proses penulisan dan template SAM. CloudFormation Fitur seperti validasi sintaks dan pelengkapan otomatis tidak hanya membuat proses lebih cepat, tetapi juga membantu meningkatkan kualitas template Anda. Saat memilih skema untuk templat Anda, berikut ini adalah praktik terbaik yang disarankan.

### CloudFormation Template

- File memiliki ekstensi `.yaml` atau `.yml`.
- File ini memiliki node tingkat atas `AWSTemplateFormatVersion` atau Sumber Daya.

### Templat SAM

- Semua kriteria yang telah dijelaskan untuk CloudFormation
- File ini memiliki node Transform tingkat atas, yang berisi nilai yang dimulai dengan `AWS::Serverless`.

Skema akan diterapkan pada modifikasi file. Misalnya, skema Template SAM akan diterapkan setelah menambahkan transformasi tanpa server ke CloudFormation template dan menyimpan file.

### Validasi Sintaks

Ekstensi YAMM akan secara otomatis menerapkan validasi tipe ke template Anda. Ini menyoroti entri dengan tipe yang tidak valid untuk properti tertentu. Jika Anda mengarahkan kursor ke entri yang disorot, ekstensi akan menampilkan tindakan korektif.

### Pelengkapan Otomatis

Saat menambahkan bidang baru, nilai yang disebutkan, atau [jenis sumber daya](#) lainnya, Anda dapat memulai fitur pelengkapan otomatis ekstensi YAMM dengan mengetikkan ruang Ctrl +.

## Bekerja dengan CloudWatchLog dengan menggunakan AWS Toolkit for Visual Studio Code

Amazon CloudWatchLog memungkinkan Anda untuk memusatkan log dari semua sistem, aplikasi, dan AWS layanan yang Anda gunakan, dalam satu layanan yang sangat skalabel. Anda kemudian dapat dengan mudah melihat log, mencari adanya kode atau pola

kesalahan tertentu, memfilter berdasarkan bidang tertentu, atau mengarsipkannya dengan aman untuk analisis pada masa mendatang. Untuk informasi lebih lanjut, lihat [Apa itu AmazonCloudWatchLog?](#) di AmazonCloudWatchPanduan Pengguna.

Topik berikut menjelaskan cara menggunakan AWS Toolkit for Visual Studio Code untuk bekerja dengan CloudWatchLog dalam AWS akun.

Topik

- [Melihat grup log CloudWatch dan aliran log menggunakan AWS Toolkit for Visual Studio Code](#)
- [Bekerja dengan CloudWatch peristiwa log di aliran log dengan menggunakan AWS Toolkit for Visual Studio Code](#)
- [Mencari grup CloudWatch log](#)

## Melihat grup log CloudWatch dan aliran log menggunakan AWS Toolkit for Visual Studio Code

Pengaliran log adalah urutan log acara yang berbagi sumber yang sama. Setiap sumber log terpisah ke dalam CloudWatch Logs membentuk pengaliran log terpisah.

Grup log adalah grup pengaliran log yang berbagi pengaturan retensi, pemantauan, dan kontrol akses yang sama. Anda dapat menentukan grup log dan menentukan pengaliran untuk dimasukkan ke dalam setiap grup. Tidak ada batas jumlah pengaliran log yang dapat tergabung dalam satu grup log.

Untuk informasi selengkapnya, lihat [Bekerja dengan Grup Log dan Pengaliran Log](#) dalam Panduan pengguna Amazon CloudWatch.

Topik

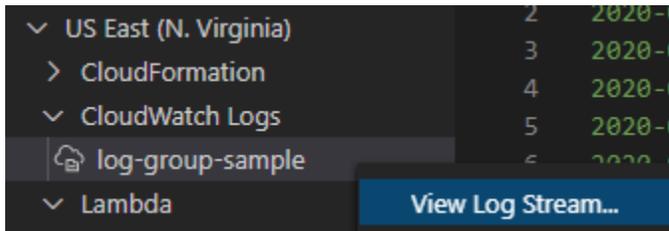
- [Melihat grup log dan aliran log dengan simpul CloudWatch Logs](#)

## Melihat grup log dan aliran log dengan simpul CloudWatch Logs

1. Dalam Kode VS, pilih **Lihat**, **Penjelajah** membuka **AWSExplorer**.
2. Klik simpul **CloudWatch Logs** untuk memperluas daftar grup log.

Grup log untuk Wilayah AWS saat ini ditampilkan di bawah simpul **CloudWatch Logs**.

3. Untuk melihat aliran log di grup log, klik kanan nama grup log, dan kemudian pilih **Streaming Log**.



4. Dari Palet Perintah, pilih aliran log dari grup untuk melihat.

#### Note

Parameter Palet Perintah menampilkan timestamp untuk acara terakhir di setiap aliran.

Parameter [Streaming Log editor](#) meluncurkan untuk menampilkan peristiwa log stream.

## Bekerja dengan CloudWatch peristiwa log di aliran log dengan menggunakan AWS Toolkit for Visual Studio Code

Setelah Anda membuka Log uapeditor, Anda dapat mengakses peristiwa log di setiap aliran. Peristiwa log adalah catatan aktivitas yang direkam oleh aplikasi atau sumber daya yang dipantau.

### Topik

- [Melihat dan menyalin informasi aliran log](#)
- [Menyimpan konten dari editor aliran log ke file lokal](#)

### Melihat dan menyalin informasi aliran log

Saat Anda membuka aliran log, Streaming Log editor menampilkan urutan aliran peristiwa log tersebut.

1. Untuk menemukan aliran log untuk dilihat, buka Streaming Log editor (lihat [Melihat grup log CloudWatch dan aliran log](#)).

Setiap baris yang mencantumkan suatu peristiwa diberi stempel waktu untuk menunjukkan kapan peristiwa itu dicatat.

2. Anda dapat melihat dan menyalin informasi tentang peristiwa aliran menggunakan opsi berikut:
  - Lihat acara berdasarkan waktu: Menampilkan peristiwa log terbaru dan lama dengan memilih Muat peristiwa yang lebih baru atau Muat acara yang lebih lama.

**Note**

Editor Aliran Log awalnya memuat batch paling baru 10.000 baris peristiwa log atau 1 MB data log (mana yang lebih kecil). Jika Anda memilih Muat peristiwa yang lebih baru, editor menampilkan peristiwa yang dicatat setelah batch terakhir dimuat. Jika Anda memilih Muat peristiwa yang lebih lama, editor menampilkan batch peristiwa yang terjadi sebelum peristiwa yang saat ini ditampilkan.

- Salin peristiwa log: Pilih peristiwa yang akan disalin, lalu klik kanan dan pilih Salin dari menu.
- Salin nama aliran log: Klik kanan tab Streaming Log editor dan pilih Salin Nama Aliran Log.

**Note**

Anda juga dapat menggunakan Palet Perintah run AWS Toolkit Salin Nama Aliran Log.

## Menyimpan konten dari editor aliran log ke file lokal

Anda dapat mengunduh konten CloudWatch log editor aliran ke logfile pada mesin lokal Anda.

**Note**

Dengan opsi ini, Anda menyimpan ke file hanya peristiwa log yang saat ini ditampilkan di editor aliran log. Sebagai contoh, jika ukuran total aliran log adalah 5 MB dan hanya 2 MB yang dimuat di editor, file yang Anda simpan juga akan berisi hanya 2 MB data log. Untuk menampilkan lebih banyak data yang akan disimpan, pilih Muat peristiwa yang lebih baru atau Muat peristiwa yang lebih lama dalam editor.

1. Untuk menemukan aliran log yang akan disalin, buka Streaming Log editor (lihat [Melihat grup log CloudWatch dan aliran log](#)).
2. Pilih Simpan ikon di samping tab yang menampilkan nama aliran log.

**Note**

Anda juga dapat menggunakan Palet Perintah `runAWSToolkit Simpan Konten Aliran Log Saat Ini`.

- Gunakan kotak dialog untuk memilih atau membuat folder unduhan untuk file log, dan klik `Simpan`.

## Mencari grup CloudWatch log

Anda dapat menggunakan Grup Log Pencarian untuk mencari semua aliran log dalam grup log.

Untuk informasi terperinci tentang layanan Amazon CloudWatch Logs, lihat topik [Bekerja dengan Grup Log dan Aliran Log](#) di Panduan CloudWatch Pengguna Amazon.

### Mencari grup log dari VS Code Command Palette

Untuk mencari grup log dari VS Code Command Palette, selesaikan langkah-langkah berikut.

Untuk informasi mendetail tentang filter dan pola Amazon CloudWatch Logs, lihat bagian [Filter dan sintaks pola](#) di Panduan CloudWatch Pengguna Amazon.

- Dari VS Code, buka Command Palette dengan menekan **cmd+shift+p** (windows:**ctrl+shift+p**).
- Dari Command Palette **AWS: Search Log Group**, masukkan perintah, lalu pilih untuk membuka dialog grup log pencarian di VS Code dan ikuti petunjuk untuk melanjutkan.

**Note**

Dari prompt pertama, Anda memiliki opsi untuk beralih AWS wilayah Anda sebelum melanjutkan ke langkah berikutnya.

- Dari prompt `Pilih Grup Log (1/3)`, pilih grup log yang ingin Anda cari.
- Dari prompt `Select Time Filter (2/3)`, pilih filter waktu yang akan diterapkan pada pencarian Anda.
- Dari prompt `Grup Log Pencarian... (3/3)`, masukkan pola pencarian Anda di bidang yang disediakan, lalu tekan **Enter** tombol untuk melanjutkan atau **ESC** tombol untuk membatalkan pencarian.

6. Hasil pencarian Anda terbuka di editor VS Code, saat pencarian selesai.

## Mencari grup log dari AWS Explorer

Untuk mencari grup log dari AWS Toolkit for Visual Studio Code Explorer, selesaikan langkah-langkah berikut.

1. Dari AWS Toolkit for Visual Studio Code Explorer, perluas CloudWatch.
2. Buka menu konteks untuk (klik kanan) Grup Log Pencarian yang ingin Anda cari, lalu pilih Grup Log Pencarian untuk membuka prompt pencarian.
3. Ikuti petunjuknya dengan memilih kerangka waktu untuk melanjutkan.
4. Saat diminta, masukkan pola pencarian Anda di bidang yang disediakan, lalu tekan **Enter** tombol untuk melanjutkan atau **ESC** tombol untuk membatalkan pencarian.
5. Hasil pencarian Anda terbuka di editor VS Code saat pencarian selesai.

## Bekerja dengan hasil log pencarian

Setelah menyelesaikan pencarian grup CloudWatch log yang berhasil, hasil pencarian Anda terbuka di editor Kode VS. Prosedur berikut menjelaskan cara bekerja dengan hasil log pencarian.

### Note

Saat melihat aliran log tunggal, fitur berikut terbatas pada hasil di aliran log Anda yang aktif saat ini.

## Menyimpan hasil grup log pencarian

Untuk menyimpan hasil grup log pencarian Anda secara lokal, selesaikan langkah-langkah berikut.

1. Dari hasil grup log pencarian Anda, pilih tombol ikon Simpan Log ke File, yang terletak di sudut kanan atas editor Kode VS.
2. Dari prompt Save As, tentukan nama dan lokasi tempat Anda ingin menyimpan file tersebut.
3. Memilih OK menyimpan file ke mesin lokal Anda.

## Mengubah rentang waktu rentang waktu

Untuk mengubah rentang waktu yang aktif dalam hasil grup log pencarian Anda, selesaikan langkah-langkah berikut.

1. Dari hasil grup log pencarian Anda, pilih Cari berdasarkan tanggal... tombol ikon, terletak di sudut kanan atas editor VS Code.
2. Dari prompt Select Time Filter, pilih rentang waktu baru untuk hasil log pencarian Anda.
3. Hasil Anda diperbarui saat prompt Select Time Filter ditutup.

## Mengubah pola pencarian

Untuk mengubah pola pencarian yang aktif di hasil grup log pencarian Anda, selesaikan langkah-langkah berikut.

1. Dari hasil grup log pencarian Anda, pilih Cari berdasarkan Pola... tombol ikon, terletak di sudut kanan atas editor VS Code.
2. Dari prompt Grup Log Pencarian, masukkan pola pencarian baru di bidang yang disediakan.
3. Tekan **Enter** tombol untuk menutup prompt dan memperbarui hasil Anda dengan pola pencarian baru.

# Bekerja dengan Amazon Elastic Container Registry

Amazon Elastic Container Registry (Amazon ECR) adalah layanan registri kontainer AWS terkelola yang aman dan dapat diskalakan. Beberapa fungsi layanan Amazon ECR dapat diakses dari Toolkit for VS Code Explorer.

- Membuat repositori.
- Membuat AWS App Runner layanan untuk repositori Anda atau gambar yang ditandai.
- Mengakses tag gambar dan URI repositori atau ARN.
- Menghapus tag gambar dan repositori.

Anda juga dapat mengakses berbagai fungsi Amazon ECR melalui konsol VS Code dengan mengintegrasikan AWS CLI dan platform lainnya, dengan VS Code.

Untuk informasi selengkapnya tentang Amazon ECR, lihat [Apa itu Amazon ECR?](#) dalam Panduan Pengguna Amazon Elastic Container Registry.

## Prasyarat

Anda harus menyelesaikan langkah-langkah ini untuk mengakses layanan Amazon ECR dari VS Code Explorer.

### Membuat pengguna IAM

Sebelum Anda dapat mengakses AWS layanan, seperti Amazon ECR, Anda harus memberikan kredensi. Ini adalah agar layanan dapat menentukan apakah Anda memiliki izin untuk mengakses sumber dayanya. Kami tidak menyarankan Anda untuk mengakses AWS langsung melalui kredensial untuk AWS akun root Anda. Sebagai gantinya, gunakan AWS Identity and Access Management (IAM) untuk membuat pengguna IAM dan kemudian tambahkan pengguna tersebut ke grup IAM dengan izin administratif. Anda kemudian dapat mengakses AWS dengan URL khusus dan kredensial untuk pengguna IAM.

Jika Anda mendaftar ke AWS tetapi tidak membuat pengguna IAM untuk Anda sendiri, Anda dapat membuatnya menggunakan konsol IAM.

Pilih salah satu opsi berikut untuk membuat pengguna administrator.

Pilih salah satu cara untuk mengelola administrator	Ke	oleh	Anda juga dapat
Di IAM Identity Center  (Disarankan)	Gunakan kredensial jangka pendek untuk mengakses AWS.  Ini sejalan dengan praktik terbaik	Mengikuti petunjuk dalam <a href="#">Memulai</a> di Panduan AWS IAM Identity Center Pengguna.	Konfigurasi akses terprogram dengan <a href="#">Mengkonfigurasi AWS CLI untuk digunakan AWS IAM Identity Center</a> di AWS Command Line Interface User Guide.

Pilih salah satu cara untuk mengelola administrator	Ke	oleh	Anda juga dapat
	keamanan. Untuk informasi tentang praktik terbaik, lihat <a href="#">praktik terbaik keamanan di IAM</a> dalam Panduan Pengguna IAM.		
Di IAM (Tidak disarankan)	Gunakan kredensial jangka panjang untuk mengakses AWS.	Ikuti instruksi dalam <a href="#">Membuat grup pengguna dan pengguna IAM pertama Anda</a> dalam Panduan Pengguna IAM.	Konfigurasi akses terprogram dengan <a href="#">Mengelola kunci akses untuk pengguna</a> IAM di Panduan Pengguna IAM.

Untuk masuk sebagai pengguna IAM baru ini, keluar dari AWS konsol, kemudian gunakan URL berikut. Di URL berikut, di mana `your_aws_account_id` adalah nomor Anda AWS adalah nomor Anda AWS1234-5678-9012, ID AWS akun Anda adalah 123456789012):

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

Masukkan nama pengguna dan kata sandi IAM yang baru saja Anda buat. Saat Anda masuk, bilah navigasi menampilkan "nama\_pengguna\_Anda @ id\_akun\_aws\_Anda".

Jika Anda tidak ingin URL tersebut agar halaman masuk Anda berisi ID akun AWS Anda, Anda dapat membuat alias akun. Dari dasbor IAM, pilih Sesuaikan dan masukkan Alias Akun. Ini bisa menjadi nama perusahaan Anda. Untuk informasi selengkapnya, lihat [ID AWS akun Anda dan aliasnya](#) di Panduan Pengguna IAM.

Untuk masuk setelah membuat alias akun, gunakan URL berikut:

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

Untuk memverifikasi tautan masuk untuk pengguna IAM untuk akun Anda, buka konsol IAM dan periksa di Tautan masuk pengguna IAM di dasbor.

Untuk informasi lebih lanjut tentang IAM, lihat [Panduan Pengguna AWS Identity and Access Management](#).

## Instal dan konfigurasi Docker

Anda dapat menginstal dan mengkonfigurasi Docker dengan memilih sistem operasi pilihan Anda dari panduan pengguna [Install Docker Engine](#) dan mengikuti instruksi.

## Instal dan konfigurasi AWS CLI versi 2

Instal dan konfigurasi AWS CLI versi 2 dengan memilih sistem operasi pilihan Anda dari panduan pengguna [Installing, update, dan uninstall AWS CLI versi 2](#).

## Topik

- [Bekerja dengan layanan Amazon Elastic Container Registry dalam VS Code](#)

# Bekerja dengan layanan Amazon Elastic Container Registry dalam VS Code

Anda dapat mengakses layanan Amazon Elastic Container Registry (Amazon ECR) langsung dari AWS Explorer dalam VS Code dan gunakan untuk mendorong gambar program ke repositori Amazon ECR. Untuk memulai, Anda perlu melakukan langkah-langkah ini:

1. Buat Dockerfile yang berisi informasi yang diperlukan untuk membangun gambar.
2. Buat gambar dari Dockerfile tersebut dan beri tag pada gambar untuk diproses.
3. Buat repositori di dalam instans Amazon ECR Anda.
4. Dorong citra yang ditandai ke repositori Anda.

## Bagian

- [Prasyarat](#)
- [1. Membuat Dockerfile](#)

- [2. Membangun citra Anda dari Dockerfile Anda](#)
- [3. Membuat repositori baru](#)
- [4. Dorong, tarik, dan hapus gambar](#)

## Prasyarat

Sebelum Anda dapat menggunakan fitur layanan Amazon ECR dari Toolkit for VS Code, Anda harus memenuhi fitur ini [prasyarat](#).

### 1. Membuat Dockerfile

Docker menggunakan file bernama Dockerfile untuk menentukan gambar yang dapat didorong dan disimpan pada repositori jarak jauh. Sebelum Anda dapat mengunggah gambar ke repositori ECR, Anda harus membuat Dockerfile dan kemudian membuat gambar dari Dockerfile tersebut.

#### Membuat Dockerfile

1. Gunakan Toolkit for VS Code explorer untuk menavigasi ke direktori tempat Anda ingin menyimpan Dockerfile Anda.
2. Membuat file baru yang dipanggil Dockerfile.

#### Note

VS Code dapat meminta Anda untuk memilih jenis file atau ekstensi file. Jika ini terjadi, pilih teks biasa. VS Code memiliki ekstensi “dockerfile”. Namun, kami tidak menganjurkan Anda menggunakannya. Ini karena ekstensi dapat menyebabkan konflik dengan versi Docker tertentu atau aplikasi terkait lainnya.

#### Edit Dockerfile Anda menggunakan VS Code

Jika Dockerfile Anda memiliki ekstensi file, buka menu konteks (klik kanan) untuk file dan hapus ekstensi file.

Setelah ekstensi file dihapus dari Dockerfile Anda:

1. Buka Dockerfile kosong langsung di VS Code.
2. Salin isi contoh berikut ke Dockerfile Anda:

## Example Templat Gambar 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
```

Ini adalah Dockerfile yang menggunakan citra Ubuntu 18.04. YangMENJALANKANpetunjuk memperbarui cache paket. Instal paket perangkat lunak untuk server web, dan kemudian menulis “Hello World!” konten ke root dokumen server web. YangMEMBONGKARInstruksi mengekspos port 80 pada kontainer, danCMDInstruksi dimulai server web.

### 3. Simpan Dockerfile Anda.

#### Important

Pastikan bahwa Dockerfile Anda tidak memiliki ekstensi yang melekat pada nama. Dockerfile dengan ekstensi dapat menyebabkan konflik dengan versi Docker tertentu atau aplikasi terkait lainnya.

## 2. Membangun citra Anda dari Dockerfile Anda

Dockerfile yang Anda buat berisi informasi yang diperlukan untuk membangun gambar untuk sebuah program. Sebelum Anda dapat mendorong gambar tersebut ke instans Amazon ECR, Anda harus membuat gambar terlebih dahulu.

### Membangun citra dari Dockerfile Anda

1. Gunakan Docker CLI atau CLI yang terintegrasi dengan instance Docker Anda untuk menavigasi ke direktori yang berisi Dockerfile Anda.
2. JalankanMembangun Dockerperintah untuk membangun gambar yang didefinisikan dalam Dockerfile Anda.

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

3. JalankanGambar Dockerperintah untuk memverifikasi bahwa citra telah dibuat dengan benar.

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

Example Contoh output:

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

4.

#### Note

Langkah ini tidak diperlukan untuk membuat atau mendorong gambar Anda, tetapi Anda dapat melihat bagaimana gambar program bekerja ketika dijalankan.

Untuk menjalankan citra yang baru dibuat menggunakanJalankan Dockerperintah.

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

Yang-Pilihan yang ditentukan dalam contoh sebelumnya memetakan terkenapelabuhan 80 pada wadah untukpelabuhan 80 dari sistem host. Jika Anda menjalankan Docker secara lokal, navigasikan ke <http://localhost:80> menggunakan peramban web Anda. Jika program berjalan dengan benar, “Hello World!” Pernyataan ditampilkan.

Untuk informasi lebih lanjut tentang Jalankan Docker perintah, lihat [Referensi Docker](#) di situs web Docker.

### 3. Membuat repositori baru

Untuk mengunggah gambar Anda ke instans Amazon ECR Anda, buat repositori baru tempat ia dapat disimpan.

Membuat repositori Amazon ECR baru

1. Dari VS Code Bilah Aktivitas, choose AWS Ikon Toolkit.
2. Perluas AWS Penjelajah Menu.
3. Temukan default AWS Wilayah yang terkait dengan AWS Akun. Kemudian, pilih untuk melihat daftar layanan yang melalui Toolkit for VS Code.
4. Pilih ECR + pilihan untuk memulai Membuat repositori baru proses.
5. Ikuti petunjuk untuk menyelesaikan proses.
6. Setelah selesai, Anda dapat mengakses repositori baru Anda dari ECR bagian dari AWS Menu Explorer.

### 4. Dorong, tarik, dan hapus gambar

Setelah membuat gambar dari Dockerfile dan membuat repositori, Anda dapat mendorong gambar Anda ke repositori Amazon ECR. Selain itu, menggunakan AWS Explorer dengan Docker dan AWS CLI, Anda dapat melakukan hal berikut:

- Menarik citra dari repositori Anda.
- Hapus gambar yang disimpan di repositori Anda.
- Hapus repositori Anda.

## Mengautentikasi Docker dengan registri default

Otentikasi diperlukan untuk bertukar data antara instans Amazon ECR dan Docker. Untuk mengautentikasi Docker dengan registri Anda:

1. Buka sistem operasi baris perintah yang terhubung ke instance AndaAWSCLI.
2. Gunakan `get-login-password` metode untuk mengotentikasi ke registri ECR pribadi Anda.

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

### Important

Di perintah sebelumnya, Anda harus memperbarui keduanya `region` dan `AWS_account_id` untuk informasi yang spesifik untuk AndaAWSakun.

## Tandai dan mendorongnya citra ke repositori Anda

Setelah Anda mengautentikasi Docker dengan instanceAWS, mendorongnya citra ke repositori Anda.

1. Gunakan `Docker` perintah untuk melihat gambar yang Anda simpan secara lokal dan mengidentifikasi salah satu yang ingin Anda tag.

```
docker images
```

Example Contoh output:

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

2. Tag gambar Anda dengan `Tag` Docker perintah.

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

3. Dorong gambar yang ditandai ke repositori Anda dengan Tag Docker perintah.

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

Example Contoh output:

```
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
```

Setelah gambar yang ditandai Anda berhasil diunggah ke repositori Anda, itu terlihat di AWS Menu Explorer.

Menarik citra dari Amazon ECR

- Anda dapat menarik gambar ke instance lokal Anda Tag Docker perintah.

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

Example Contoh output:

```
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
```

## Menghapus gambar dari repositori Amazon ECR Anda

Ada dua metode untuk menghapus citra dari VS Code. Metode pertama adalah menggunakan `AWSPenjelajah`.

1. Dari `AWSExplorer`, memperluas `ECR` daftar makanan
2. Perluas repositori yang ingin Anda hapus citra
3. Pilih tag citra yang terkait dengan citra yang ingin Anda hapus, dengan membuka menu konteks (klik kanan)
4. Pilih `Hapus Tag...` pilihan untuk menghapus semua gambar yang tersimpan terkait dengan tag tersebut

## Menghapus citra menggunakan `AWSSCLI`

- Anda juga dapat menghapus citra dari repositori Anda dengan `AWSECR batch-delete-image` perintah.

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

Example Contoh output:

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

## Menghapus repositori dari instans Amazon ECR Anda

Ada dua metode untuk menghapus repositori dari VS Code. Metode pertama adalah menggunakan AWS Penjelajah.

1. Dari AWS Explorer, memperluas ECR daftar makanan
2. Pilih repositori yang ingin Anda hapus dengan membuka menu konteks (klik kanan)
3. Pilih Hapus Repositori... pilihan untuk repositori yang dipilih

## Menghapus repositori Amazon ECR dari AWS CLI

- Anda dapat menghapus repositori dengan `aws ecr delete-repository` perintah.

### Note

Secara default, Anda tidak dapat menghapus repositori yang berisi citra. Namun, `--force` bendera memungkinkan ini.

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

### Example Contoh output:

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

```
]
}
```

## Bekerja dengan Layanan Wadah Amazon Elastic

Parameter AWS Toolkit for Visual Studio Code menyediakan beberapa dukungan untuk [Layanan Amazon Elastic Container \(Amazon ECS\)](#). Toolkit for VS Code membantu Anda dalam pekerjaan terkait Amazon ECS tertentu, seperti membuat definisi tugas.

### Topik

- [Menggunakan file definisi tugas IntelliSense untuk Amazon ECS](#)
- [Exec Layanan Kontainer Elastis Amazon di AWS Toolkit for Visual Studio Code](#)

## Menggunakan file definisi tugas IntelliSense untuk Amazon ECS

Salah satu hal yang mungkin Anda lakukan saat bekerja dengan Amazon Elastic Container Service (Amazon ECS) adalah membuat definisi tugas, seperti yang dijelaskan dalam [Membuat Definisi Tugas](#) dari Panduan Pengembang Amazon Elastic Container Service. Ketika Anda menginstal AWS Toolkit for Visual Studio Code, instalasi mencakup fungsionalitas IntelliSense untuk file definisi tugas Amazon ECS.

### Prasyarat

- Pastikan sistem Anda memenuhi prasyarat yang ditentukan dalam [Memasang Toolkit for VS Code](#).

## Menggunakan IntelliSense di file definisi tugas Amazon ECS

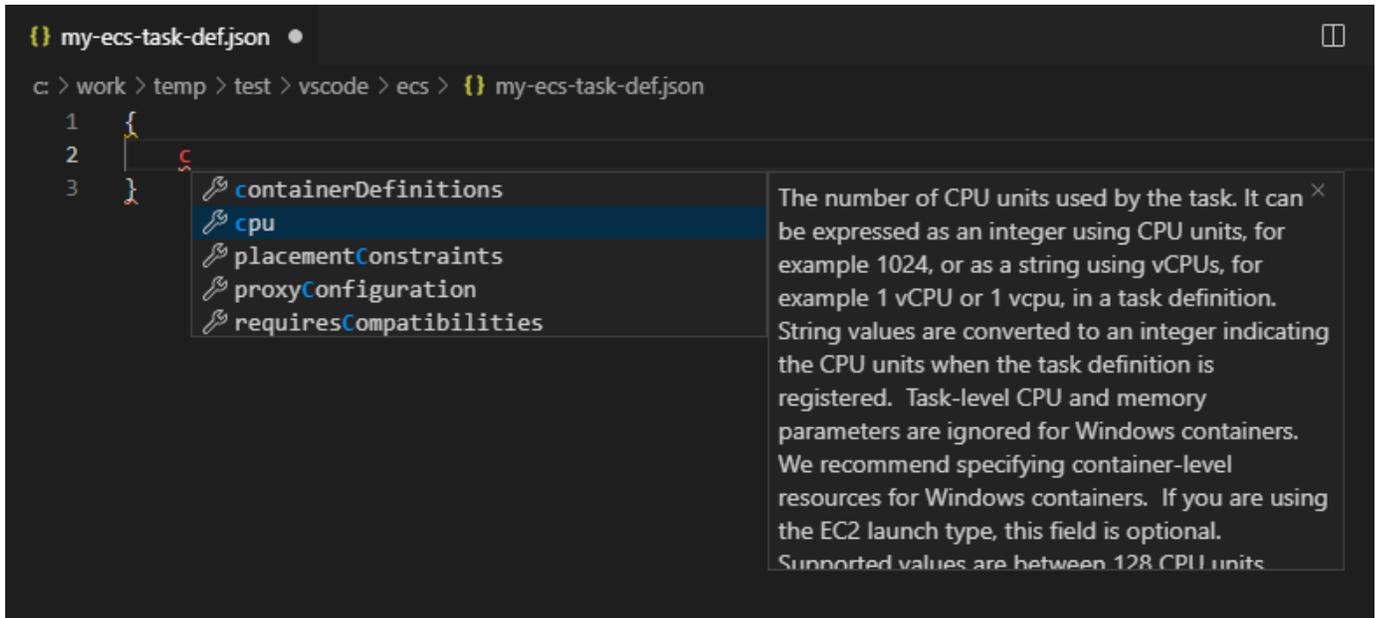
Contoh berikut menunjukkan kepada Anda bagaimana Anda dapat memanfaatkan IntelliSense di file definisi tugas Amazon ECS.

1. Buat file JSON untuk definisi tugas Amazon ECS Anda. Nama file harus memiliki `ecs-task-def.json` pada akhirnya, tetapi dapat memiliki karakter tambahan di awal.

Untuk contoh ini, buat sebuah file bernama `my-ecs-task-def.json`

2. Buka file di editor VS Code dan masukkan kurung kurawal awal.

- Masukkan huruf “c” seolah-olah Anda ingin menambahkan definisi. Amati dialog IntelliSense yang terbuka, yang serupa dengan yang berikut ini.



## Exec Layanan Kontainer Elastis Amazon di AWS Toolkit for Visual Studio Code

Anda dapat mengeluarkan perintah tunggal dalam wadah Amazon Elastic Container Service (Amazon ECS) dengan AWS Toolkit for Visual Studio Code container, menggunakan fitur Amazon ECS Exec.

### ⚠ Important

Mengaktifkan dan Menonaktifkan Amazon ECS Exec mengubah status sumber daya di akun Anda. AWS Ini termasuk menghentikan dan memulai kembali layanan. Mengubah status sumber daya saat Amazon ECS Exec diaktifkan dapat menyebabkan hasil yang tidak terduga. Untuk informasi selengkapnya tentang Amazon ECS, lihat panduan pengembang [Menggunakan Amazon ECS Exec](#) untuk Debugging.

## Prasyarat Amazon ECS Exec

Sebelum Anda dapat menggunakan fitur Amazon ECS Exec, ada beberapa syarat prasyarat yang perlu dipenuhi.

## Persyaratan Amazon ECS

Bergantung pada apakah tugas Anda di-host di Amazon EC2 atau, AWS Fargate (Fargate) Amazon ECS Exec memiliki persyaratan versi yang berbeda.

- Jika Anda menggunakan Amazon EC2, Anda harus menggunakan AMI Amazon ECS yang dioptimalkan yang dirilis setelah 20 Januari 2021, dengan versi agen 1.50.2 atau lebih tinggi. Informasi tambahan tersedia untuk Anda di panduan pengembang [AMI yang dioptimalkan Amazon ECS](#).
- Jika Anda menggunakan AWS Fargate, Anda harus menggunakan platform versi 1.4.0 atau lebih tinggi. Informasi tambahan tentang persyaratan Fargate tersedia untuk Anda di versi [AWS Fargate platform](#) panduan pengembang.

## AWS konfigurasi akun dan izin IAM

Untuk menggunakan fitur Amazon ECS Exec, Anda harus memiliki kluster Amazon ECS yang ada yang terkait dengan akun Anda. AWS Amazon ECS Exec menggunakan Systems Manager untuk membuat koneksi dengan container di kluster Anda dan memerlukan Izin Peran IAM Tugas khusus untuk berkomunikasi dengan layanan SSM.

Anda dapat menemukan informasi peran dan kebijakan IAM, khusus untuk Amazon ECS Exec, di [izin IAM yang diperlukan untuk](#) panduan pengembang ECS Exec.

## Bekerja dengan Amazon ECS Exec

Anda dapat mengaktifkan atau menonaktifkan Amazon ECS Exec langsung dari AWS Explorer di Toolkit for VS Code. Ketika Anda telah mengaktifkan Amazon ECS Exec, Anda dapat memilih kontainer dari menu Amazon ECS dan kemudian menjalankan perintah terhadapnya.

## Mengaktifkan Amazon ECS Exec

1. Dari AWS Explorer, cari dan perluas menu Amazon ECS.
2. Perluas cluster dengan layanan yang ingin Anda modifikasi.
3. Buka menu konteks untuk (klik kanan) layanan dan pilih Aktifkan Eksekusi Perintah.

 Important

Ini akan memulai penyebaran baru Layanan Anda dan mungkin memakan waktu beberapa menit. Untuk informasi lebih lanjut, lihat catatan di awal bagian ini.

### Menonaktifkan Amazon ECS Exec

1. Dari AWS Explorer, cari dan perluas menu Amazon ECS.
2. Perluas cluster yang menampung layanan yang Anda inginkan.
3. Buka menu konteks untuk (klik kanan) layanan dan pilih Nonaktifkan Eksekusi Perintah.

 Important

Ini akan memulai penyebaran baru Layanan Anda dan mungkin memakan waktu beberapa menit. Untuk informasi lebih lanjut, lihat catatan di awal bagian ini.)

### Menjalankan perintah terhadap Container

Untuk menjalankan perintah terhadap penampung menggunakan AWS Explorer, Amazon ECS Exec harus diaktifkan. Jika tidak diaktifkan, lihat prosedur Mengaktifkan ECS Exec di bagian ini.

1. Dari AWS Explorer, cari dan perluas menu Amazon ECS.
2. Perluas cluster yang menampung layanan yang Anda inginkan.
3. Perluas layanan untuk membuat daftar kontainer terkait.
4. Buka menu konteks untuk (klik kanan) wadah dan pilih Run Command in Container.
5. Prompt akan terbuka dengan daftar Tugas yang sedang berjalan, pilih ARN Tugas yang Anda inginkan.

 Note

Jika hanya satu Tugas yang berjalan untuk Layanan itu, itu akan dipilih secara otomatis dan langkah ini akan dilewati.

6. Saat diminta, ketik perintah yang ingin Anda jalankan dan tekan Enter untuk memproses.

# Bekerja dengan Amazon EventBridge

Parameter AWS Toolkit for Visual Studio Code (VS Code) menyediakan dukungan untuk [Amazon EventBridge](#). Menggunakan Toolkit for VS Code, Anda dapat bekerja dengan aspek-aspek tertentu dari EventBridge, seperti skema.

## Topik

- [Bekerja dengan Amazon EventBridge Schemas](#)

## Bekerja dengan Amazon EventBridge Schemas

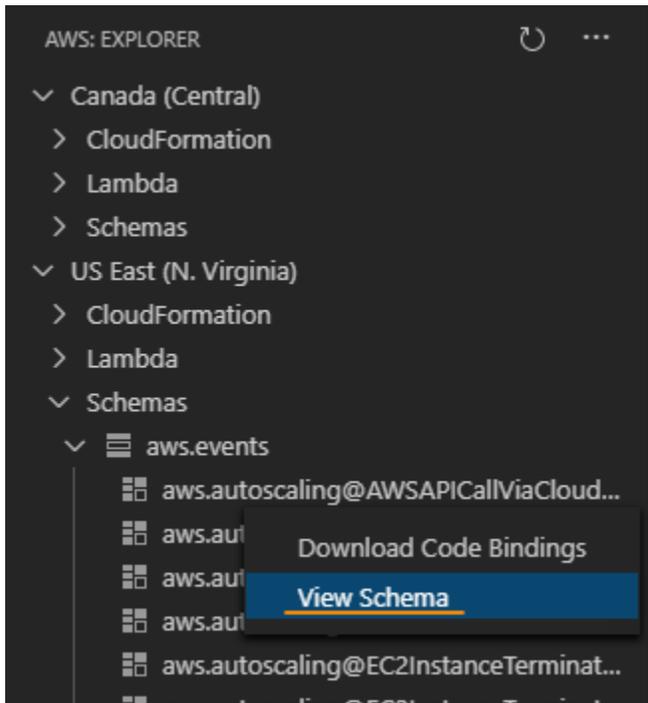
Anda dapat menggunakan AWS Toolkit for Visual Studio Code (Kode VS) untuk melakukan berbagai operasi pada [EventBridge skema Amazon](#).

## Prasyarat

- Pastikan sistem Anda memenuhi prasyarat yang ditentukan dalam [Menginstal Toolkit for VS Code](#).
- EventBridge Skema yang ingin Anda gunakan harus tersedia di AWS akun Anda. Jika tidak, buat atau unggah. Lihat [EventBridge Skema Amazon](#) di [Panduan EventBridge Pengguna Amazon](#).

## Lihat Skema yang Tersedia

1. Di AWS Explorer, perluas Skema.
2. Perluas nama registri yang berisi skema yang ingin Anda lihat. Misalnya, banyak skema yang AWS memasok ada di registri `aws.events`.
3. Untuk melihat skema di editor, buka menu konteks skema, lalu pilih Lihat Skema.

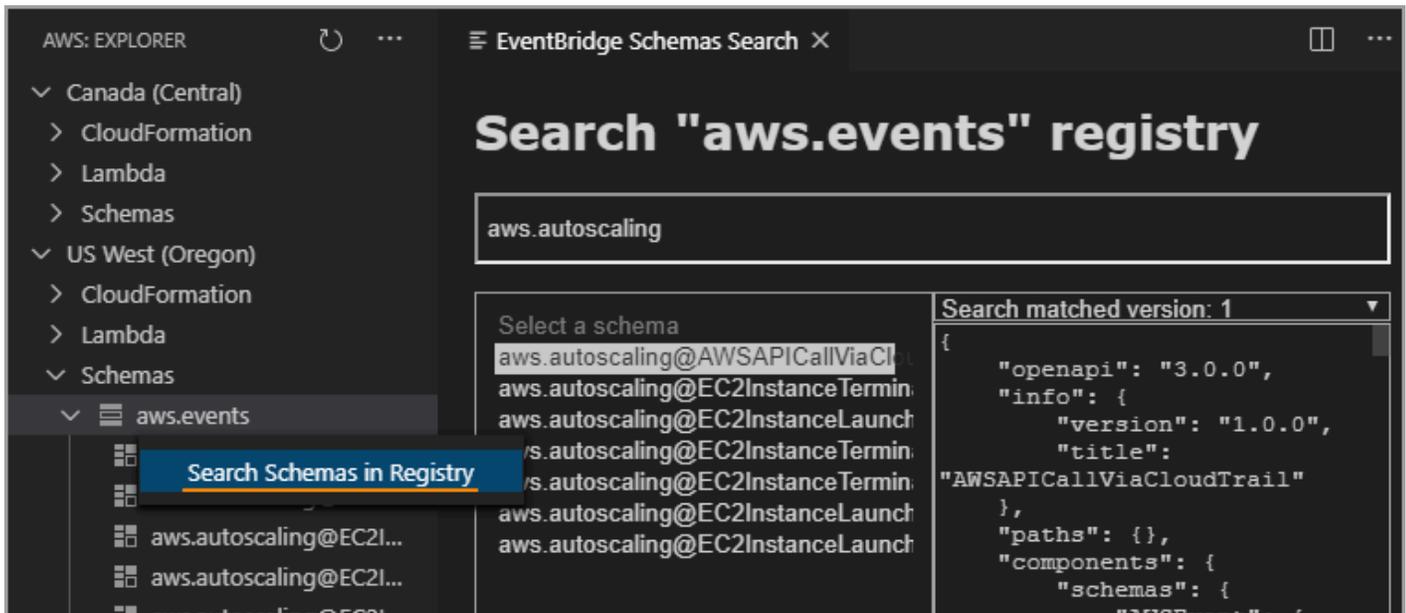


## Temukan Skema yang Tersedia

Di AWS Explorer, lakukan satu atau beberapa hal berikut:

- Mulailah mengetik judul skema yang ingin Anda temukan. AWS Explorer menyoroti judul skema yang berisi kecocokan. (Registri harus diperluas agar Anda dapat melihat judul yang disorot.)
- Buka menu konteks untuk Skema, lalu pilih Skema Pencarian. Atau perluas Skema, buka menu konteks untuk registri yang berisi skema yang ingin Anda temukan, lalu pilih Skema Pencarian di Registri. Dalam kotak dialog EventBridge Schemas Search, mulailah mengetik judul skema yang ingin Anda temukan. Kotak dialog menampilkan judul skema yang berisi kecocokan.

Untuk menampilkan skema di kotak dialog, pilih judul skema.



## Hasilkan Kode untuk Skema yang Tersedia

1. Di AWS Explorer, perluas Skema.
2. Perluas nama registri yang berisi skema yang ingin Anda buat kodenya.
3. Klik kanan judul skema, lalu pilih Download code binding.
4. Di halaman wizard yang dihasilkan, pilih yang berikut ini:
  - Versi skema
  - Bahasa pengikatan kode
  - Folder ruang kerja tempat Anda ingin menyimpan kode yang dihasilkan di mesin pengembangan lokal Anda

## AWS IAM Penganalisis Akses

Anda dapat menjalankan pemeriksaan kebijakan [AWS Identity and Access Management \(IAM\) Access Analyzer](#) pada IAM kebijakan yang ditulis dalam AWS CloudFormation template, paket Terraform, dan dokumen JSON kebijakan, menggunakan Access Analyzer di IAM file. AWS Toolkit for Visual Studio Code

IAMPemeriksaan kebijakan Access Analyzer mencakup validasi kebijakan dan pemeriksaan kebijakan khusus. Validasi kebijakan membantu memvalidasi IAM kebijakan Anda sesuai dengan

standar yang dirinci dalam [Tata Bahasa bahasa IAM JSON kebijakan](#) dan [praktik terbaik AWS Keamanan dalam IAM](#) topik, yang terdapat di Panduan Pengguna. AWS Identity and Access Management Temuan validasi kebijakan Anda mencakup peringatan keamanan, kesalahan, peringatan umum, dan saran kebijakan.

Anda juga dapat menjalankan pemeriksaan kebijakan khusus untuk akses baru, berdasarkan standar keamanan Anda. Biaya terkait dengan setiap pemeriksaan kebijakan khusus untuk akses baru. Untuk informasi rinci tentang harga, lihat situs [harga AWS IAM Access Analyzer](#). Untuk detail tentang pemeriksaan kebijakan IAM Access Analyzer, lihat topik [Cek untuk memvalidasi kebijakan](#) di AWS Identity and Access Management Panduan Pengguna.

Topik berikut menjelaskan cara bekerja dengan pemeriksaan kebijakan IAM Access Analyzer di AWS Toolkit for Visual Studio Code

Topik

- [Bekerja dengan AWS IAM Access Analyzer](#)

## Bekerja dengan AWS IAM Access Analyzer

Bagian berikut menjelaskan cara melakukan validasi IAM kebijakan dan pemeriksaan kebijakan khusus di bagian. AWS Toolkit for Visual Studio Code Untuk detail tambahan, lihat topik berikut di Panduan AWS Identity and Access Management Pengguna: [Validasi kebijakan IAM Access Analyzer dan pemeriksaan kebijakan kustom IAM Access Analyzer](#).

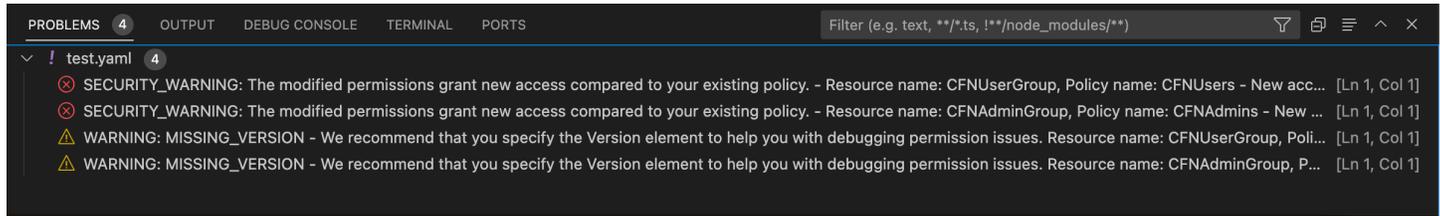
### Prasyarat

Prasyarat berikut harus dipenuhi sebelum Anda dapat bekerja dengan pemeriksaan kebijakan IAM Access Analyzer dari Toolkit.

- Instal Python versi 3.6 atau yang lebih baru.
- Instal [Validator IAM Kebijakan untuk AWS CloudFormation](#) atau [Validator IAM Kebijakan untuk Terraform yang](#) diperlukan oleh CLI alat Python dan ditentukan di jendela Pemeriksaan Kebijakan. IAM
- Konfigurasi kredensial AWS Peran Anda.

## IAMPemeriksaan kebijakan Access Analyzer

Anda dapat melakukan pemeriksaan kebijakan untuk AWS CloudFormation templat, paket Terraform, dan dokumen JSON Kebijakan, menggunakan. AWS Toolkit for Visual Studio Code Temuan pemeriksaan Anda dapat dilihat di Panel Masalah Kode VS. Gambar berikut menunjukkan Panel Masalah Kode VS.



IAMAccess Analyzer menyediakan 4 jenis pemeriksaan:

- Kebijakan Validasi
- CheckAccessNotGranted
- CheckNoNewAccess
- CheckNoPublicAccess

Bagian berikut menjelaskan cara menjalankan setiap jenis pemeriksaan.

### Note

Konfigurasi kredensial AWS Peran Anda sebelum menjalankan semua jenis pemeriksaan. File yang didukung mencakup jenis dokumen berikut: AWS CloudFormation templat, paket Terraform, dan JSON dokumen Kebijakan. Referensi jalur file biasanya disediakan oleh administrator atau tim keamanan Anda, dan dapat berupa jalur file sistem atau bucket Amazon S3. URI Untuk menggunakan bucket Amazon S3, peran Anda saat ini harus memiliki akses ke bucket Amazon S3. Biaya terkait dengan setiap pemeriksaan kebijakan khusus. Untuk detail tentang kebijakan kustom, periksa harga, lihat panduan [harga AWS IAM Access Analyzer](#).

## Menjalankan Kebijakan Validasi

Pemeriksaan Kebijakan Validasi, juga dikenal sebagai validasi kebijakan, memvalidasi kebijakan Anda terhadap tata bahasa IAM kebijakan dan praktik terbaik. AWS Untuk informasi tambahan, lihat

[Tata Bahasa bahasa IAM JSON kebijakan](#) dan [praktik terbaik AWS Keamanan dalam IAM](#) topik, yang terdapat di Panduan AWS Identity and Access Management Pengguna.

1. Dari VS Code, buka file yang didukung yang berisi AWS IAM Kebijakan, di editor Kode VS.
2. Untuk membuka pemeriksaan kebijakan IAM Access Analyzer, buka Palet Perintah Kode VS dengan menekan **CTRL+Shift+P**, cari **IAM Policy Checks**, lalu klik untuk membuka panel Pemeriksaan IAM Kebijakan di editor Kode VS.
3. Dari panel Pemeriksaan IAM Kebijakan, pilih jenis dokumen Anda dari menu tarik-turun.
4. Dari bagian Validasi Kebijakan, pilih tombol Jalankan Validasi Kebijakan untuk menjalankan pemeriksaan Kebijakan Validasi.
5. Dari Panel Masalah di VS Code, tinjau temuan pemeriksaan kebijakan Anda.
6. Perbarui kebijakan Anda dan ulangi prosedur ini, jalankan kembali pemeriksaan Kebijakan Validasi hingga temuan pemeriksaan kebijakan Anda tidak lagi menampilkan peringatan atau kesalahan keamanan.

### Berlari CheckAccessNotGranted

CheckAccessNotGranted adalah pemeriksaan kebijakan khusus untuk memverifikasi bahwa IAM tindakan tertentu tidak diizinkan oleh kebijakan Anda.

#### Note

Referensi jalur file biasanya disediakan oleh administrator atau tim keamanan Anda, dan dapat berupa jalur file sistem atau bucket Amazon S3. URI Untuk menggunakan bucket Amazon S3 URI, peran Anda saat ini harus memiliki akses ke bucket Amazon S3. Setidaknya satu tindakan atau sumber daya harus ditentukan dan file harus terstruktur setelah contoh berikut:

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

1. Dari VS Code, buka file yang didukung yang berisi AWS IAM Kebijakan, di editor Kode VS.

2. Untuk membuka pemeriksaan kebijakan IAM Access Analyzer, buka Palet Perintah Kode VS dengan menekan **CTRL+Shift+P**, cari **IAM Policy Checks**, lalu klik untuk membuka panel Pemeriksaan IAM Kebijakan di editor Kode VS.
3. Dari panel Pemeriksaan IAM Kebijakan, pilih jenis dokumen Anda dari menu tarik-turun.
4. Dari bagian Pemeriksaan Kebijakan Kustom, pilih **CheckAccessNotGranted**.
5. Di bidang input teks, Anda dapat memasukkan daftar dipisahkan koma yang berisi tindakan dan sumber daya. ARNs Setidaknya satu tindakan atau sumber daya harus disediakan.
6. Pilih tombol Jalankan Pemeriksaan Kebijakan Kustom.
7. Dari Panel Masalah di VS Code, tinjau temuan pemeriksaan kebijakan Anda. Pemeriksaan kebijakan khusus mengembalikan FAIL hasil PASS atau.
8. Perbarui kebijakan Anda dan ulangi prosedur ini, jalankan kembali **CheckAccessNotGranted** pemeriksaan hingga kembali PASS.

### Berlari CheckNoNewAccess

**CheckNoNewAccess** adalah pemeriksaan kebijakan khusus untuk memverifikasi apakah kebijakan Anda memberikan akses baru dibandingkan dengan kebijakan referensi.

1. Dari VS Code, buka file yang didukung yang berisi AWS IAM Kebijakan, di editor Kode VS.
2. Untuk membuka pemeriksaan kebijakan IAM Access Analyzer, buka Palet Perintah Kode VS dengan menekan **CTRL+Shift+P**, cari **IAM Policy Checks**, lalu klik untuk membuka panel Pemeriksaan IAM Kebijakan di editor Kode VS.
3. Dari panel Pemeriksaan IAM Kebijakan, pilih jenis dokumen Anda dari menu tarik-turun.
4. Dari bagian Pemeriksaan Kebijakan Kustom, pilih **CheckNoNewAccess**.
5. Masukkan dokumen JSON kebijakan referensi. Atau, Anda dapat menyediakan jalur file yang mereferensikan dokumen JSON kebijakan.
6. Pilih Jenis Kebijakan Referensi yang cocok dengan jenis dokumen referensi Anda.
7. Pilih tombol Jalankan Pemeriksaan Kebijakan Kustom.
8. Dari Panel Masalah di VS Code, tinjau temuan pemeriksaan kebijakan Anda. Pemeriksaan kebijakan khusus mengembalikan FAIL hasil PASS atau.
9. Perbarui kebijakan Anda dan ulangi prosedur ini, jalankan kembali **CheckNoNewAccess** pemeriksaan hingga kembali PASS.

## Berlari CheckNoPublicAccess

CheckNoPublicAccess adalah pemeriksaan kebijakan khusus untuk memverifikasi apakah kebijakan Anda memberikan akses publik ke jenis sumber daya yang didukung dalam templat Anda.

Untuk informasi spesifik tentang jenis sumber daya yang didukung, lihat [cloudformation-iam-policy-validator](#) dan [terraform-iam-policy-validator](#) GitHub repositori.

1. Dari VS Code, buka file yang didukung yang berisi AWS IAM Kebijakan, di editor Kode VS.
2. Untuk membuka pemeriksaan kebijakan IAM Access Analyzer, buka Palet Perintah Kode VS dengan menekan **CTRL+Shift+P**, cari **IAM Policy Checks**, lalu klik untuk membuka panel Pemeriksaan IAM Kebijakan di editor Kode VS.
3. Dari panel Pemeriksaan IAM Kebijakan, pilih jenis dokumen Anda dari menu tarik-turun.
4. Dari bagian Pemeriksaan Kebijakan Kustom, pilih CheckNoPublicAccess.
5. Pilih tombol Jalankan Pemeriksaan Kebijakan Kustom.
6. Dari Panel Masalah di VS Code, tinjau temuan pemeriksaan kebijakan Anda. Pemeriksaan kebijakan khusus mengembalikan FAIL hasil PASS atau.
7. Perbarui kebijakan Anda dan ulangi prosedur ini, jalankan kembali CheckNoNewAccess pemeriksaan hingga kembali PASS.

## Bekerja dengan AWS IoT di AWS Toolkit for Visual Studio Code

AWS IoT di AWS Toolkit for Visual Studio Code memungkinkan Anda untuk berinteraksi dengan AWS IoT layanan, sambil meminimalkan interupsi ke alur kerja Anda di VS Code. Panduan pengguna ini dimaksudkan untuk membantu Anda mulai menggunakan AWS IoT fitur layanan yang tersedia di AWS Toolkit for Visual Studio Code. Untuk informasi tambahan tentang AWS IoT layanan, lihat panduan pengembang [Apa AWS IoT?](#)

### Prasyarat AWS IoT

Untuk mulai menggunakan AWS IoT dari Toolkit for VS Code, pastikan AWS akun dan Kode VS memenuhi persyaratan dalam panduan ini:

- Untuk AWS persyaratan akun dan AWS izin pengguna khusus untuk AWS IoT layanan, lihat [Memulai dengan AWS IoT Core](#) panduan developer.
- Untuk Toolkit untuk persyaratan khusus VS Code, lihat [Menyiapkan Toolkit for VS Code](#) panduan pengguna.

## AWS IoT Hal

AWS IoT menghubungkan perangkat ke AWS layanan cloud dan sumber daya. Anda dapat menghubungkan perangkat AWS IoT dengan menggunakan benda-benda yang disebut barang. Suatu hal adalah representasi dari perangkat tertentu atau entitas logis. Ini bisa berupa perangkat fisik atau sensor (misalnya bola lampu atau sakelar di dinding). Untuk informasi tambahan tentang AWS IoT Hal, lihat panduan pengembang [Mengelola perangkat dengan AWS IoT](#).

### Mengelola AWS IoT barang

Toolkit for VS Code memiliki beberapa fitur yang membuat AWS IoT manajemen hal lebih efisien. Ini adalah cara agar Anda dapat menggunakan toolkit VS Code untuk mengelola AWS IoT Hal-hal:

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

Untuk membuat sesuatu

1. Dari: **AWSExplorer**, memperluas **IoT** judul layanan, dan pilih konteks (klik kanan) Hal.
2. Memilih **Membuat Hal** dari menu konteks untuk membuka kotak dialog.
3. Ikuti prompt dengan memasukkan nama untuk hal IoT Anda ke **Nama Hal** Bidang.
4. Ketika ini selesai, **ikon Hal** diikuti dengan nama yang Anda tentukan akan terlihat di **Hal** Bidang.

Untuk melampirkan sertifikat untuk sesuatu

1. Dari: **AWSExplorer**, memperluas **IoT** bagian layanan.
2. Di bawah **Hal** layar, cari **Hal** mana Anda melampirkan sertifikat.
3. **Konteks-pilih** (klik kanan) **Hal** dan pilih **Lampirkan Sertifikat** dari menu konteks, untuk membuka **pemilih masukan** dengan daftar sertifikat Anda.
4. Dari daftar, pilih **ID Sertifikat** yang sesuai dengan sertifikat yang ingin Anda lampirkan ke barang Anda.

5. Ketika ini selesai, sertifikat Anda dapat diakses diAWSExplorer, sebagai item dari hal yang Anda melampirkan ke.

Untuk melepaskan sertifikat dari sesuatu

1. Dari:AWSExplorer, memperluasIoTbagian layanan
2. DiHalayat, cari hal yang ingin Anda lepaskan sertifikat dari.
3. Konteks-pilih (klik kanan) hal dan pilih Lepaskan Sertifikat dari menu konteks.
4. Ketika ini selesai, sertifikat terpisah tidak akan lagi ditampilkan di bawah hal itu diAWSExplorer, tetapi masih akan dapat diakses dari Sertifikatayat.

Untuk menghapus sesuatu

1. Dari:AWSExplorer, memperluasIoTbagian layanan.
2. DiHalayat, cari hal yang ingin menghapus.
3. Konteks-pilih (klik kanan) benda dan pilih Memilih Haldari menu konteks untuk menghapusnya.
4. Ketika ini selesai, hal yang dihapus tidak akan lagi tersedia dariHalayat.

#### Note

Catatan: Anda hanya dapat menghapus sesuatu yang tidak memiliki sertifikat yang melekat padanya.

## Sertifikat AWS IoT

Sertifikat adalah cara umum untuk membuat koneksi yang aman antaraAWS IoTlayanan dan perangkat. Sertifikat X.509 adalah sertifikat digital yang menggunakan standar infrastruktur kunci publik X.509 untuk mengaitkan kunci publik dengan identitas yang terdapat dalam sertifikat. Untuk informasi tambahan tentangAWS IoTsertifikat, lihat panduan pengembang [Otentikasi \(IoT\)](#).

## Mengelola sertifikat

Toolkit VS Code menawarkan berbagai cara bagi Anda untuk mengelolaAWS IoTsertifikat, langsung dariAWSExplorer.

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

Untuk membuat AWS IoT sertifikat

Sertifikat X.509 dapat digunakan untuk terhubung dengan instance AWS IoT.

1. Dari: AWS Explorer, memperluas IoT bagian layanan, dan konteks-pilih (klik kanan) Sertifikat.
2. Memilih Membuat Sertifikat dari menu konteks untuk membuka kotak dialog.
3. Pilih direktori di sistem file lokal Anda untuk menyimpan key pair RSA dan sertifikat X.509.

#### Note

- Nama file default berisi ID sertifikat sebagai awalan.
- Hanya sertifikat X.509 yang disimpan bersama AWS Sakun, melalui AWS IoT layanan.
- key pair RSA Anda hanya dapat dikeluarkan satu kali, menyimpannya ke lokasi yang aman di sistem file Anda saat Anda diminta.
- Jika sertifikat atau key pair tidak dapat disimpan ke sistem file Anda saat ini, maka AWS Toolkit menghapus sertifikat dari AWS Sakun.

Mengubah status sertifikat

Status sertifikat individual ditampilkan di samping ID-nya di AWS Explorer dan dapat diatur ke: aktif, tidak aktif, atau dicabut.

#### Note

- Sertifikat Anda membutuhkan aktif status sebelum Anda dapat menggunakannya untuk menghubungkan perangkat Anda ke AWS IoT layanan.
- Sesitidak aktif sertifikat dapat diaktifkan, apakah itu telah dinonaktifkan sebelumnya atau tidak aktif secara default.

- Sertifikat yang telah dibatalkan Tidak dapat diaktifkan kembali.

1. Dari: **AWSExplorer**, memperluas **IoT** bagian layanan.
2. Di **Sertifikat** tab, cari sertifikat yang ingin Anda ubah.
3. Pilih konteks (klik kanan) sertifikat untuk membuka menu konteks yang menampilkan opsi perubahan status yang tersedia untuk sertifikat tersebut.

- Jika sertifikat memiliki status tidak aktif, pilih **aktifkan** untuk mengubah status aktif.
- Jika sertifikat memiliki status aktif, pilih **nonaktifkan** untuk mengubah status tidak aktif.
- Jika sertifikat memiliki **aktif** atau **tidak aktif** status, pilih **cabut** untuk mengubah status dibatalkan.

#### Note

Masing-masing tindakan perubahan status ini juga tersedia jika Anda memilih sertifikat yang dilampirkan ke sesuatu saat ditampilkan di **Halay**.

Untuk melampirkan kebijakan IoT ke sertifikat

1. Dari: **AWSExplorer**, memperluas **IoT** bagian layanan.
2. Di **Sertifikat** tab, cari sertifikat yang ingin Anda ubah.
3. Pilih konteks (klik kanan) sertifikat dan pilih **Lampirkan Kebijakan** dari menu konteks, untuk membuka pemilih masukan dengan daftar kebijakan yang tersedia.
4. Pilih kebijakan yang ingin Anda lampirkan ke sertifikat.
5. Setelah selesai, kebijakan yang Anda pilih akan ditambahkan ke sertifikat sebagai item sub-menu.

Untuk melepaskan kebijakan IoT dari sertifikat

1. Dari: **AWSExplorer**, memperluas **IoT** bagian layanan.
2. Di **Sertifikat** tab, cari sertifikat yang ingin Anda ubah.
3. Perluas sertifikat dan temukan kebijakan yang ingin Anda lepaskan.

4. Pilih konteks (klik kanan) kebijakan dan pilih **Lepaskan** dari menu konteks.
5. Ketika ini selesai, kebijakan tidak akan lagi menjadi item yang dapat diakses dari sertifikat Anda, tetapi akan tersedia dari **Kebijakan**ayat.

Untuk menghapus sertifikat

1. Dari **AWSExplorer**, memperluas **IoT**pos layanan.
2. Di **Sertifikat**ayat, cari sertifikat yang ingin Anda hapus.
3. Pilih konteks (klik kanan) sertifikat dan pilih **Menghapus Sertifikat** dari menu konteks.

#### Note

Anda tidak dapat menghapus sertifikat jika terpasang ke suatu benda atau memiliki status aktif. Anda dapat menghapus sertifikat yang memiliki kebijakan terlampir.

## Kebijakan AWS IoT

**AWS IoT**kebijakan inti didefinisikan melalui dokumen JSON, masing-masing berisi satu atau lebih pernyataan kebijakan. Kebijakan mendefinisikan bagaimana **AWS IoT**, **AWS**, dan perangkat Anda dapat berinteraksi satu sama lain. Untuk informasi lebih lanjut tentang cara membuat dokumen kebijakan, lihat panduan developer (panduan developer) [Kebijakan IoT](#).

#### Note

Kebijakan bernama versi sehingga Anda dapat memutar mereka kembali. Dalam **The AWSExplorer**, kebijakan IoT Anda tercantum di bawah **Kebijakan**ayat, dalam layanan IoT. Anda dapat melihat versi kebijakan dengan memperluas kebijakan. Versi default dilambangkan dengan tanda bintang.

## Mengelola kebijakan

Toolkit for VS Code menawarkan beberapa cara bagi Anda untuk mengelola **AWS IoT**kebijakan layanan. Ini adalah cara agar Anda dapat mengelola atau memodifikasi kebijakan Anda langsung dari **AWSExplorer** dalam Kode VS:

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

Untuk membuat AWS IoT kebijakan

 Note

Anda dapat membuat kebijakan baru dari AWS Explorer, tetapi dokumen JSON yang mendefinisikan kebijakan harus sudah ada dalam sistem file Anda.

1. Dari AWS Explorer, memperluas IoT bagian layanan.
2. Konteks-pilih (klik kanan) Kebijakan dan pilih Buat Kebijakan dari Dokumen, untuk membuka Nama Kebijakan Bidang input.
3. Masukkan nama dan ikuti petunjuk untuk membuka dialog yang meminta Anda memilih dokumen JSON dari sistem file Anda.
4. Pilih file JSON yang berisi definisi kebijakan Anda, kebijakan akan tersedia di AWS Explorer ketika ini selesai.

Untuk mengunggah yang baru AWS IoT versi kebijakan

Versi baru kebijakan dapat dibuat dengan mengunggah dokumen JSON ke kebijakan.

 Note

Dokumen JSON baru harus ada di sistem file Anda untuk membuat versi baru menggunakan AWS Explorer.

1. Dari AWS Explorer, memperluas IoT bagian layanan.
2. Perluas Kebijakan untuk melihat AWS IoT kebijakan

3. Pilih konteks (klik kanan) kebijakan yang ingin Anda perbarui dan pilih **Buat versi baru dari Document**.
4. Saat dialog terbuka, pilih file JSON yang berisi pembaruan untuk definisi kebijakan Anda.
5. Versi baru akan dapat diakses dari kebijakan Anda di **AWSExplorer**.

#### Untuk menyunting AWS IoT versi kebijakan

Dokumen kebijakan dapat dibuka dan diedit menggunakan Kode VS. Setelah Anda selesai mengedit dokumen, Anda dapat menyimpannya ke sistem file Anda. Kemudian, Anda dapat mengunggahnya ke **AWS IoT** layanan dari **AWSExplorer**.

1. Dari **AWSExplorer**, memperluas **IoT** bagian layanan.
2. Perluas **Kebijakan** ayat dan cari kebijakan yang ingin Anda perbarui. **Buat Kebijakan** dari Dokumen **membuka Nama Kebijakan Bidang** input.
3. Perluas kebijakan yang ingin Anda perbarui, lalu pilih **Konteks (klik kanan)** versi kebijakan yang ingin Anda edit.
4. **Memilih Lihat** dari menu konteks untuk membuka versi kebijakan di **VS Code**
5. Ketika dokumen kebijakan dibuka, buat dan simpan perubahan yang Anda inginkan.

#### Note

Pada titik ini, perubahan yang Anda buat pada kebijakan hanya disimpan ke sistem file lokal Anda. Untuk memperbarui versi dan melacaknya dengan **AWSExplorer**, ulangi langkah-langkah yang dijelaskan dalam [Upload a new policy version](#) Prosedur

#### Untuk memilih default versi kebijakan baru

1. Dari **AWSExplorer**, memperluas **IoT** bagian layanan.
2. Perluas **Kebijakan** ayat dan cari kebijakan yang ingin Anda perbarui.
3. Perluas kebijakan yang ingin Anda perbarui, lalu pilih **Konteks (klik kanan)** versi kebijakan yang ingin Anda tetapkan dan pilih **Tetapkan sebagai Default**.
4. Ketika ini selesai, versi default baru yang Anda pilih akan memiliki bintang yang terletak di sebelahnya.

## Untuk menghapus kebijakan

### Note

Sebelum Anda dapat menghapus kebijakan atau versi kebijakan, ada ketentuan yang perlu dipenuhi.

- Anda tidak dapat menghapus kebijakan jika terpasang ke sertifikat.
- Anda tidak dapat menghapus kebijakan jika memiliki versi non-default.
- Anda tidak dapat menghapus versi default kebijakan kecuali versi default baru dipilih, atau seluruh kebijakan dihapus.
- Sebelum Anda dapat menghapus seluruh kebijakan, semua versi non-default kebijakan tersebut harus dihapus terlebih dahulu.

1. Dari:AWSExplorer, memperluasloTbagian layanan.
2. PerluasKebijakanayat dan cari kebijakan yang ingin Anda perbarui.
3. Perluas kebijakan yang ingin Anda perbarui, lalu pilih Konteks (klik kanan) versi kebijakan yang ingin Anda hapus dan pilihHapus.
4. Ketika versi dihapus, versi tidak akan lagi terlihat dari Explorer.
5. Ketika satu-satunya versi yang tersisa untuk kebijakan adalah default, maka Anda dapat memilih konteks (klik kanan) kebijakan induk dan memilihHapusuntuk menghapusnya.

## Bekerja denganAWS LambdaFungsi

ParameterAWS Toolkit for Visual Studio Codemenyediakan dukungan untuk[AWS Lambda](#)fungsi. Dengan menggunakan Toolkit for VS Code, Anda dapat membuat kode untuk fungsi Lambda yang merupakan bagian dari[aplikasi nirserver](#). Selain itu, Anda dapat memanggil fungsi Lambda baik secara lokal atau pada AWS.

Lambda adalah layanan komputasi terkelola penuh yang menjalankan kode Anda sebagai respons terhadap peristiwa yang dihasilkan oleh kode kustom atau dari berbagai layanan AWS, seperti Amazon Simple Storage Service (Amazon S3), Amazon DynamoDB, Amazon Kinesis, Amazon Simple Notification Service (Amazon SNS), dan Amazon Cognito.

### Topik

- [Berinteraksi dengan Fungsi Lambda Jarak Jauh](#)

## Berinteraksi dengan Fungsi Lambda Jarak Jauh

Menggunakan Toolkit for VS Code, Anda dapat berinteraksi dengan [AWS Lambda](#) berfungsi dengan berbagai cara, seperti yang dijelaskan nanti dalam topik ini.

Untuk informasi tentang Lambda, lihat [AWS Lambda Panduan Developer](#) .

### Note

Jika Anda telah membuat fungsi Lambda dengan menggunakan AWS Management Console atau dengan cara lain, Anda dapat memanggilnya dari Toolkit. Untuk membuat fungsi baru (menggunakan VS Code) yang dapat Anda gunakan untuk AWS Lambda, Anda harus terlebih dahulu [membuat aplikasi nirserver](#).

### Topik

- [Prasyarat](#)
- [Memanggil Fungsi Lambda](#)
- [Menghapus Fungsi Lambda](#)
- [Mengimpor Fungsi Lambda](#)
- [Unggah Fungsi Lambda](#)

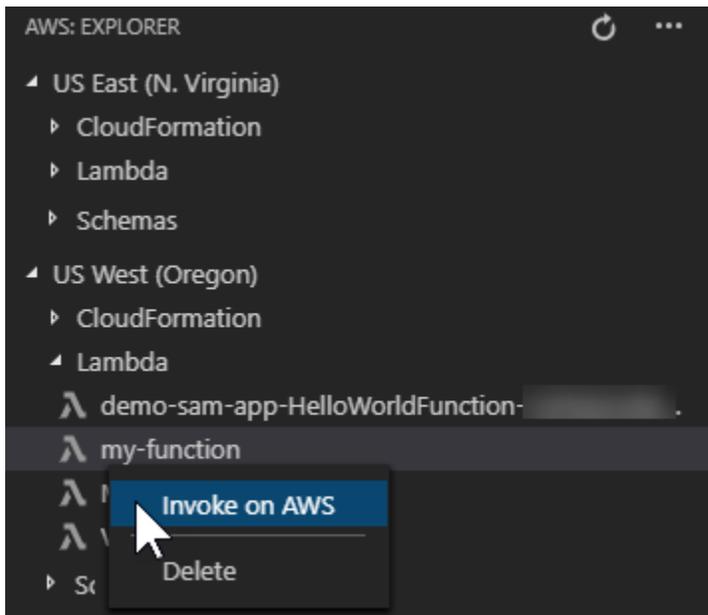
### Prasyarat

- Pastikan sistem Anda memenuhi prasyarat yang ditentukan dalam [Memasang Toolkit for VS Code](#).
- Pastikan kredensi yang Anda konfigurasi [Otentikasi dan akses](#) termasuk akses baca/tulis yang sesuai ke AWS Lambda layanan. Jika di AWS Explorer, di bawah Lambda, Anda melihat pesan yang mirip dengan “Kesalahan memuat sumber daya Lambda”, periksa izin yang dilampirkan pada kredensial tersebut. Perubahan yang Anda buat pada izin akan memakan waktu beberapa menit untuk memengaruhi AWS Penjelajah di VS Code.

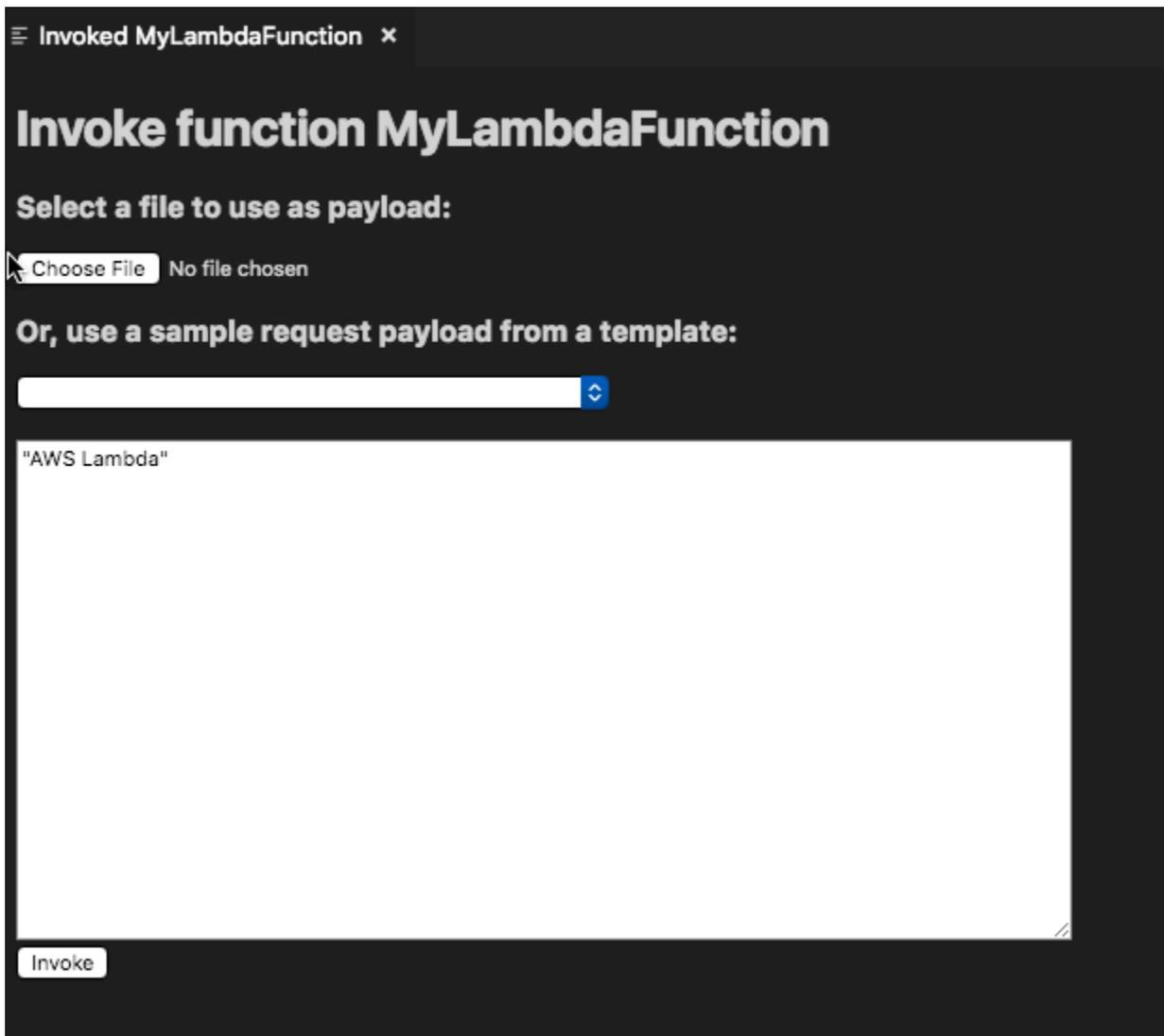
## Memanggil Fungsi Lambda

Anda dapat memanggil fungsi Lambda AWS dari Toolkit for VS Code.

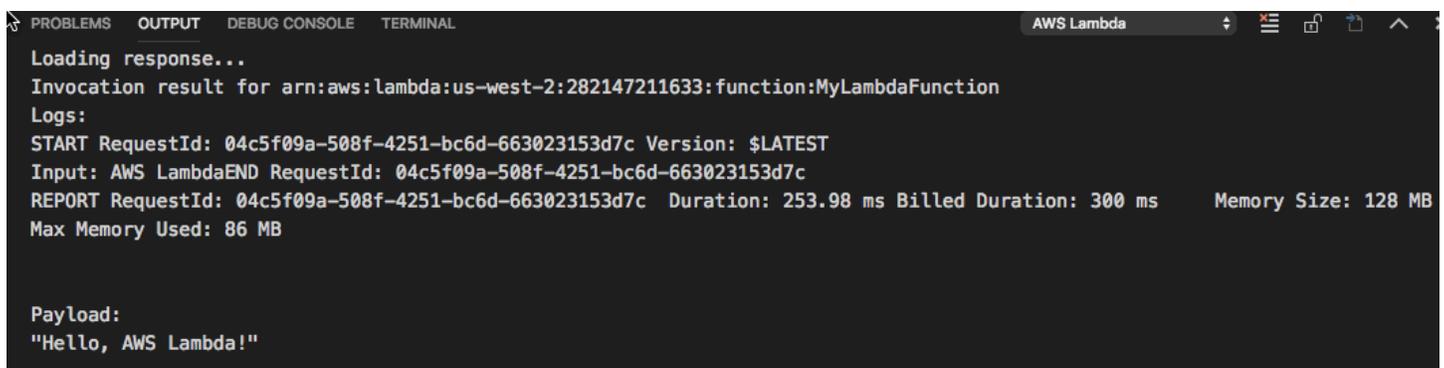
1. Di AWS Explorer, pilih nama fungsi Lambda yang ingin Anda panggil, lalu buka menu konteksnya.



2. Pilih Panggil pada AWS.
3. Di jendela pemanggilan yang terbuka, masukkan input yang dibutuhkan fungsi Lambda Anda. Fungsi Lambda mungkin, misalnya, memerlukan string sebagai input, seperti yang ditunjukkan dalam kotak teks.



Anda akan melihat output dari fungsi Lambda seperti yang Anda lakukan untuk proyek lain yang menggunakan VS Code.



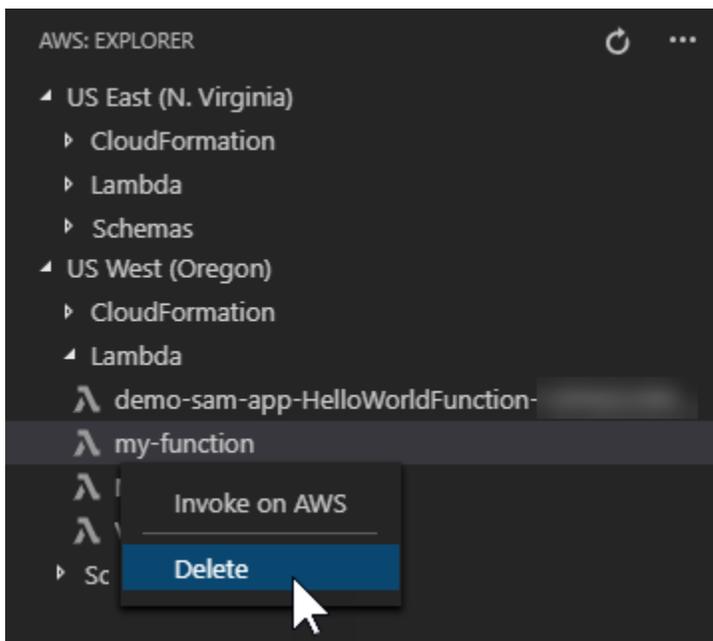
## Menghapus Fungsi Lambda

Anda juga dapat menghapus fungsi Lambda menggunakan menu konteks yang sama.

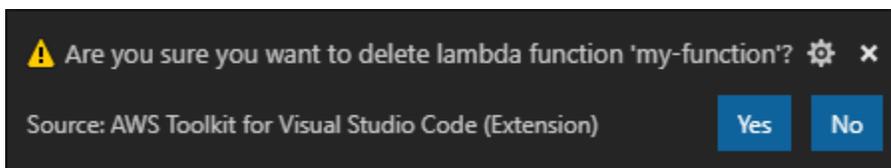
### ⚠ Warning

Jangan gunakan prosedur ini untuk menghapus fungsi Lambda yang terkait dengan [AWS CloudFormation](#) (misalnya, fungsi Lambda yang dibuat ketika [membuat aplikasi nirserver](#) sebelumnya dalam panduan ini). Fungsi-fungsi ini harus dihapus melalui tumpukan AWS CloudFormation.

1. Di AWS Explorer, pilih nama fungsi Lambda yang ingin Anda hapus, lalu buka menu konteksnya.



2. Pilih Delete (Hapus).
3. Pada pesan yang muncul, pilih Ya untuk mengonfirmasi penghapusan.



Setelah fungsi tersebut dihapus, fungsi tersebut tidak lagi tercantum dalam AWS Penjelajah.

## Mengimpor Fungsi Lambda

Anda dapat mengimpor kode dari fungsi Lambda jarak jauh ke ruang kerja VS Code Anda untuk mengedit dan debugging.

### Note

Toolkit hanya mendukung mengimpor fungsi Lambda menggunakan runtime Node.js dan Python yang didukung.

1. DiAWSPenjelajah, pilih nama fungsi Lambda yang ingin Anda impor, lalu buka menu konteksnya.
2. MemilihImpor...
3. Pilih folder untuk mengimpor kode Lambda. Folder di luar ruang kerja saat ini akan ditambahkan ke ruang kerja Anda saat ini.
4. Setelah diunduh, Toolkit menambahkan kode ke ruang kerja Anda dan membuka file yang berisi kode handler Lambda. Toolkit juga membuat konfigurasi peluncuran, yang muncul di panel run VS Code sehingga Anda dapat secara lokal menjalankan dan men-debug fungsi Lambda menggunakan AWS Serverless Application Model. Untuk informasi selengkapnya tentang penggunaan AWS SAM, lihat [the section called “Menjalankan dan men-debug aplikasi tanpa server dari template \(lokal\)”](#).

## Unggah Fungsi Lambda

Anda dapat memperbarui fungsi Lambda yang ada dengan kode lokal. Memperbarui kode dengan cara ini tidak menggunakan AWS SAM CLI untuk deployment dan tidak membuat tumpukan AWS CloudFormation. Fungsionalitas ini dapat mengunggah fungsi Lambda dengan waktu proses apa pun yang didukung oleh Lambda.

### Warning

Toolkit tidak dapat memeriksa apakah kode Anda bekerja. Pastikan kode bekerja sebelum memperbarui fungsi produksi Lambda.

1. DiAWSPenjelajah, pilih nama fungsi Lambda yang ingin Anda impor, lalu buka menu konteksnya.

2. Pilih Unggah Lambda...
3. Pilih dari tiga opsi untuk mengunggah fungsi Lambda Anda. Pilihannya meliputi:

Unggah arsip .zip yang sudah jadi

- MemiilihArsip Zipdari menu Quick Pick.
- Pilih file .zip dari sistem file Anda dan konfirmasi unggahan dengan dialog modal. Ini mengunggah file .zip apa adanya dan segera memperbarui Lambda setelah deployment.

Unggah direktori apa adanya

- MemiilihDirektoridari menu Quick Pick.
- Memiilih direktori dari sistem file Anda.
- MemiilihTidaksaat diminta untuk membangun direktori, lalu konfirmasi unggahan dengan dialog modal. Ini mengunggah direktori apa adanya dan segera memperbarui Lambda setelah deployment.

Bangun dan unggah direktori

 Note

Hal ini membutuhkanAWS SAMCLI.

- MemiilihDirektoridari menu Quick Pick.
- Memiilih direktori dari sistem file Anda.
- Memiilihya saat diminta untuk membangun direktori, lalu konfirmasi unggahan dengan dialog modal. Ini membuat kode di direktori menggunakan perintah AWS SAM CLI `aws sam build` dan segera memperbarui Lambda setelah deployment.

 Note

Toolkit akan memperingatkan Anda jika tidak dapat mendeteksi penanganan yang cocok sebelum mengunggah. Jika Anda ingin mengubah handler terkait dengan fungsi Lambda, Anda dapat melakukannya melaluiAWS Management ConsoleatauAWS CLI.

# Amazon Redshift di Toolkit untuk VS Code

Amazon Redshift adalah layanan gudang data dengan skala petabyte yang dikelola penuh di cloud. Untuk informasi mendetail tentang layanan Amazon Redshift, lihat daftar isi Panduan Pengguna [Amazon Redshift](#).

Topik berikut menjelaskan cara bekerja dengan Amazon Redshift dari AWS Toolkit for Visual Studio Code

Topik

- [Bekerja dengan Amazon Redshift dari Toolkit for VS Code](#)

## Bekerja dengan Amazon Redshift dari Toolkit for VS Code

Bagian berikut menjelaskan cara mulai bekerja dengan Amazon Redshift dari AWS Toolkit for Visual Studio Code

Untuk informasi mendetail tentang layanan Amazon Redshift, lihat topik Panduan Pengguna [Amazon Redshift](#).

Mulai

Sebelum Anda mulai bekerja dengan Amazon Redshift dari AWS Toolkit for Visual Studio Code, persyaratan berikut harus dipenuhi.

1. Anda terhubung ke AWS akun Anda dari Toolkit. Untuk informasi tambahan tentang menghubungkan ke AWS akun Anda dari Toolkit, lihat AWS topik [Menyambung ke](#) dalam Panduan Pengguna ini.
2. Anda telah membuat gudang data yang disediakan atau tanpa server.

Jika Anda belum membuat Amazon Redshift Serverless atau klaster yang disediakan Amazon Redshift, prosedur berikut menjelaskan cara membuat gudang data dengan kumpulan data sampel, dari Konsol. AWS

Membuat gudang data yang disediakan

Untuk detail tambahan tentang membuat gudang data klaster yang disediakan Amazon Redshift, lihat topik [Membuat contoh klaster Amazon Redshift di](#) Panduan Pengguna memulai Amazon Redshift.

1. [Dari browser internet pilihan Anda, masuk ke Konsol AWS Manajemen dan buka konsol Amazon Redshift di https://console.aws.amazon.com/redshift/.](https://console.aws.amazon.com/redshift/)
2. Dari konsol Amazon Redshift, pilih dasbor Provisioned Clusters.
3. Dari dashboard Provisioned Clusters, pilih tombol Create cluster untuk membuka panel Create cluster.
4. Lengkapi bidang wajib di bagian konfigurasi Cluster.
5. Di bagian Data sampel, pilih Muat data sampel kotak untuk memuat kumpulan data sampel **Tickit** ke dalam database default **Dev** dengan **public** skema.
6. Di bagian Konfigurasi basis data, masukkan nilai untuk bidang nama pengguna Admin dan kata sandi pengguna Admin.
7. Pilih Buat klaster untuk membuat gudang data yang disediakan.

### Membuat gudang data tanpa server

Untuk detail tambahan tentang membuat gudang data Amazon Redshift Tanpa Server, lihat bagian [Membuat gudang data dengan Amazon Redshift Serverless di Panduan Pengguna Amazon Redshift](#) memulai.

1. [Dari browser internet pilihan Anda, masuk ke Konsol AWS Manajemen dan buka konsol Amazon Redshift di https://console.aws.amazon.com/redshift/.](https://console.aws.amazon.com/redshift/)
2. Dari konsol Amazon Redshift, pilih tombol Coba Amazon Redshift Tanpa Server untuk membuka panel Memulai dengan Amazon Redshift Serverless.
3. Di bagian Konfigurasi, pilih Gunakan pengaturan default radial.
4. Di bagian bawah panel Memulai dengan Amazon Redshift Tanpa Server, pilih Simpan konfigurasi untuk membuat gudang data tanpa server dengan pengaturan workgroup, namespace, credential, dan enkripsi default.

### Menghubungkan ke gudang data dari Toolkit

Ada 3 metode untuk terhubung ke database dari Toolkit:

- Nama pengguna dan kata sandi basis data
- AWS Secrets Manager
- Kredensi sementara

Untuk menyambung ke database yang terletak di kluster yang sudah ada atau gudang data tanpa server dari Toolkit, selesaikan langkah-langkah berikut.

 Important

Jika Anda telah menyelesaikan langkah-langkah di bagian Prasyarat topik Panduan Pengguna ini dan gudang data Anda tidak terlihat di penjelajah Toolkit, pastikan Anda bekerja dari wilayah yang benar AWS di explorer.

Menghubungkan ke gudang data Anda dengan metode nama pengguna dan kata sandi Database

1. Dari penjelajah Toolkit, perluas Wilayah AWS tempat gudang data Anda berada.
2. Perluas Redshift dan pilih gudang data Anda untuk membuka dialog Select a Connection Type di VS Code.
3. Dari dialog Pilih Jenis Koneksi, pilih nama pengguna dan kata sandi basis data dan berikan informasi yang diperlukan oleh masing-masing petunjuknya.
4. Database, tabel, dan skema Anda yang tersedia terlihat di penjelajah Toolkit saat Toolkit terhubung ke gudang data Anda dan prosedurnya selesai.

Menghubungkan ke gudang data Anda dengan AWS Secrets Manager

 Note

Prosedur ini membutuhkan AWS rahasia database manajer rahasia untuk diselesaikan. Untuk petunjuk tentang cara mengatur rahasia database, lihat [Membuat rahasia AWS Secrets Manager database](#) di Panduan Pengguna AWS Secrets Manager.

1. Dari penjelajah Toolkit, perluas Wilayah AWS tempat gudang data Anda berada.
2. Perluas Redshift dan pilih gudang data Anda untuk membuka dialog Select a Connection Type di VS Code.
3. Dari dialog Select a Connection Type, pilih Secrets Manager dan berikan informasi yang diperlukan oleh setiap petunjuknya.
4. Database, tabel, dan skema Anda yang tersedia terlihat di penjelajah Toolkit saat Toolkit terhubung ke gudang data Anda dan prosedurnya selesai.

## Menghubungkan ke gudang data Anda dengan kredensi Sementara

1. Dari penjelajah Toolkit, perluas AWS wilayah tempat gudang data Anda berada.
2. Perluas Redshift dan pilih gudang data Anda untuk membuka dialog Select a Connection Type di VS Code.
3. Dari dialog Pilih Jenis Koneksi, pilih Kredensi sementara dan berikan informasi yang diperlukan oleh setiap petunjuknya.
4. Database, tabel, dan skema Anda yang tersedia terlihat di penjelajah Toolkit saat Toolkit terhubung ke gudang data Anda dan prosedurnya selesai.

## Mengedit koneksi ke gudang data Anda

Anda dapat mengedit koneksi ke gudang data Anda untuk mengubah database mana yang akan terhubung.

1. Dari penjelajah Toolkit, perluas Wilayah AWS tempat gudang data Anda berada.
2. Perluas Redshift, klik kanan gudang data yang terhubung dengan Anda, pilih Edit koneksi, dan berikan nama database yang ingin Anda sambungkan.
3. Database, tabel, dan skema Anda yang tersedia terlihat di penjelajah Toolkit saat Toolkit terhubung ke gudang data Anda dan prosedurnya selesai.

## Menghapus koneksi ke gudang data Anda

1. Dari penjelajah Toolkit, perluas Wilayah AWS tempat gudang data Anda berada.
2. Perluas Redshift, klik kanan gudang data dengan koneksi yang ingin Anda hapus, dan pilih Hapus koneksi. Melakukannya menghapus database, tabel, dan skema yang tersedia dari penjelajah Toolkit.
3. Untuk menyambung kembali ke gudang data Anda, pilih Klik untuk menghubungkan dan memberikan informasi yang diperlukan oleh setiap petunjuk. Secara default, menghubungkan kembali menggunakan metode otentikasi sebelumnya untuk terhubung ke gudang data. Untuk menggunakan metode yang berbeda, pilih panah kembali dalam dialog hingga Anda mencapai prompt otentikasi.

## Menjalankan Pernyataan SQL

Prosedur berikut menjelaskan cara membuat dan menjalankan pernyataan SQL dalam database Anda dari AWS Toolkit for Visual Studio Code

### Note

Untuk menyelesaikan langkah-langkah di setiap prosedur berikut, Anda harus terlebih dahulu menyelesaikan bagian Menghubungkan ke gudang data dari Toolkit, yang terletak di topik Panduan Pengguna ini.

1. Dari penjelajah Toolkit, perluas Redshift, lalu perluas gudang data yang berisi database yang ingin Anda kueri.
2. Pilih Create-Notebook untuk menentukan nama file dan lokasi untuk menyimpan buku catatan Anda secara lokal, lalu pilih OK untuk membuka buku catatan di editor VS Code Anda.
3. Dari editor VS Code, masukkan pernyataan SQL yang ingin Anda simpan di buku catatan ini.
4. Pilih tombol Run All untuk menjalankan pernyataan SQL yang Anda masukkan.
5. Output untuk pernyataan SQL Anda ditampilkan di bawah pernyataan yang Anda masukkan.

### Menambahkan Markdown ke notebook

1. Dari buku catatan Anda di editor VS Code, pilih tombol Markdown untuk menambahkan sel Markdown ke buku catatan Anda.
2. Masukkan Markdown Anda ke dalam sel yang disediakan.
3. Sel Markdown dapat diedit menggunakan alat editor yang terletak di sudut kanan atas sel Markdown.

### Menambahkan kode ke buku catatan

1. Dari buku catatan Anda di editor Kode VS, pilih tombol Kode untuk menambahkan sel Kode ke buku catatan Anda.
2. Masukkan kode Anda ke dalam sel yang disediakan.
3. Anda dapat memilih untuk menjalankan kode Anda di atas atau di bawah sel Kode dengan memilih tombol yang sesuai dari alat editor sel, yang terletak di sudut kanan atas sel Kode.

# Bekerja dengan Amazon S3

Amazon Simple Storage Service (Amazon S3) adalah layanan penyimpanan data yang dapat diskalakan. AWS Toolkit for Visual Studio Code ini memungkinkan Anda mengelola objek dan sumber daya Amazon S3 langsung dari VS Code.

Untuk informasi terperinci tentang layanan Amazon S3, lihat Panduan Pengguna [Amazon S3](#).

Topik berikut menjelaskan cara bekerja dengan objek dan sumber daya Amazon S3 dari AWS Toolkit for Visual Studio Code

Topik

- [Bekerja dengan sumber daya Amazon S3](#)
- [Cara menggunakan objek Amazon S3](#)

## Bekerja dengan sumber daya Amazon S3

Anda dapat menggunakan Amazon S3 dari AWS Toolkit for Visual Studio Code untuk melihat, mengelola, dan mengedit bucket Amazon S3 dan sumber daya lainnya.

Bagian berikut menjelaskan cara bekerja dengan sumber daya Amazon S3 dari AWS Toolkit for Visual Studio Code Untuk informasi tentang bekerja dengan objek Amazon S3, seperti folder dan file, dari AWS Toolkit for Visual Studio Code, lihat topik [Bekerja dengan objek S3](#) dalam Panduan Pengguna ini.

## Membuat sebuah bucket Amazon S3

1. Dari penjelajah Toolkit, buka menu konteks (klik kanan) untuk layanan S3, dan pilih Buat Bucket... . Atau, pilih ikon Create Bucket untuk membuka kotak dialog Create Bucket.
2. Di bidang Nama Bucket, masukkan nama yang valid untuk bucket.

Tekan Enter untuk membuat ember dan menutup kotak dialog. Bucket baru Anda kemudian ditampilkan di bawah layanan S3 di toolkit.

### Note

Karena Amazon S3 memungkinkan bucket Anda digunakan sebagai URL yang dapat diakses publik, nama bucket yang Anda pilih harus unik secara global. Jika akun

lain sudah membuat bucket dengan nama yang ingin Anda gunakan, Anda harus menggunakan nama yang berbeda.

Jika Anda tidak dapat membuat bucket baru, periksa AWS Toolkit Logs di tab Output. Jika Anda mencoba menggunakan nama bucket yang tidak valid, `BucketAlreadyExists` terjadi kesalahan.

Untuk informasi selengkapnya, lihat [Pembatasan dan batasan bucket](#) di Panduan Pengguna Layanan Penyimpanan Sederhana Amazon.

## Menambahkan folder ke bucket Amazon S3

Anda dapat mengatur konten bucket S3 dengan mengelompokkan objek Anda ke dalam folder. Anda juga dapat membuat folder di dalam folder.

1. Dari penjelajah Toolkit, perluas layanan S3 untuk melihat daftar sumber daya S3 Anda.
2. Pilih ikon Buat Folder untuk membuka kotak dialog Buat Folder. Atau, buka menu konteks (klik kanan) untuk ember atau folder, lalu pilih Buat Folder.
3. Masukkan nilai ke bidang Nama Folder dan tekan Enter untuk membuat folder dan menutup kotak dialog. Folder baru Anda ditampilkan di bawah sumber daya S3 yang sesuai di menu toolkit.

## Menghapus sebuah bucket Amazon S3

Saat Anda menghapus bucket S3, Anda juga menghapus folder dan objek yang dikandungnya. Jadi, ketika Anda mencoba menghapus bucket, Anda diminta untuk mengonfirmasi bahwa Anda ingin menghapusnya.

1. Dari menu utama toolkit, perluas layanan Amazon S3 untuk melihat daftar sumber daya S3 Anda.
2. Buka menu konteks (klik kanan) untuk bucket atau folder, lalu pilih Delete S3 Bucket.
3. Saat diminta, masukkan nama bucket ke dalam kolom teks, lalu tekan Enter untuk menghapus bucket dan menutup prompt konfirmasi.

### Note

Jika bucket berisi objek, bucket akan dikosongkan sebelum dihapus. Jika Anda mencoba untuk menghapus sejumlah besar sumber daya atau objek pada satu waktu, itu bisa

memakan waktu untuk dihapus. Setelah dihapus, Anda menerima pemberitahuan yang mengatakan bahwa mereka berhasil dihapus.

## Cara menggunakan objek Amazon S3

File, folder, dan data lain yang disimpan dalam bucket sumber daya S3 dikenal sebagai objek S3.

Bagian berikut menjelaskan cara bekerja dengan objek Amazon S3 dari file. AWS Toolkit for Visual Studio Code Untuk detail tentang bekerja dengan sumber daya Amazon S3, seperti bucket S3, dari AWS Toolkit for Visual Studio Code, lihat topik [Bekerja dengan sumber daya S3](#) di Panduan Pengguna ini.

### Pagination objek

Jika Anda bekerja dengan sejumlah besar objek dan folder Amazon S3, pagination memungkinkan Anda menentukan jumlah item yang ingin ditampilkan di halaman.

1. Arahkan ke VS Code Activity Bar dan pilih Extensions.
2. Dari ekstensi AWS Toolkit, pilih ikon pengaturan, lalu pilih Pengaturan Ekstensi.
3. Pada halaman Pengaturan, gulir ke bawah ke pengaturan AWS> S3: Item Maks Per Halaman.
4. Ubah nilai default ke jumlah item S3 yang ingin Anda tampilkan sebelum “muat lebih banyak” ditampilkan.

#### Note

Nilai yang valid mencakup angka apa pun antara 3 dan 1000. Pengaturan ini hanya berlaku untuk jumlah objek atau folder yang ditampilkan pada satu waktu. Semua ember yang Anda buat ditampilkan sekaligus. Secara default, Anda dapat membuat hingga 100 bucket di setiap AWS akun Anda.

5. Tutup halaman Pengaturan untuk mengonfirmasi perubahan Anda.

Anda juga dapat memperbarui pengaturan dalam file berformat JSON dengan memilih ikon Open Settings (JSON) di kanan atas halaman Pengaturan.

## Mengunggah dan mengunduh objek Amazon S3

Anda dapat mengunggah file yang disimpan secara lokal ke bucket Amazon S3 Anda atau mengunduh objek Amazon S3 jarak jauh ke sistem lokal Anda, dari file. AWS Toolkit for Visual Studio Code

Unggah file menggunakan Toolkit

1. Dari penjelajah Toolkit, perluas layanan Amazon S3 untuk melihat daftar sumber daya S3 Anda.
2. Pilih ikon Upload File yang terletak di sebelah bucket atau folder untuk membuka dialog Upload File. Atau Anda dapat membuka menu konteks (klik kanan) dan memilih Unggah File.

### Note

Untuk mengunggah file ke folder atau sumber daya induk objek, buka menu konteks (klik kanan) untuk objek S3 apa pun dan pilih Unggah ke Induk.

3. Gunakan pengelola file sistem Anda untuk memilih file, lalu pilih Unggah File untuk menutup dialog dan mengunggah file.

Unggah file menggunakan Command Palette

Anda dapat menggunakan antarmuka Toolkit atau Command Palette untuk mengunggah file ke bucket.

1. Untuk memilih file untuk diunggah, pilih tab file itu di VS Code.
2. Tekan Ctrl+Shift+P untuk menampilkan Command Palette.
3. Di Command Palette, masukkan frasa `upload file` untuk menampilkan daftar perintah yang direkomendasikan.
4. Pilih perintah AWS: Unggah File untuk membuka dialog AWS: Unggah File.
5. Saat diminta, pilih file yang ingin Anda unggah, lalu pilih bucket tempat Anda ingin mengunggah file tersebut.
6. Konfirmasikan unggahan Anda untuk menutup dialog dan memulai proses pengunggahan. Saat unggahan selesai, objek ditampilkan di menu toolkit dengan metadata yang mencakup ukuran objek, tanggal modifikasi terakhir, dan jalur.

## Mengunduh objek Amazon S3

1. Dari penjelajah Toolkit, perluas layanan S3.
2. Dari bucket atau folder, buka menu konteks (klik kanan) untuk objek yang ingin Anda unduh. Kemudian, pilih Unduh Sebagai untuk membuka kotak dialog Unduh Sebagai. Atau, sebagai alternatif, pilih ikon Unduh Sebagai di dekat objek.
3. Menggunakan pengelola file sistem Anda, pilih folder tujuan, masukkan nama file, lalu pilih Unduh untuk menutup dialog dan mulai mengunduh.

## Mengedit objek jarak jauh

Anda dapat menggunakan AWS Toolkit for Visual Studio Code untuk mengedit objek Amazon S3 Anda yang disimpan di sumber daya Amazon S3 jarak jauh Anda.

1. Dari penjelajah Toolkit, perluas layanan S3.
2. Perluas sumber daya S3 yang berisi file yang ingin Anda edit.
3. Untuk mengedit file, pilih ikon pensil (Edit File).
4. Untuk mengedit file yang terbuka dalam mode hanya-baca, lihat file di editor VS Code, lalu pilih ikon pensil yang terletak di sudut kanan atas UI.

### Note

- Jika Anda memulai ulang atau keluar dari VS Code, IDE Anda terputus dari sumber daya S3 Anda. Jika ada file S3 jarak jauh yang sedang diedit saat Anda memutuskan sambungan, pengeditan akan berhenti. Anda harus memulai ulang VS Code dan membuka kembali tab edit untuk melanjutkan pengeditan.
- Tombol Edit File ada di sudut kanan atas UI. Ini hanya terlihat saat Anda secara aktif melihat file hanya-baca di editor VS Code.
- File non-teks tidak dapat dibuka dalam mode hanya-baca. Mereka selalu terbuka dalam mode edit.
- Anda tidak dapat beralih kembali ke mode read-only dari mode edit-only, hanya sebaliknya.

## Menyalin jalur objek Amazon S3

Prosedur berikut menjelaskan cara menyalin jalur objek Amazon S3 dari file. AWS Toolkit for Visual Studio Code

1. Dari penjelajah Toolkit, perluas layanan S3.
2. Perluas bucket sumber daya yang berisi objek yang ingin Anda salin jalurnya.
3. Buka menu konteks (klik kanan) untuk objek yang ingin Anda salin jalurnya, lalu pilih Salin Jalur untuk menyalin jalur objek ke clipboard lokal Anda.

## Membuat URL yang telah ditetapkan sebelumnya untuk objek Amazon S3

Anda dapat berbagi objek Amazon S3 pribadi dengan orang lain dengan memberikan izin terbatas waktu untuk mengunduh melalui fitur URL yang telah ditentukan sebelumnya. Untuk informasi selengkapnya, lihat [Berbagi objek dengan URL yang telah ditetapkan sebelumnya](#).

1. Dari penjelajah Toolkit, perluas layanan S3.
2. Dari bucket atau folder, buka menu konteks (klik kanan) untuk objek yang ingin Anda bagikan. Kemudian, pilih Generate Presigned URL untuk membuka palet Command.
3. Dari Command Palette, masukkan jumlah menit URL yang dapat digunakan untuk mengakses objek Anda. Kemudian, pilih Enter untuk mengonfirmasi dan menutup dialog.
4. Setelah URL presigned dibuat, VS Code Status Bar menampilkan URL presigned untuk objek yang telah disalin ke clipboard lokal Anda.

## Menghapus objek Amazon S3

Jika objek berada dalam bucket yang tidak berversi, Anda dapat menghapusnya secara permanen. Untuk bucket yang mengaktifkan versi, permintaan penghapusan tidak akan menghapus objek tersebut secara permanen. Sebaliknya, Amazon S3 menyisipkan delete marker dalam bucket. Untuk informasi selengkapnya, lihat [Menghapus versi objek](#).

1. Dari penjelajah Toolkit, perluas layanan S3 untuk melihat daftar sumber daya S3 Anda.
2. Buka menu konteks (klik kanan) untuk objek yang ingin Anda hapus, lalu pilih Hapus untuk membuka dialog konfirmasi.
3. Pilih Hapus. untuk mengonfirmasi bahwa Anda ingin menghapus objek S3. Kemudian, tutup dialog.

# Bekerja dengan irserver

AWS Toolkit for Visual Studio Code menyediakan dukungan untuk [AWS Serverless Application](#). Topik berikut menjelaskan bagaimana untuk memulai membuat dan bekerja dengan AWS Serverless Application Model (AWS SAM) aplikasi, dari AWS Toolkit for Visual Studio Code.

Topik

- [Memulai dengan aplikasi tanpa server](#)
- [Menjalankan dan men-debug fungsi Lambda sambil AWS SAM mengecualikan sumber daya template](#)
- [Menjalankan dan men-debug sumber daya Amazon API Gateway lokal](#)
- [Opsi konfigurasi untuk men-debug aplikasi nirserver](#)
- [Pemecahan masalah aplikasi nirserver](#)

## Memulai dengan aplikasi tanpa server

Bagian berikut menjelaskan cara memulai membuat AWS Serverless Application dari AWS Toolkit for Visual Studio Code, menggunakan AWS Serverless Application Model (AWS SAM) dan AWS CloudFormation tumpukan.

### Prasyarat

Sebelum Anda dapat membuat atau bekerja dengan AWS Serverless Application, prasyarat berikut harus diselesaikan.

#### Note

Operasi berikut mungkin mengharuskan Anda untuk keluar atau memulai ulang VS Code sebelum perubahan selesai.

- Instal antarmuka baris AWS SAM perintah (CLI). Untuk informasi dan petunjuk tambahan tentang cara menginstal AWS SAM CLI, lihat topik [Memasang AWS SAM CLI](#) di Panduan Pengguna ini. [AWS Serverless Application Model](#)
- Dari file AWS konfigurasi Anda, identifikasi AWS Region default Anda. Untuk informasi selengkapnya tentang file konfigurasi Anda, lihat topik [pengaturan file konfigurasi dan kredensi](#) di [AWS Command Line Interface](#) Panduan Pengguna.

- Instal SDK bahasa Anda dan konfigurasi rantai alat Anda. Untuk informasi tambahan tentang cara mengkonfigurasi toolchain Anda dari AWS Toolkit for Visual Studio Code lihat topik [konfigurasi toolchain Anda](#) di Panduan Pengguna ini.
- Instal [ekstensi dukungan bahasa YAMB](#) dari pasar VS Code. Ini diperlukan agar CodeLens fitur file AWS SAM template dapat diakses. Untuk informasi tambahan tentang CodeLens, lihat [CodeLens](#) bagian dalam dokumentasi VS Code

## Izin IAM untuk aplikasi tanpa server

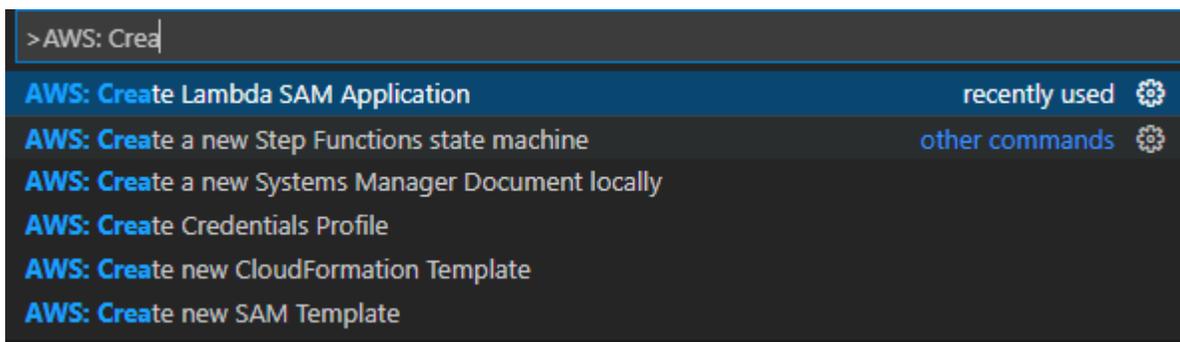
Dalam Toolkit for VS Code Anda harus memiliki profil kredensial yang berisi AWS Identity and Access Management izin (IAM) yang diperlukan untuk menyebarkan dan menjalankan aplikasi tanpa server. Anda harus memiliki akses baca/tulis yang sesuai ke layanan berikut: AWS CloudFormation, IAM, Lambda, Amazon API Gateway, Amazon Simple Storage Service (Amazon S3), dan Amazon Elastic Container Registry (Amazon ECR).

Untuk informasi tambahan tentang menyiapkan autentikasi yang diperlukan untuk menyebarkan dan menjalankan aplikasi tanpa server, lihat [Mengelola akses dan izin sumber daya](#) di Panduan Pengembang. AWS Serverless Application Model Untuk informasi tentang cara mengatur kredensial Anda, lihat [AWS IAM kredensialnya](#) di Panduan Pengguna ini.

## Membuat aplikasi tanpa server baru (lokal)

Prosedur ini menunjukkan cara membuat aplikasi tanpa server dengan Toolkit for VS Code dengan menggunakan AWS SAM Output dari prosedur ini adalah direktori lokal di host pengembangan Anda yang berisi contoh aplikasi tanpa server, yang dapat Anda buat, uji, modifikasi, dan terapkan secara lokal ke Cloud. AWS

1. Untuk membuka Command Palette, pilih View, Command Palette, dan kemudian enter AWS.
2. Pilih AWSToolkit Buat Aplikasi Lambda SAM.



**Note**

Jika AWS SAM CLI tidak diinstal, Anda mendapatkan kesalahan di sudut kanan bawah editor VS Code. Jika ini terjadi, verifikasi bahwa Anda telah memenuhi semua [asumsi](#) dan [prasyarat](#).

3. Pilih runtime untuk AWS SAM aplikasi Anda.

**Note**

Jika Anda memilih salah satu runtime dengan “(Image)”, aplikasi Anda adalah tipe paket Image. Jika Anda memilih salah satu runtime tanpa “(Image)”, aplikasi Anda adalah tipe Zip. Untuk informasi selengkapnya tentang perbedaan antara jenis paket Image dan Zip, lihat [paket deployment Lambda](#) dalam Panduan Developer AWS Lambda.

4. Bergantung pada runtime yang Anda pilih, Anda mungkin diminta untuk memilih manajer ketergantungan dan arsitektur runtime untuk aplikasi SAM Anda.

### Dependency Manager

Pilih antara Gradle atau Maven.

**Note**

Pilihan alat otomatisasi build ini hanya tersedia untuk runtime Java.

### Architecture

Pilih antara x86\_64 atau arm64.

Opsi untuk menjalankan aplikasi tanpa server Anda di lingkungan emulasi berbasis ARM64 alih-alih lingkungan berbasis x86\_64 default tersedia untuk runtime berikut:

- nodejs12.x (ZIP dan gambar)
- nodejs14.x (ZIP dan gambar)
- python3.8 (ZIP dan gambar)
- python3.9 (ZIP dan gambar)

- python3.10 (ZIP dan gambar)
- python3.11 (ZIP dan gambar)
- python3.12 (ZIP dan gambar)
- java8.al2 dengan Gradle (ZIP dan gambar)
- java8.al2 dengan Maven (hanya ZIP)
- java11 dengan Gradle (ZIP dan gambar)
- java11 dengan Maven (hanya ZIP)

 Important

Anda harus menginstal AWS CLI versi 1.33.0 atau yang lebih baru untuk memungkinkan aplikasi berjalan di lingkungan berbasis ARM64. Untuk informasi selengkapnya, lihat [Prasyarat](#).

5. Pilih lokasi untuk proyek baru Anda. Anda dapat menggunakan folder ruang kerja yang ada jika ada yang terbuka, Pilih folder lain yang sudah ada, atau buat folder baru dan pilih. Untuk contoh ini, pilih Tidak ada folder ruang kerja yang terbuka untuk membuat folder bernamaMY-SAM-APP.
6. Masukkan nama untuk proyek baru Anda. Untuk contoh ini, gunakan my-sam-app-nodejs. Setelah Anda menekan Enter, Toolkit for VS Code membutuhkan beberapa saat untuk membuat proyek.

Ketika proyek dibuat, aplikasi Anda ditambahkan ke ruang kerja Anda saat ini. Anda akan melihatnya tercantum di jendela Explorer.

## Membuka aplikasi tanpa server (lokal)

Untuk membuka aplikasi tanpa server di host pengembangan lokal Anda, buka folder yang berisi file template aplikasi.

1. Dari File, pilih Open Folder... .
2. Di kotak dialog Open Folder, arahkan ke folder aplikasi tanpa server yang ingin Anda buka.
3. Pilih tombol Pilih Folder.

Ketika Anda membuka folder aplikasi, itu ditambahkan ke jendela Explorer.

## Menjalankan dan men-debug aplikasi tanpa server dari template (lokal)

Anda dapat menggunakan Toolkit for VS Code untuk mengonfigurasi cara men-debug aplikasi tanpa server dan menjalankannya secara lokal di lingkungan pengembangan Anda.

Anda mulai mengonfigurasi perilaku debug dengan menggunakan [CodeLens](#) fitur Kode VS untuk mengidentifikasi fungsi Lambda yang memenuhi syarat. CodeLens memungkinkan interaksi sadar konten dengan kode sumber Anda. Untuk informasi tentang memastikan bahwa Anda dapat mengakses CodeLens fitur, tinjau [Prasyarat](#) bagian dari sebelumnya dalam topik ini.

### Note

Dalam contoh ini, Anda men-debug aplikasi yang menggunakan JavaScript. Namun, Anda dapat menggunakan fitur debugging Toolkit for VS Code dengan bahasa dan runtime berikut:

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

Pilihan bahasa Anda juga memengaruhi cara CodeLens mendeteksi penangan Lambda yang memenuhi syarat. Untuk informasi selengkapnya, lihat [Menjalankan dan men-debug fungsi Lambda sambil AWS SAM mengecualikan sumber daya template](#).

Dalam prosedur ini, Anda menggunakan contoh aplikasi yang dibuat di [Membuat aplikasi tanpa server baru \(lokal\)](#) bagian sebelumnya dalam topik ini.

1. Untuk melihat file aplikasi Anda di File Explorer VS Code, pilih View, Explorer.
2. Dari folder aplikasi (misalnya, my-sample-app), buka `template.yaml` file.

### Note

Jika Anda menggunakan templat dengan nama yang berbedatemplate.yaml, CodeLens indikator tidak tersedia secara otomatis di file YAMAL. Ini berarti Anda harus menambahkan konfigurasi debug secara manual.

3. Di editor untuk `template.yaml`, buka Resources bagian template yang mendefinisikan sumber daya tanpa server. Dalam hal ini, ini adalah `HelloWorldFunction` sumber daya tipe `AWS::Serverless::Function`.

Dalam CodeLens indikator untuk sumber daya ini, pilih Tambahkan Konfigurasi Debug.

4. Di Command Palette, pilih runtime di mana AWS SAM aplikasi Anda akan berjalan.
5. Di editor untuk file `launch.json`, edit atau konfirmasi nilai untuk properti konfigurasi berikut:
  - `"name"` – Masukkan nama yang mudah dibaca untuk muncul di bidang tarik-turun Konfigurasi di tampilan Jalankan.
  - `"target"`— Pastikan nilainya `"template"` sedemikian rupa sehingga AWS SAM template adalah titik masuk untuk sesi debug.
  - `"templatePath"` – Masukkan jalur relatif atau absolut untuk file `template.yaml`.
  - `"logicalId"`— Pastikan bahwa nama cocok dengan yang ditentukan di bagian Sumber daya AWS SAM template. Dalam hal ini, ini adalah `HelloWorldFunction` dari tipe `AWS::Serverless::Function`.

Untuk informasi selengkapnya tentang entri ini dan entri lainnya dalam file `launch.json`, lihat [Opsi konfigurasi untuk men-debug aplikasi nirserver](#).

6. Jika Anda puas dengan konfigurasi debug Anda, simpan `launch.json`. Kemudian, untuk memulai debugging, pilih tombol “putar” hijau di tampilan RUN.

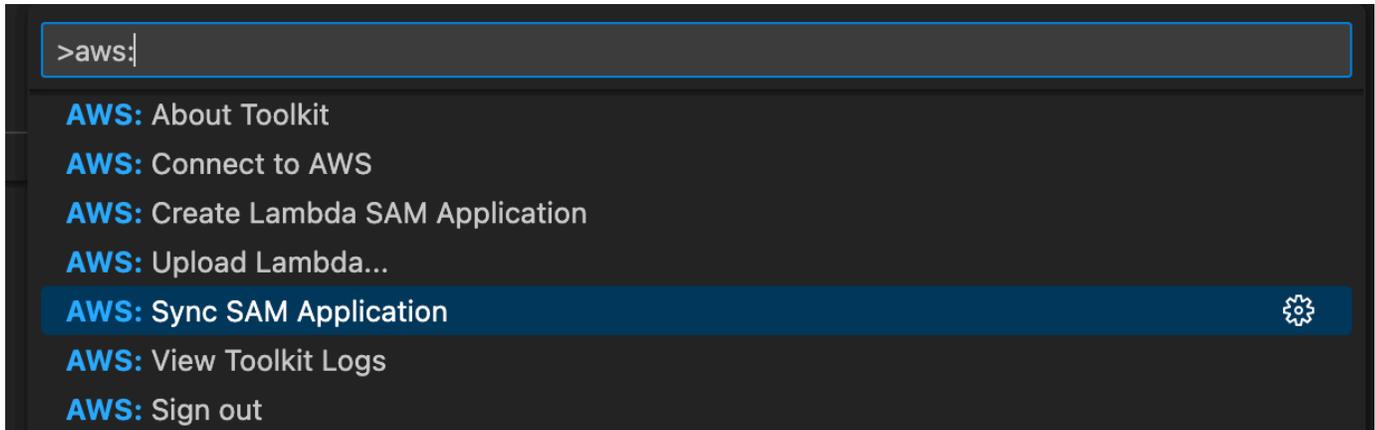
Ketika sesi debug dimulai, panel KONSOL DEBUG menampilkan output debug dan menampilkan nilai apa pun yang dikembalikan oleh fungsi Lambda. (Saat men-debug AWS SAM aplikasi, AWSToolkit dipilih sebagai saluran Output di panel Output.)

## Menyinkronkan aplikasi AWS SAM

AWS Toolkit for Visual Studio Code menjalankan perintah AWS SAM CLI `sam sync` untuk menyebarkan aplikasi tanpa server Anda ke file. AWS Cloud Untuk informasi tambahan tentang AWS SAM sinkronisasi, lihat topik [referensi perintah AWS SAM CLI](#) di Panduan Pengembang AWS Serverless Application Model

Prosedur berikut menjelaskan cara menerapkan aplikasi tanpa server Anda ke AWS Cloud with `sam sync` dari Toolkit for VS Code.

1. Dari menu utama di VS Code, buka Command Palette dengan memperluas View dan memilih Command Palette.
2. Dari Command Palette cari AWS dan pilih Sync SAM Application untuk mulai mengatur sinkronisasi Anda.



3. Pilih AWS Wilayah untuk menyinkronkan aplikasi tanpa server Anda.
4. Pilih file `template.yaml` yang akan digunakan untuk deployment.
5. Pilih bucket Amazon S3 yang sudah ada atau masukkan nama bucket Amazon S3 baru untuk menerapkan aplikasi Anda.

 Important

Bucket Amazon S3 Anda harus memenuhi persyaratan berikut:

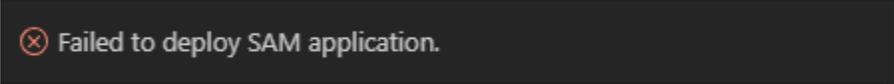
- Bucket harus berada di Wilayah yang Anda sinkronkan.
- Nama bucket Amazon S3 harus unik secara global di semua nama bucket yang ada di Amazon S3.

6. Jika aplikasi nirserver Anda menyertakan fungsi dengan tipe paket Image, masukkan nama repositori Amazon ECR yang dapat digunakan oleh deployment ini. Repositori harus berada di Wilayah tempat Anda men-deploy.
7. Pilih tumpukan penerapan dari daftar penerapan sebelumnya, atau buat tumpukan penerapan baru dengan memasukkan nama tumpukan baru. Kemudian, lanjutkan untuk memulai proses sinkronisasi.

**Note**

Tumpukan yang digunakan dalam penerapan sebelumnya ditarik kembali per ruang kerja dan wilayah.

8. Selama proses sinkronisasi, status penerapan Anda ditangkap di tab Terminal VS Code. Verifikasi bahwa sinkronisasi Anda berhasil dari tab terminal, jika terjadi kesalahan, Anda menerima pemberitahuan.



```
✘ Failed to deploy SAM application.
```

**Note**

Untuk detail tambahan tentang sinkronisasi Anda, AWS Toolkit for Visual Studio Code log dapat diakses dari Command Palette.

Untuk mengakses AWS Toolkit for Visual Studio Code log Anda dari Command Palette, perluas View, pilih Command Palette **AWS: View AWS Toolkits Logs**, lalu cari, dan pilih ketika terisi dalam daftar.

Ketika penerapan selesai, Anda melihat aplikasi Anda terdaftar di AWSExplorer. Untuk informasi selengkapnya tentang cara menjalankan fungsi Lambda yang dibuat sebagai bagian dari aplikasi, lihat topik [Berinteraksi dengan Fungsi Lambda Jarak Jauh](#) di Panduan Pengguna ini.

## Menghapus aplikasi tanpa server dari AWS Cloud

Menghapus aplikasi nirserver melibatkan penghapusan tumpukan AWS CloudFormation yang sebelumnya Anda deploy ke AWS Cloud. Perhatikan bahwa prosedur ini tidak menghapus direktori aplikasi Anda dari host lokal Anda.

1. Buka [AWS Penjelajah](#).
2. Di jendela AWSToolkit Explorer, perluas Wilayah yang berisi aplikasi yang digunakan yang ingin Anda hapus, lalu perluas. AWS CloudFormation

3. Buka menu konteks (klik kanan) untuk nama AWS CloudFormation tumpukan yang sesuai dengan aplikasi tanpa server yang ingin Anda hapus, lalu pilih Hapus Tumpukan. AWS CloudFormation
4. Untuk mengonfirmasi bahwa Anda ingin menghapus tumpukan yang dipilih, pilih Ya.

Jika penghapusan tumpukan berhasil, Toolkit for VS Code menghapus nama tumpukan dari daftar di Explorer. AWS CloudFormation AWS

## Menjalankan dan men-debug fungsi Lambda sambil AWS SAM mengecualikan sumber daya template

Saat menguji AWS SAM aplikasi, Anda dapat memilih untuk menjalankan dan men-debug hanya fungsi Lambda dan mengecualikan sumber daya lain yang didefinisikan template AWS SAM . Pendekatan ini melibatkan penggunaan [CodeLens](#) fitur untuk mengidentifikasi penanganan fungsi Lambda dalam kode sumber yang dapat langsung Anda panggil.

Handler Lambda yang terdeteksi CodeLens bergantung pada bahasa dan runtime yang Anda gunakan untuk aplikasi Anda.

Bahasa/runtime	Kriteria fungsi Lambda harus diidentifikasi oleh indikator CodeLens
C # (dotnetcore2.1, 3.1;. NET5.0)	<p>Fungsi tersebut memiliki fitur berikut:</p> <ul style="list-style-type: none"> <li>• Ini adalah fungsi publik dari kelas publik.</li> <li>• Ini memiliki satu atau dua parameter. Dengan dua parameter, parameter kedua harus mengimplementasikan <code>ILambdaContext</code> antarmuka.</li> <li>• Ini memiliki <code>*.csproj</code> file di folder induknya dalam folder ruang kerja VS Code.</li> </ul> <p>Ekstensi <a href="#">ms-dotnettools.csharp</a> (atau ekstensi apa pun yang menyediakan simbol bahasa untuk C#) diinstal dan diaktifkan.</p>
JavaScript/TypeScript (Node.js 12.x, 14.x)	<p>Fungsi tersebut memiliki fitur berikut:</p>

Bahasa/runtime	Kriteria fungsi Lambda harus diidentifikasi oleh indikator CodeLens
	<ul style="list-style-type: none"><li>• Ini adalah fungsi yang diekspor dengan hingga tiga parameter.</li><li>• Ini memiliki <code>package.json</code> file di folder induknya dalam folder ruang kerja VS Code.</li></ul>
Python (3.7, 3.8, 3.9, 3.10, 3.11, 3.12)	<p>Fungsi tersebut memiliki fitur berikut:</p> <ul style="list-style-type: none"><li>• Ini adalah fungsi tingkat atas.</li><li>• Ini memiliki <code>requirements.txt</code> file di folder induknya dalam folder ruang kerja VS Code.</li></ul> <p>Ekstensi <a href="#">ms-python.python</a> (atau ekstensi apa pun yang menyediakan simbol bahasa untuk Python) diinstal dan diaktifkan.</p>

Bahasa/runtime	Kriteria fungsi Lambda harus diidentifikasi oleh indikator CodeLens
Jawa (8, 8.al2, 11)	<p>Fungsi tersebut memiliki fitur berikut:</p> <ul style="list-style-type: none"><li>• Ini adalah fungsi publik dari kelas publik, non-abstrak.</li><li>• Ini memiliki satu, dua, atau tiga parameter:<ul style="list-style-type: none"><li>• Satu parameter: Parameter bisa apa saja.</li><li>• Dua parameter: Parameter harus a <code>java.io.InputStream</code> dan A <code>java.io.OutputStream</code> ATAU parameter terakhir harus <code>acom.amazonaws.services.lambda.runtime.Context</code> .</li><li>• Tiga parameter: Parameter harus a <code>java.io.InputStream</code> <code>java.io.OutputStream</code> AND dan parameter terakhir harus <code>acom.amazonaws.services.lambda.runtime.Context</code> .</li></ul></li><li>• Ini memiliki file <code>build.gradle</code> (Gradle) atau <code>pom.xml</code> (Maven) di folder induknya dalam folder ruang kerja VS Code.</li></ul> <p><a href="#">Ekstensi redhat.java</a> (atau ekstensi apa pun yang menyediakan simbol bahasa untuk Java) diinstal dan diaktifkan. Ekstensi ini membutuhkan Java 11, tidak peduli runtime Java mana yang Anda gunakan.</p> <p><a href="#">Vscjava. vscode-java-debug</a> ekstensi (atau ekstensi apa pun yang menyediakan debugger Java) diinstal dan diaktifkan.</p>

Bahasa/runtime	Kriteria fungsi Lambda harus diidentifikasi oleh indikator CodeLens
Pergi (1.x)	<p>Fungsi tersebut memiliki fitur berikut:</p> <ul style="list-style-type: none"> <li>• Ini adalah fungsi tingkat atas.</li> <li>• Dibutuhkan antara 0 dan 2 argumen. Jika ada dua argumen, argumen pertama harus menerapkan <code>context.Context</code>.</li> <li>• Ia mengembalikan antara 0 dan 2 argumen. Jika ada lebih dari 0 argumen, argumen terakhir harus diteruskan <code>error</code>.</li> <li>• Ini memiliki <code>go.mod</code> file dalam folder ruang kerja VS Code.</li> </ul> <p><a href="#">Ekstensi <code>golang.go</code></a> diinstal, dikonfigurasi, dan diaktifkan.</p>

Untuk menjalankan dan men-debug aplikasi nirserver langsung dari kode aplikasi

1. Untuk melihat file aplikasi Anda di VS Code File Explorer, pilih View, Explorer.
2. Dari folder aplikasi (misalnya, `my-sample-app`), perluas folder fungsi (dalam hal ini, `hello-world`) dan buka file `app.js`
3. Pada CodeLens indikator yang mengidentifikasi penanganan fungsi Lambda yang memenuhi syarat, pilih `Add Debug Configuration`
4. Di Command Palette, pilih runtime di mana AWS SAM aplikasi Anda akan berjalan.
5. Di editor untuk file `launch.json`, edit atau konfirmasi nilai untuk properti konfigurasi berikut:
  - `"name"`— Masukkan nama yang ramah pembaca untuk muncul di bidang tarik-turun Konfigurasi di tampilan Jalankan.
  - `"target"` – Pastikan bahwa nilainya adalah `"code"` agar penanganan fungsi Lambda dipanggil secara langsung.

- "lambdaHandler" – Masukkan nama metode dalam kode Anda yang dipanggil Lambda untuk memanggil fungsi Anda. Misalnya, untuk aplikasi di JavaScript, defaultnya adalah `app.lambdaHandler`.
- "projectRoot" – Masukkan jalur ke file aplikasi yang berisi fungsi Lambda.
- "runtime" – Masukkan atau konfirmasi runtime yang valid untuk lingkungan eksekusi Lambda, misalnya, "nodejs.12x".
- "payload" – Pilih salah satu opsi berikut untuk menentukan muatan peristiwa yang ingin Anda berikan ke fungsi Lambda Anda sebagai input:
  - "json": JSON -diformat pasangan kunci-nilai yang menentukan payload acara.
  - "path": Jalur ke file yang digunakan sebagai muatan peristiwa.

Dalam contoh di bawah ini, "json" opsi mendefinisikan payload.

Untuk informasi selengkapnya tentang entri ini dan entri lainnya dalam file `launch.json`, lihat [Opsi konfigurasi untuk men-debug aplikasi nirserver](#).

6.

Jika Anda puas dengan konfigurasi debug, untuk memulai debugging, pilih panah putar hijau di sebelahnya. RUN

Saat sesi debugging dimulai, DEBUGCONSOLEpanel menampilkan output debugging dan menampilkan nilai apa pun yang dikembalikan oleh fungsi Lambda. (Saat men-debug AWS SAM aplikasi, AWS Toolkit dipilih sebagai saluran Output di panel Output.)

## Menjalankan dan men-debug sumber daya Amazon API Gateway lokal

Anda dapat menjalankan atau men-debug AWS SAM Sumber daya lokal API Gateway, ditentukan dalam `template.yaml`, dengan menjalankan konfigurasi peluncuran Kode VStype=aws-sam dengan `invokeTarget.target=api`.

### Note

API Gateway mendukung dua jenis API, REST dan HTTP. Namun, fitur API Gateway dengan AWS Toolkit for Visual Studio Code hanya mendukung REST API. Terkadang HTTP API disebut "API Gateway V2 API."

## Untuk menjalankan dan men-debug sumber daya API Gateway lokal

1. Pilih salah satu pendekatan berikut untuk membuat konfigurasi peluncuran untuk sumber daya AWS SAM API Gateway:
  - Opsi 1: Kunjungi kode sumber penanganan (file .js, .cs, atau .py) di file .js, .cs, atau .py)AWS SAMproyek, arahkan kursor ke handler Lambda, dan pilihTambahkan Konfigurasi Debug CodeLens. Kemudian, dalam menu, pilih item bertandaAPI Peristiwa.
  - Opsi 2: EditLaunch . j sondan buat konfigurasi peluncuran baru menggunakan sintaks berikut.

```
{
  "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. Dalam VS CodeJalankanpanel, pilih konfigurasi peluncuran (bernamamyConfigdalam contoh di atas).
3. (Opsional) Tambahkan titik henti ke kode proyek Lambda Anda.
4. JenisF5atau pilihMainkandi dalamJalankanPanel.
5. Di panel output, lihat hasilnya.

## Konfigurasi

Ketika Anda menggunakan `invokeTarget.target` nilai properti `api`, Toolkit mengubah validasi dan perilaku konfigurasi peluncuran untuk mendukung bidang `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": {}
}
```

Ganti nilai-nilai dalam contoh sebagai berikut:

`invokeTarget.logicalId`

Sumber daya API.

`path`

Jalur API yang diminta oleh konfigurasi peluncuran, misalnya, `"path": "/hello"`.

Harus merupakan jalur API valid yang diselesaikan dari `template.yaml` yang ditentukan oleh `invokeTarget.templatePath`.

## httpMethod

Salah satu kata kerja berikut: "delete", "get", "head", "options", "patch", "post", "put".

## payload

Muatan JSON (badan HTTP) untuk mengirim permintaan, dengan struktur dan aturan yang sama dengan bidang [lambda.payload](#).

`payload.path` menunjuk ke file yang berisi muatan JSON.

`payload.json` menentukan muatan JSON sebaris.

## headers

Peta opsional pasangan nama-nilai, yang Anda gunakan untuk menentukan header HTTP untuk disertakan dalam permintaan, seperti yang ditunjukkan dalam contoh berikut.

```
"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

String opsional yang menetapkan querystring dari permintaan, misalnya, "querystring": "abc=def&ghi=jkl".

## AWS

Bagaimana informasi koneksi AWS disediakan. Untuk informasi selengkapnya, lihat [AWS koneksi \("aws"\)](#) properti tabel di [Opsi konfigurasi untuk men-debug aplikasi nirserver](#) Bagian.

## sam

Bagaimana AWS SAM CLI membangun aplikasi. Untuk informasi selengkapnya, lihat [AWS Properti SAM CLI \("sam"\)](#) tabel di [Opsi konfigurasi untuk men-debug aplikasi nirserver](#) Bagian.

## Opsi konfigurasi untuk men-debug aplikasi nirserver

Ketika Anda membuka `launch.json` file untuk mengedit konfigurasi debug, Anda dapat menggunakan Kode VS [IntelliSense](#) fitur untuk melihat dan secara otomatis menyelesaikan properti yang valid. Untuk memicu IntelliSense di editor, tekan `Ctrl+Spasi`.

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

IntelliSense memungkinkan Anda menemukan dan menentukan properti untuk menjalankan fungsi Lambda secara langsung atau dengan AWS SAM templat. Anda juga dapat menentukan properti untuk "lambda" (bagaimana fungsi berjalan), "sam" (bagaimana AWS SAM CLI membangun aplikasi), dan "aws" (bagaimana informasi koneksi AWS disediakan).

AWS SAM: Pemanggilan penanganan Lambda langsung/Pemanggilan Lambda berbasis templat

Properti	Deskripsi
type	Menentukan ekstensi mana yang mengelola konfigurasi peluncuran. Selalu atur ke <code>aws-sam</code> untuk menggunakan AWS SAM CLI untuk membangun dan men-debug secara lokal.
name	Menentukan nama yang mudah dibaca untuk muncul di daftar Konfigurasi peluncuran debug.
request	Menentukan tipe konfigurasi yang akan dilakukan oleh ekstensi yang ditunjuk ( <code>aws-sam</code> ). Selalu atur ke <code>direct-invoke</code> untuk memulai fungsi Lambda.
invokeTarget	Menentukan titik masuk untuk memanggil sumber daya.  Untuk memanggil fungsi Lambda secara langsung, tetapkan nilai untuk bidang <code>invokeTarget</code> berikut: <ul style="list-style-type: none"> <li><code>target</code> – Atur ke <code>code</code>.</li> </ul>

Properti	Deskripsi
	<ul style="list-style-type: none"> <li>• <code>lambdaHandler</code> – Nama dari penanganan fungsi Lambda untuk dipanggil.</li> <li>• <code>projectRoot</code> — Jalur untuk file aplikasi yang berisi penanganan fungsi Lambda.</li> <li>• <code>architecture</code> - Arsitektur prosesor dari lingkungan yang ditiru tempat aplikasi SAM Lambda lokal Anda berjalan. Untuk runtime tertentu, Anda dapat memilih <code>arm64</code> bukan default <code>x86_64</code> arsitektur. Untuk informasi selengkapnya, lihat <a href="#">Membuat aplikasi tanpa server baru (lokal)</a>.</li> </ul> <p>Untuk memohon sumber daya Lambda dengan AWS SAM template, tetapkan nilai untuk berikut <code>invokeTarget</code> bidang:</p> <ul style="list-style-type: none"> <li>• <code>target</code> – Atur ke template.</li> <li>• <code>templatePath</code> — Jalur ke AWS SAM berkas templat.</li> <li>• <code>logicalId</code> – Nama sumber daya dari <code>AWS::Lambda::Function</code> atau <code>AWS::Serverless::Function</code> untuk dipanggil. Anda dapat menemukan nama sumber daya di format YAWS SAM templat. Perhatikan bahwa AWS Toolkit secara implisit mengenali fungsi yang didefinisikan dengan <code>PackageType: Image</code> di dalam AWS SAM templat sebagai <a href="#">Berbasis gambar</a> fungsi Lambda. Untuk informasi lebih lanjut, lihat <a href="#">Paket deployment Lambda</a> di Panduan Developer AWS Lambda.</li> </ul>

### Properti Lambda ("**lambda**")

Properti	Deskripsi
<code>environmentVariables</code>	Melewati parameter operasional ke fungsi Lambda Anda. Misalnya, jika Anda ingin menulis ke bucket Amazon S3 alih-alih melakukan hard-coding nama bucket yang Anda tulis, konfigurasi nama bucket sebagai variabel lingkungan.

Properti	Deskripsi
	<p><b>Note</b></p> <p>Saat menentukan variabel lingkungan untuk aplikasi tanpa server, Anda harus menambahkan konfigurasi ke kedua AWS SAMtemplat (<code>template.yaml</code>) dan <code>launch.json</code> berkas.</p> <p>Contoh pemformatan untuk variabel lingkungan di AWS SAMtemplat:</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>Contoh pemformatan untuk variabel lingkungan di <code>launch.json</code> berkas:</p> <pre>"environmentVariables": {   "SAMPLE1": "My sample 1 value" }</pre>
payload	<p>Menyediakan dua opsi untuk muatan peristiwa yang Anda berikan ke fungsi Lambda Anda sebagai input.</p> <ul style="list-style-type: none"> <li>• <code>"json"</code>: Pasangan nilai kunci berformat JSON yang menentukan muatan peristiwa.</li> <li>• <code>"path"</code>: Jalur ke file yang digunakan sebagai muatan peristiwa.</li> </ul>

Properti	Deskripsi
<code>memoryMB</code>	Menentukan megabyte (MB) memori yang disediakan untuk menjalankan fungsi Lambda yang dipanggil.
<code>runtime</code>	Menentukan runtime yang digunakan fungsi Lambda. Untuk informasi selengkapnya, lihat <a href="#">runtime AWS Lambda</a> .
<code>timeoutSec</code>	Menetapkan waktu yang diizinkan, dalam detik, sebelum waktu sesi debug habis.

Properti	Deskripsi
pathMappings	<p>Menentukan di mana kode lokal dalam kaitannya dengan di mana ia berjalan dalam wadah.</p> <p>Secara default, Toolkit for VS Code set <code>localRoot</code> ke root kode fungsi Lambda di ruang kerja lokal, dan <code>remoteRoot</code> kepada <code>/var/task</code>, yang merupakan direktori kerja default untuk kode yang berjalan di Lambda. Jika direktori kerja diubah di Dockerfile atau dengan <code>WorkingDirectory</code> parameter di AWS CloudFormation file templat setidaknya satu <code>pathMapping</code> entri harus ditentukan agar debugger berhasil memetakan breakpoint yang disetel secara lokal ke kode yang berjalan di wadah Lambda.</p> <p>Contoh pemformatan untuk <code>pathMappings</code> di dalam <code>launch.json</code> berkas:</p> <pre data-bbox="597 940 1507 1255">"pathMappings": [   {     "localRoot": " \${workspaceFolder}/sam-app/ HelloWorldFunction ",     "remoteRoot": " /var/task "   } ]</pre> <p>Peringatan:</p> <ul style="list-style-type: none"><li>• Untuk fungsi Lambda berbasis gambar .NET, <code>remoteRoot</code> entri harus berupa direktori build.</li><li>• Untuk fungsi Lambda berbasis Node.js, Anda hanya dapat menentukan entri pemetaan jalur tunggal.</li></ul>

Toolkit for VS Code AWS SAM CLI untuk membangun dan men-debug aplikasi nirkabel secara lokal. Anda dapat mengonfigurasi perilaku perintah AWS SAM CLI menggunakan properti dari konfigurasi "sam" di file `launch.json`.

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

Properti	Deskripsi	Nilai default
<code>buildArguments</code>	Mengonfigurasi cara perintah <code>sam build</code> membangun kode sumber Lambda Anda. Untuk melihat opsi pembangunan, lihat <a href="#">membangun sam</a> dalam Panduan Developer AWS Serverless Application Model.	Empty string
<code>containerBuild</code>	Menunjukkan apakah akan membangun fungsi Anda di dalam kontainer Docker seperti Lambda.	<code>false</code>
<code>dockerNetwork</code>	Menentukan nama atau ID dari jaringan Docker yang harus dihubungkan ke kontainer Lambda Docker, bersama dengan jaringan bridge default. Jika tidak ditentukan, kontainer Lambda hanya menghubungkan ke jaringan Docker bridge default.	Empty string
<code>localArguments</code>	Menentukan argumen pemanggilan lokal tambahan.	Empty string
<code>skipNewImageCheck</code>	Menentukan apakah perintah harus melewati menarik ke bawah citra Docker terbaru untuk runtime Lambda.	<code>false</code>
<code>template</code>	Menyesuaikan AWS SAM templat yang menggunakan parameter untuk masukan nilai pelanggan. Untuk	<code>"parameters": {}</code>

Properti	Deskripsi	Nilai default
	informasi selengkapnya, lihat <a href="#">Parameter</a> dalam Panduan Pengguna AWS CloudFormation.	

### Properti koneksi AWS ("aws")

Properti	Deskripsi	Nilai default
<code>credentials</code>	Memilih profil tertentu (misalnya, <code>profile:default</code> ) dari file kredensial Anda untuk mendapatkan kredensial AWS.	Yang AWS kredensi yang sudah ada <a href="#">bersama AWS file konfigurasi</a> atau <a href="#">dibagikan AWS File kredensi</a> berikan ke Toolkit for VS Code.
<code>region</code>	Menetapkan Wilayah AWS dari layanan (misalnya, <code>us-east-1</code> ).	Wilayah AWS default yang terkait dengan profil kredensial aktif.

### Contoh: Konfigurasi peluncuran templat

Berikut ini adalah contoh peluncuran file konfigurasi untuk AWS SAM target templat:

```
{
  "configurations": [
    {
      "type": "aws-sam",
      "request": "direct-invoke",
      "name": "my-example:HelloWorldFunction",
      "invokeTarget": {
        "target": "template",
        "templatePath": "template.yaml",
        "logicalId": "HelloWorldFunction"
      },
      "lambda": {
        "payload": {},
        "environmentVariables": {}
      }
    }
  ]
}
```

```
    }
  }
]
}
```

## Contoh: Konfigurasi peluncuran kode

Berikut adalah contoh file konfigurasi peluncuran untuk target fungsi 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": {}
      }
    }
  ]
}
```

## Pemecahan masalah aplikasi nirserver

Topik ini merinci kesalahan umum yang mungkin Anda hadapi saat membuat aplikasi tanpa server dengan Toolkit for VS Code dan cara mengatasinya.

### Topik

- [Bagaimana cara menggunakan samconfig.toml dengan konfigurasi peluncuran SAM?](#)
- [Kesalahan: “RuntimeError: Kontainer tidak ada”](#)
- [Error: “Docker.errors.apiError: Kesalahan Server... Anda telah mencapai batas kecepatan tarik.”](#)
- [Kesalahan: “500 Kesalahan Server: Pemasangan C:\Users\...”](#)
- [Menggunakan WSL, webviews \(misalnya, “Invoke onAWS“bentuk\) rusak](#)

- [Debugging aplikasi TypeScript, tetapi breakpoints tidak bekerja](#)

Bagaimana cara menggunakan `samconfig.toml` dengan konfigurasi peluncuran SAM?

Tentukan lokasi CLI SAM Anda `samconfig.toml` dengan mengkonfigurasi `--config-file` argumen di `sam.localArguments` properti konfigurasi peluncuran Anda. Misalnya, jika file `samconfig.toml` terletak di tingkat atas ruang kerja Anda:

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

Kesalahan: “`RuntimeError: Kontainer tidak ada`”

Parameter `sam build` perintah dapat menunjukkan kesalahan ini jika sistem Anda tidak memiliki ruang disk yang cukup untuk kontainer Docker. Jika penyimpanan sistem Anda hanya memiliki 1-2 GB ruang yang tersedia, `sam build` mungkin gagal selama pemrosesan, bahkan jika penyimpanan sistem tidak sepenuhnya penuh sebelum membangun dimulai. Untuk informasi selengkapnya, lihat [masalah GitHub](#).

Error: “`Docker.errors.apiError: Kesalahan Server... Anda telah mencapai batas kecepatan tarik.`”

Docker Hub membatasi permintaan yang dapat dilakukan pengguna anonim. Jika sistem Anda mencapai batas, Docker gagal dan kesalahan ini muncul dalam tampilan OUTPUT 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")
```

Pastikan bahwa layanan Docker sistem telah diautentikasi dengan kredensi Docker Hub Anda.

Kesalahan: “`500 Kesalahan Server: Pemasangan C:\Users\...`”

Pengguna Windows mungkin melihat kesalahan pemasangan Docker ini saat melakukan debug AWS SAM aplikasi:

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

Cobalah menyegarkan kredensial untuk drive bersama Anda (dalam pengaturan Docker).

Menggunakan WSL, webviews (misalnya, “Invoke onAWS“bentuk) rusak

Ini adalah masalah Kode VS yang dikenal untuk pengguna Cisco VPN. Untuk informasi selengkapnya, lihat [masalah GitHub](#).

Solusi disarankan dalam [masalah pelacakan WSL ini](#).

Debugging aplikasi TypeScript, tetapi breakpoints tidak bekerja

Ini akan terjadi jika tidak ada peta sumber untuk menghubungkan file JavaScript dikompilasi ke file sumber TypeScript. Untuk mengoreksi ini, bukatsconfig.jsonfile dan memastikan opsi dan nilai berikut ditetapkan: "inlineSourceMap": true.

## Menggunakan dokumen otomatisasi Systems Manager

AWS Systems Managermemberi Anda visibilitas dan kontrol atas infrastruktur Anda diAWS. Systems Manager menyediakan antarmuka pengguna terpadu sehingga Anda dapat melihat data operasional dari beberapaAWSlayanan dan otomatisasi tugas operasional di seluruhAWSsumber daya.

SEBUAH[Dokumen Systems Manager](#)menentukan tindakan yang dilakukan Systems Manager pada instans terkelola Anda. Dokumen otomatisasi adalah jenis dokumen Otomatisasi Systems Manager yang Anda gunakan untuk melakukan tugas pemeliharaan dan deployment umum umum seperti membuat atau memperbarui Amazon Machine Image (AMI). Topik ini menguraikan cara membuat, mengedit, menerbitkan, dan menghapus dokumen otomatisasi denganAWS Toolkit for Visual Studio Code.

Topik

- [Asumsi dan prasyarat](#)
- [Izin IAM untuk dokumen otomatisasi Systems Manager](#)
- [Membuat dokumen otomatisasi Systems Manager](#)

- [Membuka dokumen otomatisasi Systems Manager](#)
- [Mengedit dokumen otomatisasi Systems Manager](#)
- [Menerbitkan dokumen otomatisasi Systems Manager](#)
- [Menghapus dokumen otomatisasi Systems Manager](#)
- [Melaksanakan dokumen otomatisasi Systems Manager](#)
- [Memecahkan masalah dokumen Otomasi Systems Manager di Toolkit for VS Code](#)

## Asumsi dan prasyarat

Sebelum memulai, pastikan:

- Anda telah menginstal Visual Studio Code dan versi terbaru dari AWS Toolkit for Visual Studio Code. Untuk informasi selengkapnya, lihat [Instalasi AWS Toolkit for Visual Studio Code](#).
- Kau sudah familiar dengan Systems Manager. Untuk informasi selengkapnya, lihat [Panduan Pengguna AWS Systems Manager](#).
- Anda sudah familiar dengan kasus penggunaan Otomasi Systems Manager. Untuk informasi selengkapnya, lihat [AWS Otomasi Systems Manager](#) di dalam AWS Systems Manager Panduan Pengguna.

## Izin IAM untuk dokumen otomatisasi Systems Manager

Dalam Toolkit for VS Code Anda harus memiliki profil kredensi yang berisi AWS Identity and Access Management (IAM) izin yang diperlukan untuk membuat, mengedit, menerbitkan, dan menghapus dokumen Otomasi Systems Manager. Dokumen kebijakan berikut mendefinisikan izin IAM yang diperlukan yang dapat digunakan dalam kebijakan utama:

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

Untuk informasi tentang cara memperbarui kebijakan IAM, lihat [Membuat kebijakan IAM](#) di dalam Panduan Pengguna IAM. Untuk informasi tentang cara mengatur profil kredensi Anda, lihat [AWS IAM kredensialnya](#).

## Membuat dokumen otomatisasi Systems Manager

Anda dapat membuat dokumen otomatisasi baru di JSON atau YAML menggunakan Kode Studio Visual. Ketika Anda membuat dokumen Otomasi baru, itu akan disajikan dalam file tanpa judul. Anda dapat memberi nama file Anda dan menyimpannya dalam VS Code, namun nama file tidak terlihat AWS.

Membuat dokumen Otomasi baru

1. Buka VS Code.
2. Pada Lihat menu, pilih Palet Perintah untuk membuka Command Palette.
3. Di Palet Perintah, masukkan **AWSToolkit Membuat Dokumen Systems Manager baru Secara Lokal**.
4. Pilih salah satu template starter untuk contoh Hello World.
5. Pilih salah satu JSON atau YAML.

Dokumen Otomasi baru dibuat.

### Note

Dokumen Otomasi baru Anda di VS Code tidak muncul secara otomatis AWS. Anda harus mempublikasikannya AWS sebelum Anda dapat menjalankannya.

## Membuka dokumen otomatisasi Systems Manager

Anda menggunakan AWS Explorer untuk menemukan dokumen Otomasi Systems Manager yang ada. Saat Anda membuka dokumen Otomasi yang ada, dokumen tersebut muncul sebagai file tanpa judul di VS Code.

Untuk membuka dokumen Otomasi

1. Buka VS Code.
2. Dari navigasi sebelah kiri, pilih AWS untuk membuka AWS Explorer.
3. Di AWS Explorer, untuk Systems Manager, pilih ikon download pada dokumen yang ingin Anda buka dan kemudian pilih versi dokumen. File akan terbuka dalam format untuk versi itu. Jika tidak, pilih salah satu Unduh sebagai JSON atau Unduh sebagai YAML.

### Note

Menyimpan dokumen Automation secara lokal sebagai file di VS Code tidak membuatnya muncul di AWS. Perlu dipublikasikan AWS sebelum mengeksekusi.

## Mengedit dokumen otomatisasi Systems Manager

Jika Anda memiliki dokumen Otomasi, dokumen tersebut muncul di Dimiliki oleh Me kategori dokumen Systems Manager di AWS Explorer. Anda dapat memiliki dokumen otomatisasi yang sudah ada di AWS, dan Anda dapat memiliki dokumen baru atau yang diperbarui yang sebelumnya Anda terbitkan AWS dari VS Code.

Ketika Anda membuka dokumen Automation untuk mengedit dalam VS Code, Anda dapat melakukan lebih banyak dengan itu daripada yang Anda bisa di AWS Management Console. Misalnya:

- Ada validasi skema pada keduanya JSON dan YAML format.
- Ada cuplikan yang tersedia di editor dokumen bagi Anda untuk membuat salah satu jenis langkah otomatisasi.
- Ada dukungan lengkap otomatis pada berbagai opsi di JSON dan YAML.

## Bekerja dengan versi

Dokumen Otomasi Systems Manager menggunakan versi untuk manajemen perubahan. Anda dapat memilih versi default untuk dokumen Otomasi di VS Code.

Untuk mengatur versi default

- DiAWSExplorer, arahkan ke dokumen yang ingin Anda atur versi default, buka menu konteks (klik kanan) untuk dokumen, dan pilih **Tetapkan versi default**.

### Note

Jika dokumen yang dipilih hanya memiliki satu versi, Anda tidak akan dapat mengubah default.

## Menerbitkan dokumen otomatisasi Systems Manager

Setelah mengedit dokumen Otomasi di VS Code, Anda dapat mempublikasikannya ke AWS.

Untuk memublikasikan dokumen Otomasi

1. Buka dokumen Automation yang ingin Anda publikasikan menggunakan prosedur yang diuraikan [Membuka dokumen otomatisasi Systems Manager](#).
2. Buat perubahan yang ingin Anda terbitkan. Untuk informasi selengkapnya, lihat [Mengedit dokumen otomatisasi Systems Manager](#).
3. Di bagian kanan atas file terbuka, pilih ikon unggah.
4. Di kotak dialog alur kerja penerbitan, pilih **AWS Wilayah** yang dokumen otomatisasi Anda ingin menerbitkan dokumen otomatisasi.
5. Jika Anda menerbitkan dokumen baru, pilih **Buat Cepat**. Jika tidak, pilih **Pembaruan cepat** untuk memperbarui dokumen Otomasi yang ada di dalamnya **AWS Wilayah**.
6. Masukkan nama untuk dokumen Otomasi ini.

Ketika Anda memublikasikan pembaruan ke dokumen Otomasi yang ada ke AWS, versi baru ditambahkan ke dokumen.

## Menghapus dokumen otomatisasi Systems Manager

Anda dapat menghapus dokumen Otomasi di VS Code. Menghapus dokumen Otomasi menghapus dokumen dan semua versi dokumen.

### Important

- Menghapus adalah tindakan destruktif yang tidak dapat dibatalkan.
- Menghapus dokumen Automation yang telah dijalankan tidak menghapus AWS sumber daya yang dibuat atau dimodifikasi saat dimulai.

### Menghapus dokumen Otomasi

1. Buka VS Code.
2. Dari navigasi sebelah kiri, pilih AWS membuka AWSExplorer.
3. Di AWSExplorer, untuk Systems Manager, buka menu konteks (klik kanan) untuk dokumen yang ingin Anda hapus, dan pilih Hapus Dokumen.

## Melaksanakan dokumen otomatisasi Systems Manager

Setelah dokumen Automation Anda dipublikasikan ke AWS, Anda dapat melakukannya untuk melakukan tugas-tugas atas nama Anda di AWS akun. Untuk menjalankan dokumen Otomasi, Anda menggunakan AWS Management Console, API Systems Manager, AWS CLI, atau AWS Tools for PowerShell. Untuk petunjuk tentang cara menjalankan dokumen otomatisasi, lihat [Menjalankan otomatisasi sederhana](#) di dalam AWS Systems Manager Panduan Pengguna.

Atau, jika Anda ingin menggunakan salah satu AWS SDK dengan API Systems Manager untuk menjalankan dokumen Otomasi Anda, lihat [AWS Referensi SDK](#).

### Note

Mengeksekusi dokumen Otomasi dapat membuat sumber daya baru di AWS dan dapat dikenakan biaya penagihan. Kami sangat menyarankan agar Anda memahami apa yang akan dibuat dokumen Otomasi Anda di akun Anda sebelum Anda memulainya.

## Memecahkan masalah dokumen Otomasi Systems Manager di Toolkit for VS Code

Saya menyimpan dokumen Automation saya di VS Code, tetapi saya tidak melihatnya di AWS Management Console.

Menyimpan dokumen Otomasi dalam VS Code tidak mempublikasikan dokumen Automation ke AWS. Untuk informasi selengkapnya tentang menerbitkan dokumen otomatisasi Anda, lihat [Menerbitkan dokumen otomatisasi Systems Manager](#).

Penerbitan dokumen Otomasi saya gagal dengan kesalahan izin.

Pastikan Anda AWS profil kredensi memiliki izin yang diperlukan untuk mempublikasikan dokumen Otomasi. Sebagai contoh kebijakan izin, lihat [Izin IAM untuk dokumen otomatisasi Systems Manager](#).

Saya menerbitkan dokumen Automation saya ke AWS, tapi aku tidak melihatnya di AWS Management Console.

Pastikan bahwa Anda telah menerbitkan dokumen yang sama AWS Wilayah yang sedang Anda jelajahi AWS Management Console.

Saya telah menghapus dokumen Automation saya, tetapi saya masih ditagih untuk sumber daya yang dibuatnya.

Menghapus dokumen Otomasi tidak menghapus sumber daya yang dibuat atau diubah. Anda dapat mengidentifikasi AWS sumber daya yang telah Anda buat dari [AWS Konsol Manajemen Penagihan](#), jelajahi tagihan Anda, dan pilih sumber daya apa yang akan dihapus dari sana.

## Bekerja dengan AWS Step Functions

AWS Toolkit for Visual Studio Code (Kode VS) menyediakan dukungan untuk [AWS Step Functions](#). Menggunakan Toolkit for VS Code, Anda dapat membuat, memperbarui, dan menjalankan mesin status Step Functions.

Topik

- [Bekerja dengan AWS Step Functions](#)

## Bekerja dengan AWS Step Functions

Anda dapat menggunakan AWS Toolkit for Visual Studio Code (Kode VS) untuk melakukan berbagai operasi dengan [mesin negara](#).

### Topik

- [Prasyarat](#)
- [Bekerja dengan mesin negara di VS Code](#)
- [Templat mesin negara](#)
- [Visualisasi grafik mesin negara](#)
- [Cuplikan kode](#)
- [Penyelesaian dan validasi kode](#)

### Prasyarat

- Pastikan sistem Anda memenuhi prasyarat yang ditentukan dalam [Menginstal Toolkit for VS Code, lalu instal toolkit](#).
- Pastikan Anda telah mengonfigurasi kredensial Anda sebelum membuka Explorer.AWS

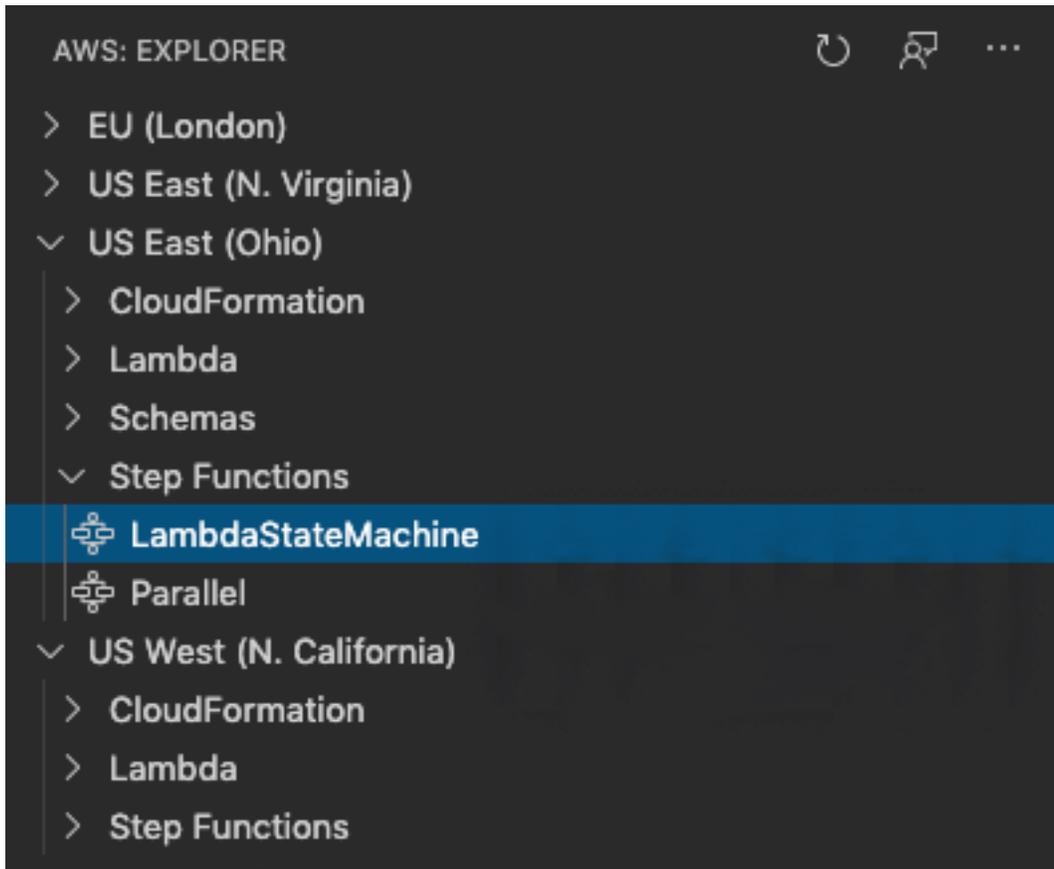
### Bekerja dengan mesin negara di VS Code

Anda dapat menggunakan VS Code untuk berinteraksi dengan mesin status jarak jauh, dan mengembangkan mesin status secara lokal dalam format JSON atau YAMB. Anda dapat membuat atau memperbarui mesin status, membuat daftar mesin status yang ada, menjalankannya, dan mengunduhnya. Kode VS juga memungkinkan Anda membuat mesin status baru dari templat, melihat visualisasi mesin status Anda, dan menyediakan cuplikan kode, penyelesaian kode, serta validasi kode.

### Daftar mesin negara yang ada

Jika Anda telah membuat mesin status, Anda dapat melihat daftarnya:

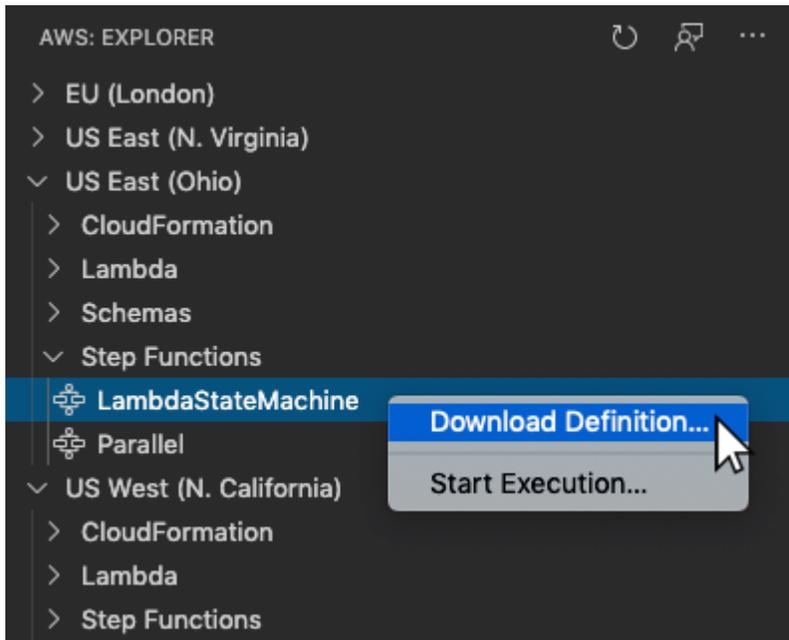
1. Buka AWS Explorer.
2. Pilih Step Functions
3. Verifikasi bahwa itu mencantumkan semua mesin negara di akun Anda.



## Unduh mesin negara

Untuk mengunduh mesin negara:

1. Di AWS Explorer, klik kanan mesin status yang ingin Anda unduh.
2. Pilih Unduh, lalu pilih lokasi di mana Anda ingin mengunduh mesin negara.
3. Verifikasi bahwa itu diunduh dengan benar.



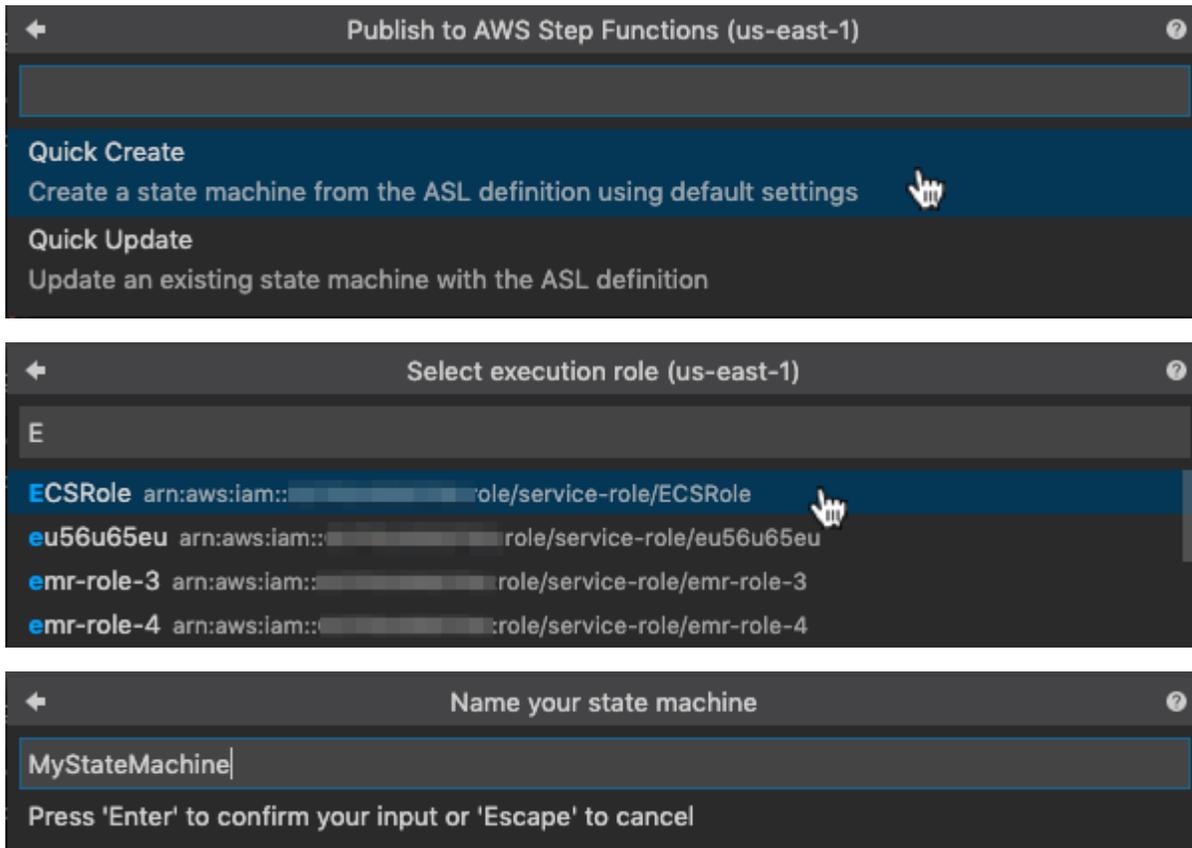
## Buat mesin status

Anda dapat membuat mesin status baru sendiri, atau Anda dapat menggunakan templat. Untuk informasi selengkapnya tentang membuat mesin status dari templat, lihat bagian State Machine Templates. Untuk membuat mesin negara baru:

1. Buat file [Amazon States Language](#) (ASL) baru dengan definisi mesin status Anda. Gunakan menu di kanan bawah untuk mengaturnya sebagai Bahasa Negara Amazon.
2. Pilih Publish to Step Functions.

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

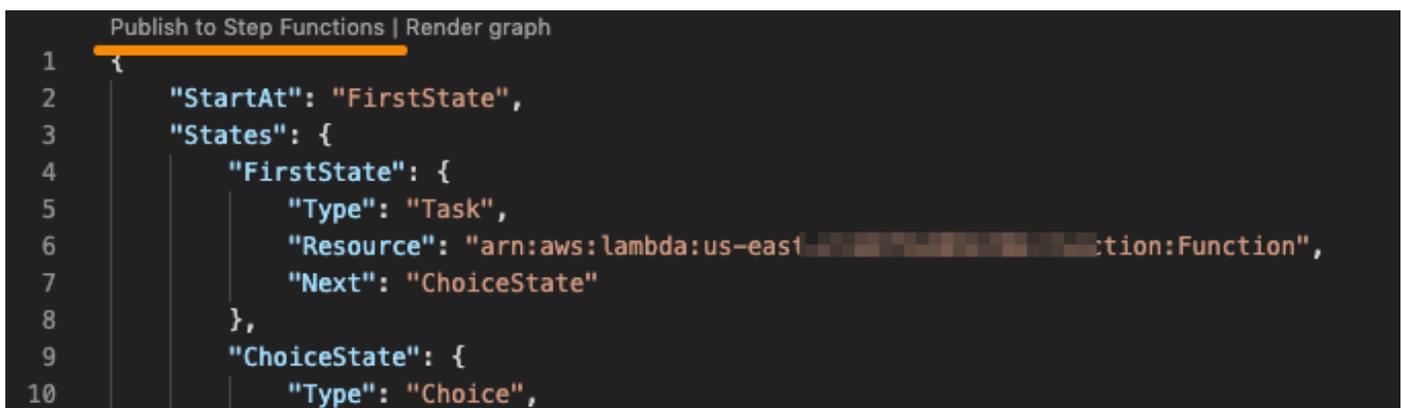
3. Pilih Buat Cepat, pilih peran, dan beri nama mesin status Anda.



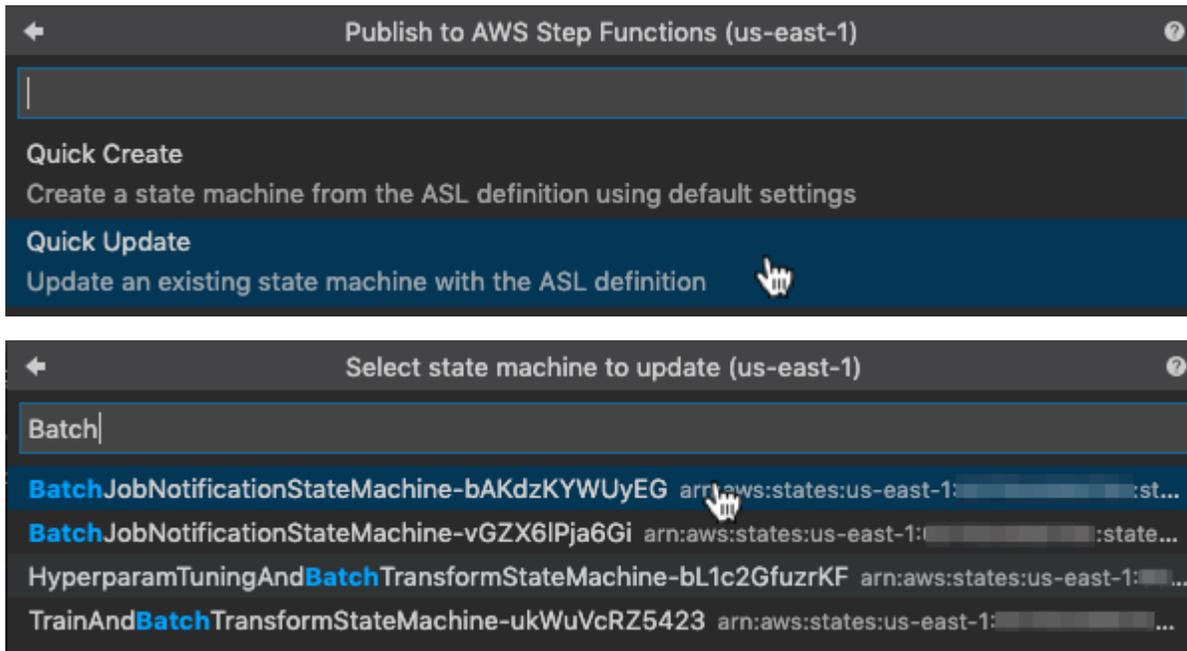
Perbarui mesin negara

Untuk memperbarui mesin status:

1. Edit file ASL dengan definisi mesin status Anda.
2. Pilih Publish to Step Functions.



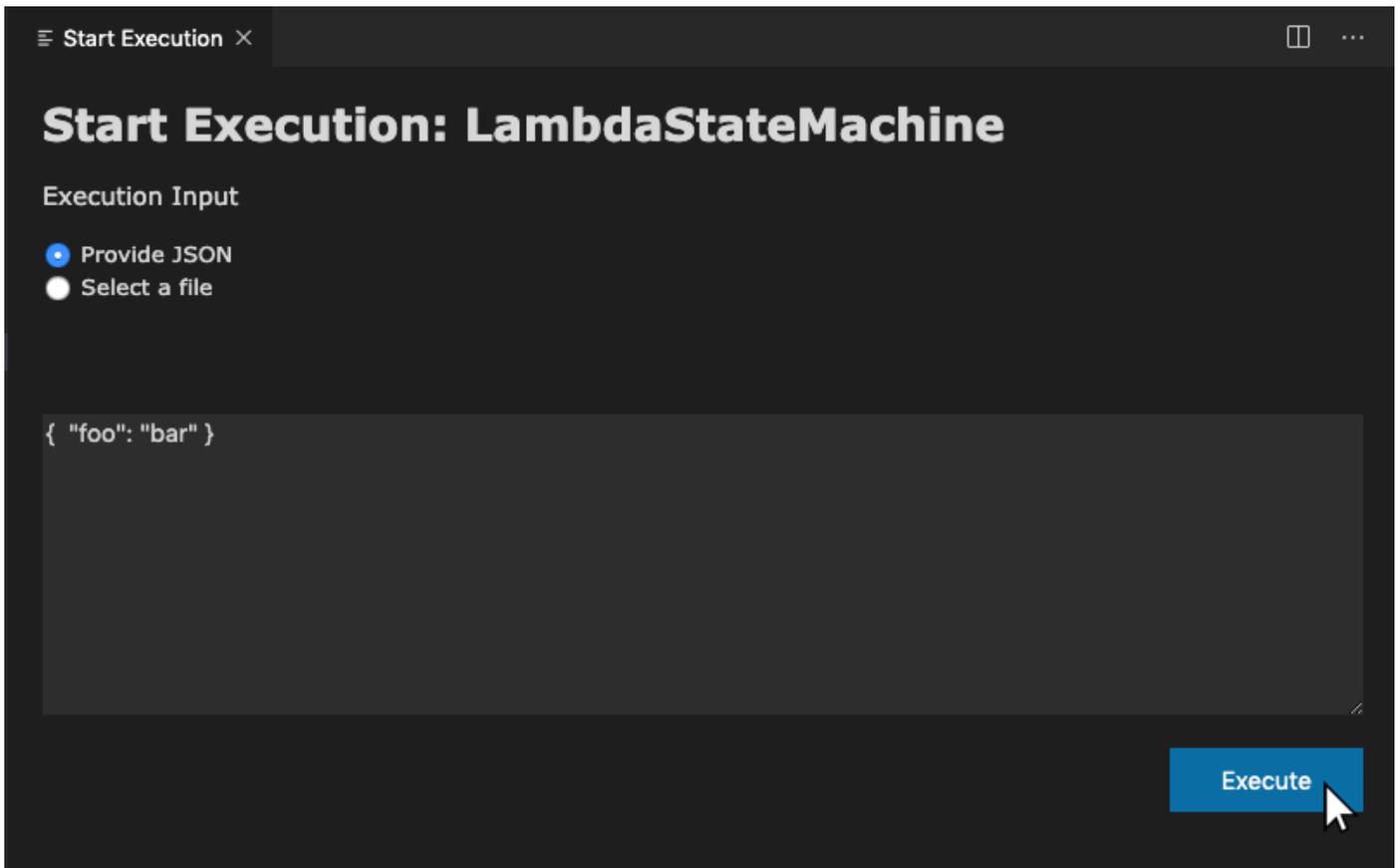
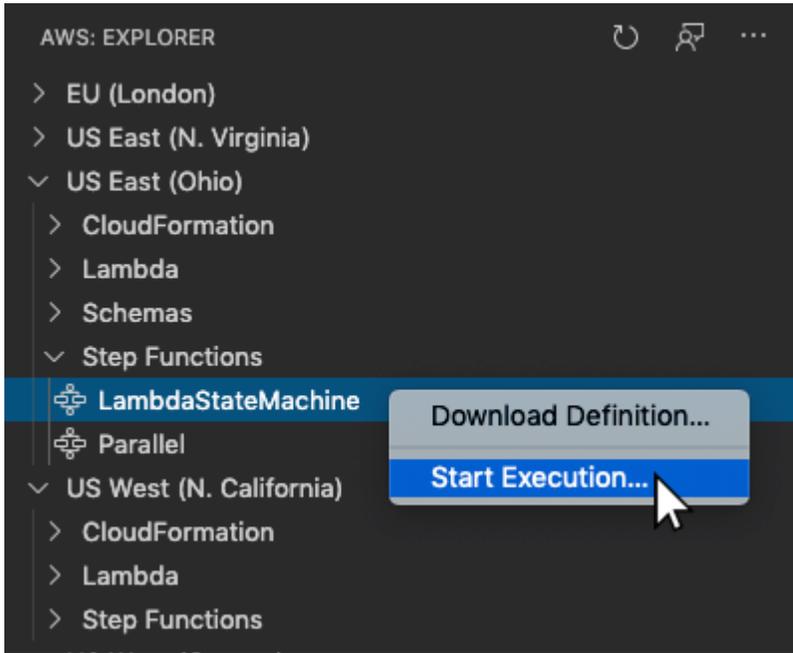
3. Pilih Pembaruan Cepat, lalu pilih mesin status yang ingin Anda perbarui.



Jalankan mesin negara

Untuk menjalankan mesin status:

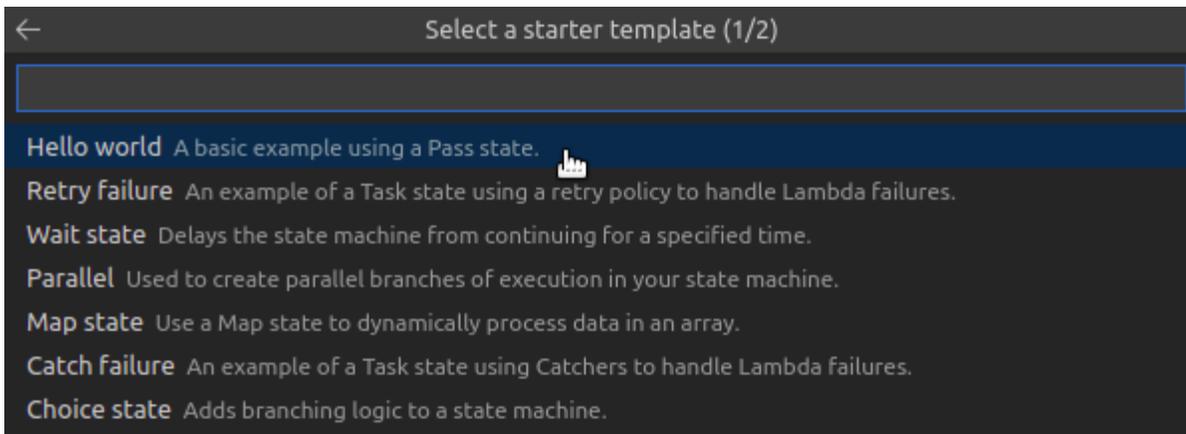
1. Di AWS Explorer, klik kanan mesin status yang ingin Anda jalankan.
2. Berikan masukan untuk mesin negara. Anda dapat mencoba input dari file, dan masukan dalam kotak teks.
3. Mulai mesin status dan verifikasi bahwa itu berjalan dengan sukses.



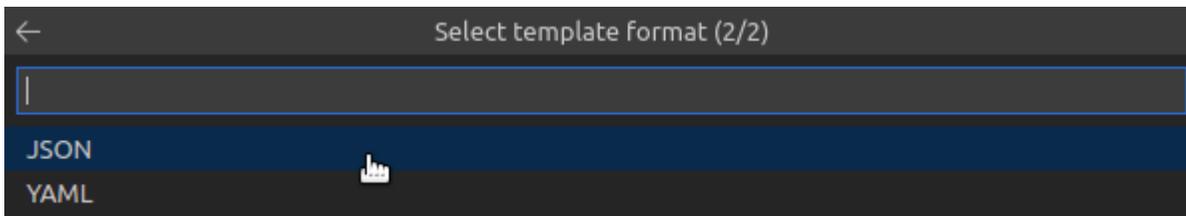
## Templat mesin negara

Saat Anda membuat mesin status, Anda memiliki opsi untuk membuatnya dari templat. Template berisi definisi mesin status sampel dengan beberapa status yang umum digunakan, dan memberi Anda titik awal. Untuk menggunakan templat mesin negara:

1. Buka Command Palette di VS Code.
2. Pilih AWS Toolkit Buat mesin status Step Functions baru.
3. Pilih template yang ingin Anda gunakan.



4. Pilih apakah Anda ingin menggunakan format template JSON atau YAMAL.



## Visualisasi grafik mesin negara

Visualisasi grafik memungkinkan Anda melihat seperti apa mesin status Anda dalam format grafis. Saat Anda membuat visualisasi grafik, tab lain akan terbuka dan menampilkan visualisasi mesin status JSON atau YAMB. Anda kemudian dapat membandingkan definisi mesin negara yang Anda tulis bersamaan dengan visualisasinya. Saat Anda mengubah definisi mesin status Anda, visualisasi akan diperbarui.

**Note**

Untuk membuat visualisasi definisi mesin negara, definisi harus terbuka di editor aktif. Jika Anda menutup atau mengganti nama file definisi, visualisasi akan ditutup.

Untuk membuat visualisasi grafik mesin negara:

1. Tentukan mesin negara Anda.
2. Buka Command Palette di VS Code.
3. Untuk membuat visualisasi, gunakan tombol visualisasi di sudut kanan atas, atau pilih AWS Render grafik.

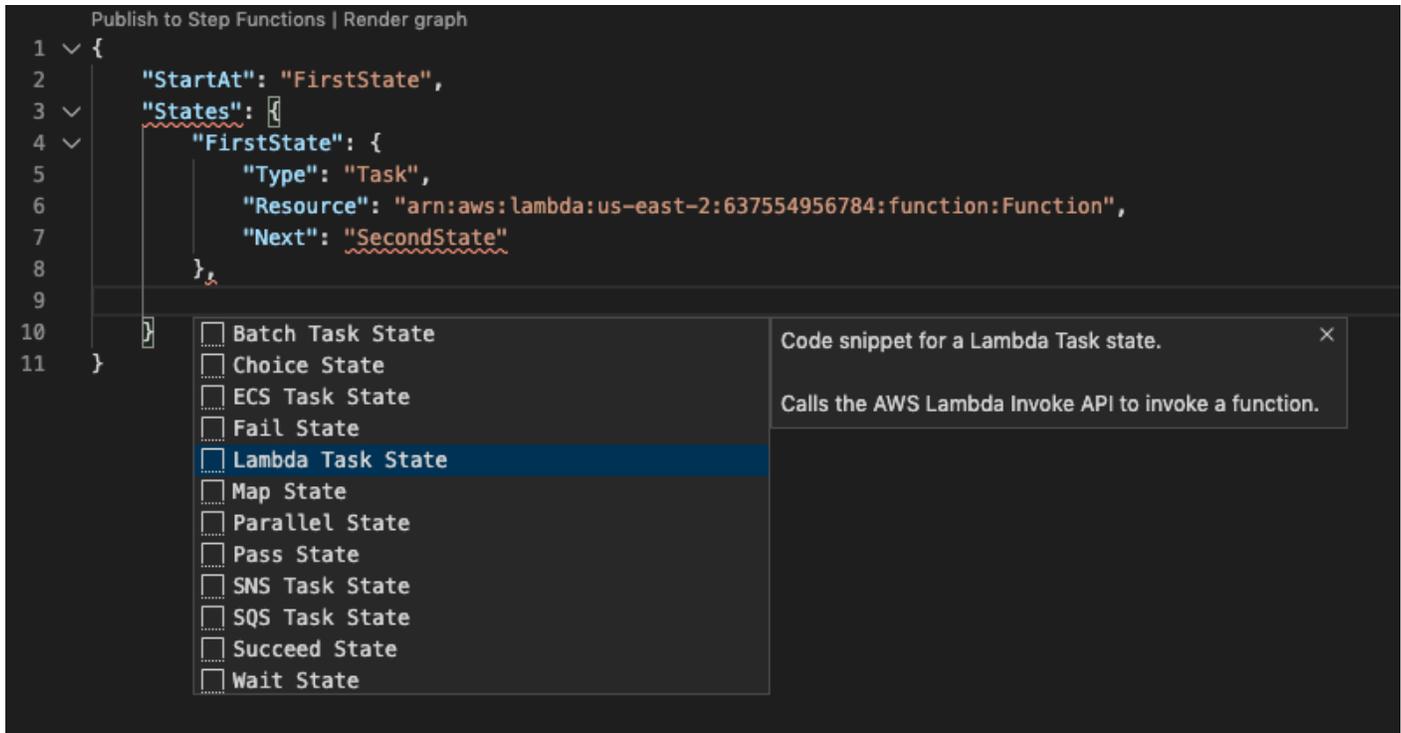
The screenshot displays the AWS Step Functions ASL editor in VS Code. The left pane shows the JSON definition for a state machine named 'Choice-example.asl.json'. The right pane shows the rendered state machine graph. The graph starts at a 'Start' node, leading to 'FirstState', then to a 'ChoiceState'. From 'ChoiceState', the flow branches into 'FirstMatchState' and 'SecondMatchState'. 'FirstMatchState' leads to 'DefaultState', and 'SecondMatchState' leads to 'NextState'. Both 'DefaultState' and 'NextState' lead to the 'End' node.

```
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   "End": {}
47 }
48 }
```

## Cuplikan kode

Cuplikan kode memungkinkan Anda memasukkan bagian pendek kode. Untuk menggunakan cuplikan kode:

1. Buka file dan simpan dengan ekstensi `.asl.json` untuk format JSON, atau `.asl.yaml` untuk format YAMAL.
2. Buat mesin negara baru dengan properti States.
3. Tempatkan kursor di dalam Negara.
4. Gunakan kombinasi tombol `Control + Space`, dan pilih cuplikan kode pilihan Anda.
5. Gunakan Tab untuk melintasi variabel dan parameter dalam cuplikan kode.
6. Uji Coba Lagi dan Tangkap cuplikan dengan menempatkan kursor dalam status terkait.



## Penyelesaian dan validasi kode

Untuk melihat cara kerja penyelesaian kode:

1. Buat beberapa negara.
2. Tempatkan kursor setelah properti Berikutnya, `StartAt`, atau `Default`.
3. Gunakan kombinasi tombol `Control + Space` untuk membuat daftar penyelesaian yang tersedia. Properti tambahan dapat diakses menggunakan `Control + Space` lagi, dan akan didasarkan `Type` pada `State`.
4. Saat Anda bekerja, validasi kode akan terjadi untuk:
  - Properti yang hilang

- Nilai yang salah
- Tidak ada status terminal
- Tidak ada keadaan yang ditunjuk

```

"FirstMatchState": {
  "Type": "Task",
  "Resource": "arn:aws:lambda:us-east-2:637554956784:function:Function",
  "Next": ""
},
"SecondMatchState": {
  "Type": "Task",
  "Resource": "arn:aws:lambda:us-east-2:637554956784:function:Function",
  "Next": ""
},
"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": [
    {
      "Type": "Task",
      "Resource": "arn:aws:lambda:us-east-2:637554956784:function:Function",
      "End": true
    }
  ]
}

```

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.

## Bekerja dengan Threat Composer

Anda dapat menggunakan AWS Toolkit for Visual Studio Code untuk bekerja dengan alat Threat Composer. Threat Composer adalah alat pemodelan ancaman yang dapat menyederhanakan proses pemodelan ancaman Anda.

Untuk informasi rinci tentang alat Threat Composer, lihat repositori [Threat Composer GitHub](#) .

Topik berikut menjelaskan cara bekerja dengan Threat Composer di. AWS Toolkit for Visual Studio Code

Topik

- [Bekerja dengan Threat Composer dari Toolkit](#)

## Bekerja dengan Threat Composer dari Toolkit

Dengan Threat Composer Anda dapat membuat, melihat, dan mengedit model ancaman Threat Composer langsung di VS Code. Untuk informasi rinci tentang alat Threat Composer, lihat repositori [Threat Composer GitHub](#) .

Bagian berikut menjelaskan cara mengakses alat Threat Composer di. AWS Toolkit for Visual Studio Code

### Mengakses Threat Composer dari Toolkit

Ada 3 cara utama Anda dapat mengakses Threat Composer dari Toolkit.

Mengakses Threat Composer melalui model ancaman yang ada

Untuk membuka Threat Composer, buka file model ancaman (ekstensi `.tc.json`) yang ada di VS Code. Threat Composer secara otomatis membuka dan membuat visualisasi file model ancaman Anda di jendela editor VS Code.

Membuat model ancaman Threat Composer baru

1. Dari menu utama VS Code, perluas File, lalu pilih File Baru.
2. Dari dialog File Baru, pilih Threat Composer File... .
3. Saat diminta, masukkan `file name`, lalu tekan **enter** tombol untuk membuka Threat Composer dan buat visualisasi file model ancaman kosong Anda di jendela editor VS Code baru.

Membuat model ancaman Threat Composer baru dari Command Palette

1. Dari VS Code, buka Command Palette dengan menekan **Cmd + Shift + P** atau **Ctrl + Shift + P** (Windows).

2. Di bidang pencarian, masukkan **Threat Composer** dan pilih Buat File Komposer Ancaman Baru saat mengisi hasil.
3. Saat diminta, masukkan `file name`, lalu tekan **enter** tombol untuk membuka Threat Composer dan buat visualisasi file model ancaman kosong Anda di jendela editor VS Code baru.

## Bekerja dengan sumber daya

Selain mengakses AWS layanan yang terdaftar secara default di AWS Explorer, Anda juga dapat pergi ke Sumber Daya dan memilih dari ratusan sumber daya untuk ditambahkan ke antarmuka. Di AWS, sumber daya adalah entitas yang dapat Anda gunakan. Beberapa sumber daya yang dapat ditambahkan termasuk Amazon AppFlow, Amazon Kinesis Data AWS IAM Streams, peran, VPC Amazon, dan CloudFront distribusi Amazon.

Setelah membuat pilihan, Anda dapat pergi ke Sumber Daya dan memperluas jenis sumber daya untuk mencantumkan sumber daya yang tersedia untuk jenis itu. Misalnya, jika Anda memilih jenis `AWS Toolkit:Lambda::Function` sumber daya, Anda dapat mengakses sumber daya yang menentukan fungsi yang berbeda, propertinya, dan atributnya.

Setelah menambahkan jenis sumber daya ke Sumber Daya, Anda dapat berinteraksi dengannya dan sumber dayanya dengan cara berikut:

- Lihat daftar sumber daya yang ada yang tersedia di AWS Wilayah saat ini untuk jenis sumber daya ini.
- Lihat versi read-only JSON file yang menjelaskan sumber daya.
- Salin pengenalan sumber daya untuk sumber daya.
- Lihat AWS dokumentasi yang menjelaskan tujuan dari jenis sumber daya dan skema (dalam JSON dan YAML format) untuk pemodelan sumber daya.
- Buat sumber daya baru dengan mengedit dan menyimpan templat JSON berformat yang sesuai dengan skema. \*
- Perbarui atau hapus sumber daya yang ada. \*

### Important

\* Dalam rilis saat ini opsi untuk membuat, mengedit, dan menghapus sumber daya adalah fitur eksperimental. AWS Toolkit for Visual Studio Code Karena fitur eksperimental terus diuji

dan diperbarui, mereka mungkin memiliki masalah kegunaan. Dan fitur eksperimental dapat dihapus dari AWS Toolkit for Visual Studio Code tanpa pemberitahuan.

Untuk mengizinkan penggunaan fitur eksperimental untuk sumber daya, buka panel Pengaturan di Kode VS Anda IDE, lalu perluas Ekstensi dan pilih AWS Toolkit.

Di bawah Eksperimen AWS Toolkit, pilih jsonResourceModification untuk memungkinkan Anda membuat, memperbarui, dan menghapus sumber daya.

Untuk informasi selengkapnya, lihat [Bekerja dengan fitur eksperimental](#).

## IAM izin untuk mengakses sumber daya

Anda memerlukan AWS Identity and Access Management izin khusus untuk mengakses sumber daya yang terkait dengan AWS layanan. Misalnya, IAM entitas, seperti pengguna atau peran, memerlukan izin Lambda untuk mengakses sumber daya. `AWS Toolkit::Lambda::Function`

Selain izin untuk sumber daya layanan, IAM entitas memerlukan izin untuk mengizinkan Toolkit for VS Code AWS memanggil operasi Cloud API Control atas namanya. API Operasi Cloud Control memungkinkan IAM pengguna atau peran untuk mengakses dan memperbarui sumber daya jarak jauh.

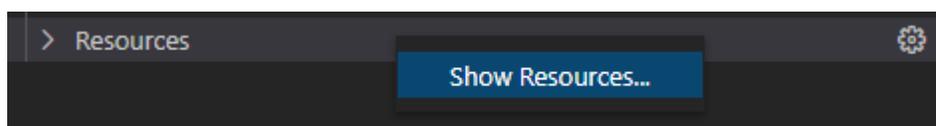
Cara termudah untuk memberikan izin adalah dengan melampirkan kebijakan AWS terkelola PowerUserAccess, ke IAM entitas yang memanggil API operasi ini menggunakan antarmuka Toolkit. [Kebijakan terkelola](#) ini memberikan berbagai izin untuk melakukan tugas pengembangan aplikasi, termasuk memanggil API operasi.

Untuk izin tertentu yang menentukan API operasi yang diizinkan pada sumber daya jarak jauh, lihat [API Panduan Pengguna Kontrol AWS Cloud](#).

## Menambah dan berinteraksi dengan sumber daya yang ada

1. Di AWS Explorer, klik kanan Resources dan pilih Show Resources.

Panel menampilkan daftar jenis sumber daya yang tersedia untuk dipilih.



2. Di panel pemilihan, pilih jenis sumber daya yang akan ditambahkan ke AWS Explorer dan tekan Kembali atau pilih OK untuk mengonfirmasi.

Jenis sumber daya yang Anda pilih tercantum di bawah Sumber Daya.

 Note

Jika Anda telah menambahkan jenis sumber daya ke AWS Explorer dan kemudian menghapus kotak centang untuk jenis itu, itu tidak lagi terdaftar di bawah Sumber daya setelah Anda memilih OK. Hanya jenis sumber daya yang saat ini dipilih yang terlihat di AWS Explorer.

3. Untuk melihat sumber daya yang sudah ada untuk jenis sumber daya, perluas entri untuk jenis itu.

Daftar sumber daya yang tersedia ditampilkan di bawah jenis sumber dayanya.

4. Untuk berinteraksi dengan sumber daya tertentu, klik kanan namanya dan pilih salah satu opsi berikut:
  - Copy Resource Identifier: Salin pengenal untuk sumber daya tertentu ke clipboard. (Misalnya, `AWS Toolkit:DynamoDB::Table` sumber daya dapat diidentifikasi menggunakan `TableName` properti.)
  - Pratinjau: Lihat versi read-only dari template JSON -format yang menjelaskan sumber daya.

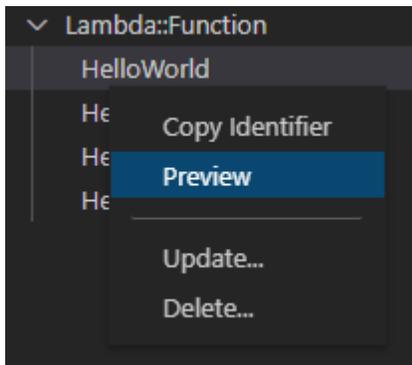
Setelah template sumber daya ditampilkan, Anda dapat memodifikasinya dengan memilih ikon Perbarui di sebelah kanan tab editor. Untuk memperbarui sumber daya, Anda harus [???](#) mengaktifkan yang diperlukan.

- Pembaruan: Edit template JSON yang diformat untuk sumber daya di editor Kode VS. Untuk informasi selengkapnya, lihat [Membuat dan mengedit sumber daya](#).
- Hapus: Hapus sumber daya dengan mengonfirmasi penghapusan di kotak dialog yang ditampilkan. (Menghapus sumber daya saat ini merupakan [???](#) versi AWS Toolkit for Visual Studio Code.)

 Warning

Jika Anda menghapus sumber daya, AWS CloudFormation tumpukan apa pun yang menggunakan sumber daya itu akan gagal diperbarui. Untuk memperbaiki kegagalan pembaruan ini, Anda perlu membuat ulang sumber daya atau menghapus referensi ke

dalamnya di AWS CloudFormation template tumpukan. Untuk informasi lebih lanjut, lihat [artikel Pusat Pengetahuan ini](#).



## Membuat dan mengedit sumber daya

### Important

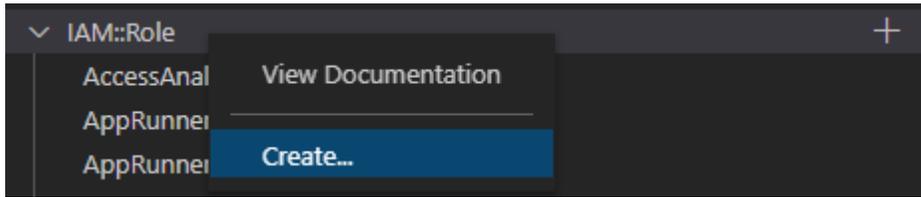
Pembuatan dan pembaruan sumber daya saat ini ada [???](#) dalam versi AWS Toolkit for Visual Studio Code.

Membuat sumber daya baru melibatkan penambahan jenis sumber daya ke daftar Sumber Daya dan kemudian mengedit templat JSON berformat -yang mendefinisikan sumber daya, propertinya, dan atributnya.

Misalnya, sumber daya yang termasuk dalam jenis `AWS Toolkit:SageMaker::UserProfile` sumber daya ditentukan dengan templat yang membuat profil pengguna untuk Amazon SageMaker Studio. Template yang mendefinisikan sumber daya profil pengguna ini harus sesuai dengan skema tipe sumber daya untuk `AWS Toolkit:SageMaker::UserProfile` Jika templat tidak sesuai dengan skema karena properti yang hilang atau salah, misalnya, sumber daya tidak dapat dibuat atau diperbarui.

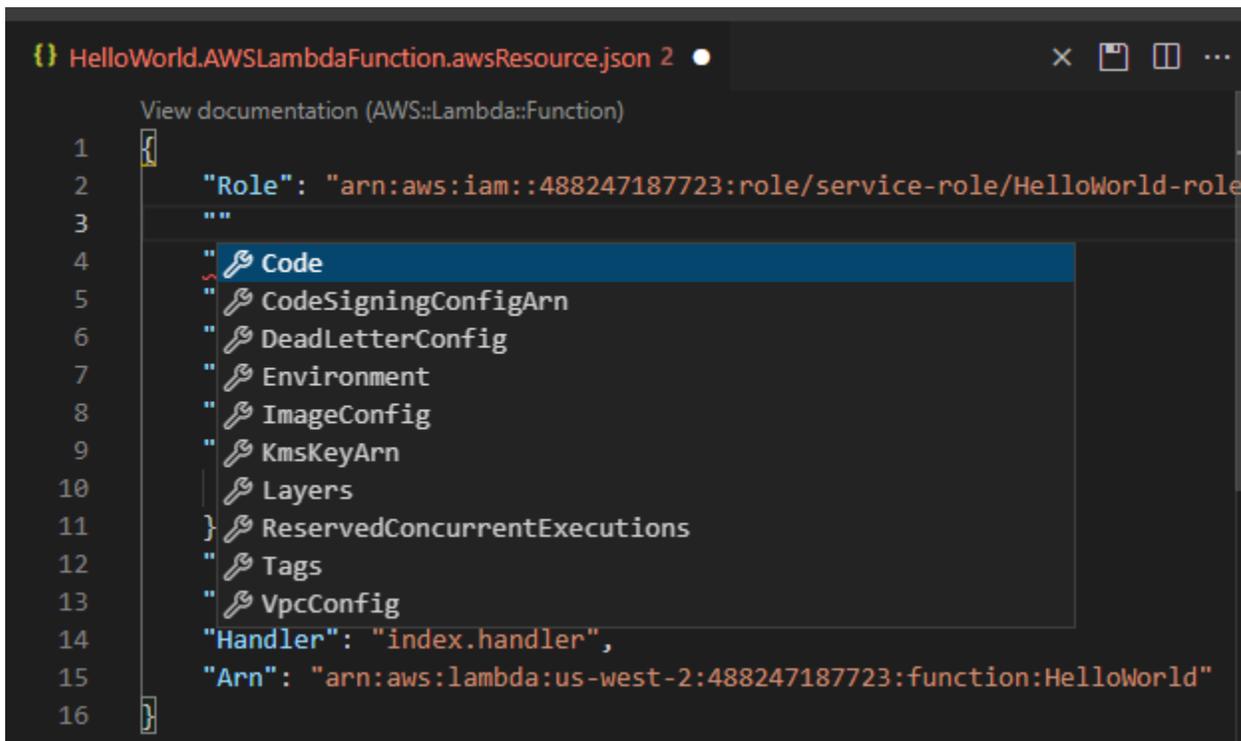
1. Tambahkan jenis sumber daya untuk sumber daya yang ingin Anda buat dengan mengklik kanan Resources dan memilih Show Resources.
2. Setelah jenis sumber daya ditambahkan di bawah Sumber daya, pilih ikon plus (“+”) untuk membuka file templat di editor baru.

Atau, Anda dapat mengklik kanan nama jenis sumber daya dan memilih Buat. Anda juga dapat mengakses informasi tentang cara memodelkan sumber daya dengan memilih Lihat Dokumentasi.



3. Di editor, mulailah mendefinisikan properti yang membentuk template sumber daya. Fitur pelengkapan otomatis menyarankan nama properti yang sesuai dengan skema template Anda. Saat Anda mengarahkan kursor ke jenis properti, panel menampilkan deskripsi untuk apa properti itu digunakan. Untuk informasi rinci tentang skema, pilih Lihat Dokumentasi.

Teks apa pun yang tidak sesuai dengan skema sumber daya ditunjukkan oleh garis bawah merah bergelombang.



4. Setelah Anda selesai mendeklarasikan sumber daya Anda, pilih ikon Simpan untuk memvalidasi template Anda dan menyimpan sumber daya ke Cloud jarak jauh AWS .

Jika template Anda mendefinisikan sumber daya sesuai dengan skemanya, pesan akan ditampilkan untuk mengonfirmasi bahwa sumber daya telah dibuat. (Jika sumber daya sudah ada, pesan mengonfirmasi bahwa sumber daya telah diperbarui.)

Setelah sumber daya dibuat, itu ditambahkan ke daftar di bawah judul tipe sumber daya.

5. Jika file Anda berisi kesalahan, pesan akan ditampilkan untuk menjelaskan bahwa sumber daya tidak dapat dibuat atau diperbarui. Pilih Lihat Log untuk mengidentifikasi elemen template yang perlu Anda perbaiki.

# Keamanan untuk AWS Toolkit for Visual Studio Code

## Topik

- [Perlindungan data di AWS Toolkit for Visual Studio Code](#)

## Perlindungan data di AWS Toolkit for Visual Studio Code

[Model tanggung jawab AWS bersama model](#) berlaku untuk perlindungan data di AWS Toolkit for Visual Studio Code. Seperti yang dijelaskan dalam model AWS ini, bertanggung jawab untuk melindungi infrastruktur global yang menjalankan semua AWS Cloud. Anda bertanggung jawab untuk mempertahankan kendali atas konten yang di-host pada infrastruktur ini. Anda juga bertanggung jawab atas tugas-tugas konfigurasi dan manajemen keamanan untuk layanan AWS yang Anda gunakan. Untuk informasi selengkapnya tentang privasi data, lihat [Privasi Data FAQ](#). Untuk informasi tentang perlindungan data di Eropa, lihat [Model Tanggung Jawab AWS Bersama dan](#) posting GDPR blog di Blog AWS Keamanan.

Untuk tujuan perlindungan data, kami menyarankan Anda melindungi Akun AWS kredensi dan mengatur pengguna individu dengan AWS IAM Identity Center atau AWS Identity and Access Management (IAM). Dengan cara itu, setiap pengguna hanya diberi izin yang diperlukan untuk memenuhi tanggung jawab tugasnya. Kami juga menyarankan supaya Anda mengamankan data dengan cara-cara berikut:

- Gunakan otentikasi multi-faktor (MFA) dengan setiap akun.
- Gunakan SSL/TLS untuk berkomunikasi dengan AWS sumber daya. Kami membutuhkan TLS 1.2 dan merekomendasikan TLS 1.3.
- Siapkan API dan pencatatan aktivitas pengguna dengan AWS CloudTrail.
- Gunakan solusi AWS enkripsi, bersama dengan semua kontrol keamanan default di dalamnya layanan AWS.
- Gunakan layanan keamanan terkelola lanjut seperti Amazon Macie, yang membantu menemukan dan mengamankan data sensitif yang disimpan di Amazon S3.
- Jika Anda memerlukan FIPS 140-3 modul kriptografi yang divalidasi saat mengakses AWS melalui antarmuka baris perintah atau, gunakan titik akhir. API FIPS Untuk informasi selengkapnya tentang FIPS titik akhir yang tersedia, lihat [Federal Information Processing Standard \(FIPS\) 140-3](#).

Kami sangat merekomendasikan agar Anda tidak pernah memasukkan informasi identifikasi yang sensitif, seperti nomor rekening pelanggan Anda, ke dalam tanda atau bidang isian bebas seperti bidang Nama. Ini termasuk saat Anda bekerja dengan AWS Toolkit for Visual Studio Code atau layanan AWS lainnya menggunakan konsolAPI,, AWS CLI, AWS SDKs atau. Data apa pun yang Anda masukkan ke dalam tanda atau bidang isian bebas yang digunakan untuk nama dapat digunakan untuk log penagihan atau log diagnostik. Jika Anda memberikan URL ke server eksternal, kami sangat menyarankan agar Anda tidak menyertakan informasi kredensial dalam URL untuk memvalidasi permintaan Anda ke server tersebut.

# Riwayat dokumen untuk Panduan AWS Toolkit for Visual Studio Code Pengguna

Tabel berikut menjelaskan perubahan penting dalam setiap rilis AWS Toolkit for Visual Studio Code. Untuk pemberitahuan tentang pembaruan dokumentasi ini, Anda dapat berlangganan [RSSumpnan](#).

Perubahan	Deskripsi	Tanggal
<a href="#">AWS Identity and Access Management (IAM) Pembaruan Access Analyzer</a>	Konten IAM Access Analyzer yang diperbarui untuk menyertakan API referensi baru.	Juli 10, 2024
<a href="#">AWS Identity and Access Management (IAM) Access Analyzer</a>	Menambahkan topik panduan pengguna baru untuk IAM Access Analyzer.	23 Mei 2024
<a href="#">Connect ke aliran AWS otorisasi diperbarui</a>	Alur otorisasi diperbarui untuk mencerminkan perubahan pada proses autentikasi dan pemisahan Amazon Q dari AWS Toolkit for Visual Studio Code	April 30, 2024
<a href="#">Ekstensi Amazon Q untuk Kode VS</a>	Mulai 30 April 2024, CodeWhisperer sekarang menjadi bagian dari Amazon Q dan Amazon Q tersedia sebagai ekstensi untuk VS Code.	April 30, 2024
<a href="#">Support untuk Virtual Private Cloud di Lingkungan Dev</a>	Konten yang diperbarui yang mencakup perubahan UI untuk didukung VPC di Lingkungan Pengembang.	Januari 21, 2024

---

<a href="#">Komposer Aplikasi</a>	Ditambahkan topik Application Composer baru ke Panduan AWS Toolkit for Visual Studio Code Pengguna.	28 November 2023
<a href="#">SSODukungan untuk CodeCatalyst</a>	Konten yang diperbarui untuk mencakup dukungan Pusat IAM Identitas untuk CodeCatalyst dan Lingkungan Pengembang.	17 November 2023
<a href="#">Menambahkan tautan unduhan VS Code dan Toolkit</a>	Konten yang diperbarui dengan tautan unduhan untuk VS Code dan AWS Toolkit for Visual Studio Code.	1 November 2023
<a href="#">Topik Amazon Redshift</a>	Menambahkan topik Amazon Redshift baru ke AWS Toolkit for Visual Studio Code Panduan Pengguna.	17 Oktober 2023
<a href="#">Connect ke aliran AWS otorisasi diperbarui</a>	Alur otorisasi diperbarui untuk fokus pada metode otentikasi khusus layanan.	September 29, 2023
<a href="#">Panduan pengguna yang dibuat: Buat template CloudFormation</a>	Membuat panduan pengguna baru yang menjelaskan cara Membuat CloudFormation template menggunakan Toolkit for VS Code	Desember 17, 2021
<a href="#">Pembaruan UI Kecil</a>	Memperbarui teks yang ada untuk “Pratinjau Status Mesin” ke “Render grafik” agar lebih cocok dengan UI.	Desember 14, 2021

<a href="#">Panduan pengguna yang dibuat untuk Amazon Elastic Container Service Exec</a>	Ini adalah ikhtisar dari Amazon ECS Exec.	13 Desember 2021
<a href="#">Panduan pengguna yang dibuat untuk layanan AWS IoT Toolkit for VS Code</a>	Panduan pengguna ini dimaksudkan untuk membantu Anda mulai menggunakan AWS IoT layanan untuk Toolkit for VS Code.	22 November 2021
<a href="#">Support untuk fitur eksperimental</a>	Menambahkan dukungan untuk mengaktifkan fitur eksperimental untuk AWS layanan.	Oktober 14, 2021
<a href="#">Support untuk AWS sumber daya</a>	Menambahkan dukungan untuk mengakses jenis sumber daya bersama dengan opsi antarmuka untuk membuat, mengedit, dan menghapus sumber daya.	Oktober 14, 2021
<a href="#">Ikhtisar ECR layanan Amazon untuk AWS Toolkit for Visual Studio Code</a>	Menambahkan ikhtisar dan panduan untuk fitur dan fungsi ECR layanan Amazon yang dapat diakses di VS Code	Oktober 14, 2021
<a href="#">Support untuk ARM64 lingkungan</a>	Anda sekarang dapat menjalankan aplikasi tanpa server di lingkungan emulasi ARM64 berbasis serta di lingkungan berbasis x86_64.	1 Oktober 2021
<a href="#">AWS Serverless Application</a>	Menambahkan dukungan untuk menjalankan AWS SAM aplikasi di ARM64 platform	30 September 2021

---

<a href="#">Pemutakhiran format bagian Node.js</a>	Per umpan balik pelanggan, pemformatan yang diperbarui untuk TypeScript Node.js/.	Agustus 12, 2021
<a href="#">Dukungan App Runner</a>	Menambahkan dukungan untuk AWS App Runner untuk AWS Toolkit for Visual Studio Code.	Agustus 11, 2021
<a href="#">Fungsi Debugging Go</a>	Menambahkan dukungan untuk men-debug fungsi Go lokal.	10 Mei 2021
<a href="#">Debugging fungsi Java</a>	Ditambahkan dukungan untuk debugging fungsi Java lokal.	22 April 2021
<a href="#">YAMLdukungan untuk AWS Step Functions</a>	Menambahkan YAML dukungan untuk AWS Step Functions.	4 Maret 2021
<a href="#">Mendebug sumber daya Amazon API Gateway</a>	Menambahkan dukungan untuk men-debug sumber daya Amazon API Gateway lokal.	1 Desember 2020
<a href="#">APIGerbang Amazon</a>	Menambahkan dukungan untuk Amazon API Gateway.	1 Desember 2020
<a href="#">AWS Serverless Application</a>	Menambahkan dukungan untuk gambar kontainer Lambda dengan aplikasi tanpa server.	1 Desember 2020
<a href="#">AWS Systems Manager dukungan</a>	Menambahkan dukungan untuk dokumen Otomasi Systems Manager.	30 September 2020
<a href="#">CloudWatch Log</a>	Ditambahkan dukungan untuk CloudWatch Log.	24 Agustus 2020

---

<a href="#">Amazon S3</a>	Ditambahkan dukungan untuk Amazon S3.	30 Juli 2020
<a href="#">AWS Step Functions dukungan</a>	Menambahkan dukungan untuk AWS Step Functions.	31 Maret 2020
<a href="#">Konten Keamanan</a>	Menambahkan konten keamanan.	6 Februari 2020
<a href="#">Bekerja dengan Amazon EventBridge Schemas</a>	Ditambahkan dukungan untuk Amazon EventBridge Schemas	1 Desember, 2019
<a href="#">AWS CDK</a>	Pratinjau rilis AWS CDK layanan.	25 November 2019
<a href="#">Menggunakan proses kredensi eksternal</a>	Menambahkan informasi tentang menggunakan proses kredensi eksternal untuk mendapatkan AWS kredensial.	25 Septbucket 2019
<a href="#">Menggunakan IntelliSense untuk file definisi tugas</a>	IntelliSense dukungan ditambahkan untuk bekerja dengan file Definisi ECS tugas Amazon.	24 September 2019
<a href="#">Panduan Pengguna untuk AWS Toolkit for Visual Studio Code</a>	Rilis untuk ketersediaan umum.	11 Juli 2019
<a href="#">Panduan Pengguna untuk AWS Toolkit for Visual Studio Code</a>	Memperbarui struktur dokumen untuk kejelasan dan kemudahan penggunaan.	3 Juli 2019
<a href="#">Instalasi AWS Toolkit for Visual Studio Code</a>	Menambahkan informasi tentang menginstal bahasa SDKs untuk mendukung berbagai toolchain.	12 Juni 2019

[Konfigurasi rantai alat  
Anda](#)

Menambahkan informasi  
tentang mengkonfigurasi  
berbagai toolchain.

12 Juni 2019

[Rilis Awal](#)

Rilis awal panduan pengguna  
untuk AWS Toolkit for Visual  
Studio Code.

28 Maret 2019

Terjemahan disediakan oleh mesin penerjemah. Jika konten terjemahan yang diberikan bertentangan dengan versi bahasa Inggris aslinya, utamakan versi bahasa Inggris.