



Panduan Pengguna

# AWS Toolkit for VS Code



# AWS Toolkit for VS Code: Panduan Pengguna

Copyright © 2026 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 merendahkan atau mendiskreditkan Amazon. Semua merek dagang lain yang tidak dimiliki oleh Amazon merupakan hak milik masing-masing pemiliknya, yang mungkin atau mungkin tidak terafiliasi, 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
Konfigurasi toolchain untuk .NET Core .....	10
Konfigurasi toolchain untuk Node.js .....	10
Konfigurasi toolchain untuk Python .....	11
Konfigurasi toolchain untuk Java .....	11
Konfigurasi toolchain untuk Go .....	12
Menggunakan toolchain Anda .....	12
Otentikasi dan akses .....	13
Pusat Identitas IAM .....	13
Kredensial IAM .....	13
Membuat pengguna IAM .....	14
Membuat file kredensial bersama dari AWS Toolkit for Visual Studio Code .....	15
Tambahkan profil kredensi tambahan .....	16
AWS ID Pembangun .....	17

Menggunakan proses kredensi eksternal .....	17
Memperbarui firewall dan gateway .....	17
AWS Toolkit for Visual Studio Code Titik akhir .....	17
Titik akhir plugin Amazon Q .....	18
Titik akhir Pengembang Amazon Q .....	18
Titik Akhir Transformasi Kode Q Amazon .....	19
Titik akhir otentikasi .....	19
Titik Akhir Identitas .....	19
Telemetry .....	20
Referensi .....	20
Bekerja dengan AWS .....	22
Fitur eksperimental .....	23
AWS Penjelajah .....	23
AWS Dokumen .....	24
Memulai dengan AWS Dokumen .....	25
Melihat dokumentasi, pelengkapan otomatis, dan validasi di VS Code .....	25
Amazon CodeCatalyst .....	26
Apa itu Amazon CodeCatalyst? .....	26
Mulai menggunakan CodeCatalyst .....	27
Bekerja dengan CodeCatalyst sumber daya .....	27
Bekerja dengan Lingkungan Dev .....	31
Pemecahan masalah .....	34
Amazon API Gateway .....	35
AWS App Runner .....	36
Prasyarat .....	36
Harga .....	39
Membuat layanan App Runner .....	40
Mengelola layanan App Runner .....	43
AWS Pembuat Aplikasi .....	45
Bekerja dengan AWS Application Builder .....	46
AWS Komposer Infrastruktur .....	50
Bekerja dengan Komposer AWS Infrastruktur .....	50
AWS CDK .....	51
AWS CDK aplikasi .....	52
CloudFormation tumpukan .....	54
Menghapus tumpukan CloudFormation .....	54

Buat CloudFormation template .....	55
CloudWatch Log Amazon .....	57
Melihat grup CloudWatch log dan aliran log .....	57
Bekerja dengan peristiwa CloudWatch log .....	58
Mencari grup log .....	60
CloudWatch Log Ekor Hidup .....	63
Amazon DocumentDB .....	64
Bekerja dengan Amazon DocumentDB .....	65
Amazon EC2 .....	70
Bekerja dengan Amazon EC2 .....	71
Pemecahan Masalah Amazon EC2 .....	79
Amazon ECR .....	81
Bekerja dengan Amazon ECR .....	82
Membuat layanan App Runner .....	93
Amazon ECS .....	94
Menggunakan IntelliSense untuk file definisi tugas .....	94
Amazon ECS Exec .....	95
Amazon EventBridge .....	98
Bekerja dengan Amazon EventBridge Schemas .....	98
AWS Penganalisis Akses IAM .....	100
Bekerja dengan AWS IAM Access Analyzer .....	101
AWS IoT .....	105
AWS IoT prasyarat .....	105
AWS IoT Hal-hal .....	105
AWS IoT sertifikat .....	107
AWS IoT kebijakan .....	110
AWS Lambda Fungsi .....	113
Bekerja dengan AWS Lambda Fungsi .....	114
AWS Lambda console ke IDE .....	120
AWS Lambda LocalStack dukungan .....	121
Debugging jarak jauh Lambda .....	127
Amazon Redshift .....	137
Bekerja dengan Amazon Redshift .....	138
Amazon S3 .....	143
Bekerja dengan sumber daya S3 .....	143
Bekerja dengan objek S3 .....	145

Studio SageMaker Terpadu Amazon .....	149
AWS Serverless Application .....	149
Memulai .....	149
Bekerja dengan Serverless Land .....	157
Menjalankan dan men-debug fungsi Lambda langsung dari kode .....	159
Menjalankan dan men-debug sumber daya Amazon API Gateway lokal .....	164
Opsi konfigurasi untuk men-debug aplikasi nirserver .....	168
Pemecahan masalah .....	175
AWS Systems Manager .....	177
Asumsi dan prasyarat .....	178
Izin IAM untuk dokumen Otomasi Systems Manager .....	178
Membuat dokumen Systems Manager Automation baru .....	179
Membuka dokumen Systems Manager Automation yang sudah ada .....	180
Mengedit dokumen Otomasi Systems Manager .....	180
Menerbitkan dokumen Otomasi Systems Manager .....	181
Menghapus dokumen Otomasi Systems Manager .....	182
Menjalankan dokumen Otomasi Systems Manager .....	182
Pemecahan masalah .....	183
AWS Step Functions .....	183
Bekerja dengan Step Functions .....	184
Bekerja dengan Workflow Studio .....	188
Komposer Ancaman .....	192
Bekerja dengan Threat Composer .....	192
Sumber Daya .....	193
Izin IAM untuk mengakses sumber daya .....	194
Menambah dan berinteraksi dengan sumber daya yang ada .....	195
Membuat dan mengedit sumber daya .....	197
Pemecahan masalah .....	199
Memecahkan masalah praktik terbaik .....	199
Profil... tidak dapat ditemukan di file konfigurasi .....	200
Skema SAM json: tidak dapat mengubah skema dalam file template.yaml .....	201
Keamanan .....	202
Perlindungan data .....	202
Riwayat dokumen .....	204
.....	ccxii

# 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 [ekstensi IDE 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 JetBrains link

- 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 Ini membuat AWS layanan dan sumber daya Anda tersedia, langsung dari lingkungan pengembangan terintegrasi VS Code (IDE) Anda.

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

Topik

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

## Memasang AWS Toolkit for Visual Studio Code

### Prasyarat

Untuk mulai bekerja dengan AWS Toolkit for Visual Studio Code dari VS Code, persyaratan 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 IDE Anda. 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 IDE Anda 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 — SDK yang relevan 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>
  - SDK Python: <https://www.python.org/downloads>
  - SDK Jawa: <https://aws.amazon.com/corretto/>
  - Pergi SDK: <https://golang.org/doc/install>
- AWS SAM CLI - Ini adalah alat AWS CLI 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 untuk AWS akun dari [portal AWS pendaftaran](#). Untuk informasi lebih lanjut tentang cara 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 AWS IAM Identity Center

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

2. Dari panel Masuk dengan Pusat Identitas IAM, masukkan URL Mulai untuk organisasi Anda. URL ini disediakan untuk Anda oleh admin atau help desk di perusahaan Anda.
3. Pilih AWS Wilayah Anda dari menu tarik-turun. Ini adalah AWS wilayah yang meng-host direktori identitas Anda.
4. Pilih tombol Lanjutkan dan konfirmasikan 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 Kredensial IAM

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

### Mengautentikasi dan terhubung dengan Kredensial IAM

1. Dari panel Masuk AWS Toolkit, pilih IAM Credential, lalu pilih tombol Lanjutkan 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 panel Masuk dengan SSO, 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 IAM Identity Center

- Dari panel Masuk AWS Toolkit, pilih tab Workforce, lalu pilih tombol Lanjutkan untuk melanjutkan.
- Dari panel Masuk dengan Pusat Identitas IAM, masukkan URL Mulai untuk organisasi Anda. URL ini disediakan untuk Anda oleh admin atau help desk di perusahaan Anda.
- Pilih AWS Wilayah Anda dari menu tarik-turun. Ini adalah AWS wilayah yang meng-host direktori identitas Anda.
- Pilih tombol Lanjutkan dan konfirmasikan 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 rantai alat Anda untuk berbagai bahasa.

### Konfigurasi toolchain untuk .NET Core

1. Pastikan Anda telah menginstal AWS Toolkit for VS [Code](#).
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 berisitemplate.yaml.

### Konfigurasi toolchain untuk Node.js

1. Pastikan Anda telah menginstal AWS Toolkit for VS [Code](#).
2. Buka AWS SAM aplikasi, atau [buat satu](#).
3. Buka folder yang berisitemplate.yaml.

**Note**

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

## Konfigurasi toolchain untuk Python

1. Pastikan Anda telah menginstal AWS Toolkit for VS [Code](#).
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 berisitemplate.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 dari berikut ini:
  - Kulit bash: `./venv/Scripts/activate`
  - PowerShell: `./venv/Scripts/Activate.ps1`

## Konfigurasi toolchain untuk Java

1. Pastikan Anda telah menginstal AWS Toolkit for VS [Code](#).
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.

## Konfigurasi toolchain untuk Go

1. Pastikan Anda telah menginstal AWS Toolkit for VS [Code](#).
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 men-debug runtime Go1.15+.

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

## Menggunakan toolchain Anda

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

# Otentikasi dan akses untuk AWS Toolkit for Visual Studio Code

Anda tidak perlu mengautentikasi 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 mengautentikasi 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 Kredensi IAM](#)
- [AWS Builder ID untuk pengembang](#)
- [Menggunakan proses kredensi eksternal](#)
- [Memperbarui firewall dan gateway untuk memungkinkan akses](#)

## 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 IAM Identity Center for Software Development Kits (SDKs), lihat bagian [otentikasi IAM Identity Center](#) pada Panduan Referensi Alat AWS SDKs dan Alat.

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

## AWS Kredensi IAM

AWS Autentikasi kredensi IAM dengan AWS akun Anda melalui kunci akses yang disimpan secara lokal.

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

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

### Important

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

- Jika Anda telah menetapkan kredensial IAM 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 autentikasi IAM Identity Center. Untuk informasi tambahan tentang praktik terbaik AWS IAM, lihat [praktik terbaik Keamanan di bagian IAM](#) dari Panduan Pengguna AWS Identity and Access Management.
- Untuk menghindari risiko keamanan, jangan gunakan pengguna IAM 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 Identitas IAM?](#) dalam AWS IAM Identity Center User Guide.

## Membuat pengguna IAM

Sebelum Anda dapat mengatur AWS Toolkit for Visual Studio Code untuk mengautentikasi dengan AWS akun Anda, Anda harus menyelesaikan Langkah 1: Buat pengguna IAM Anda dan Langkah 2: Dapatkan kunci akses Anda di [Authenticate using long-term credentials](#) topic in the and Tools Reference Guide.AWS SDKs

### 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 kredensial 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 IAM Credential, lalu pilih tombol Lanjutkan 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 kredensi eksternal

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

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

## Memperbarui firewall dan gateway untuk memungkinkan akses

Jika Anda memfilter akses ke AWS domain atau titik akhir URL tertentu dengan menggunakan solusi pemfilteran konten web, titik akhir berikut harus diizinkan terdaftar untuk mengakses semua layanan dan fitur yang tersedia melalui Amazon Q. AWS Toolkit for Visual Studio Code

## AWS Toolkit for Visual Studio Code Titik akhir

Berikut ini adalah daftar titik akhir AWS Toolkit for Visual Studio Code tertentu dan referensi yang perlu diizinkan terdaftar.

### Titik akhir

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

### File yang dihosting

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

## Dukungan skema

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

## cSharpSamScrip instalasi debug

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

## Titik akhir plugin Amazon Q

Berikut ini adalah daftar titik akhir dan referensi khusus plugin Amazon Q yang perlu diizinkan terdaftar.

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

## Titik akhir Pengembang Amazon Q

Berikut ini adalah daftar titik akhir dan referensi khusus Pengembang Amazon Q yang perlu diizinkan terdaftar.

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

## Titik Akhir Transformasi Kode Q Amazon

Berikut ini adalah daftar titik akhir dan referensi spesifik Amazon Q Code Transform yang perlu diizinkan terdaftar.

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

## Titik akhir otentikasi

Berikut ini adalah daftar endpoint otentikasi dan referensi yang perlu diizinkan terdaftar.

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

## Titik Akhir Identitas

Daftar berikut berisi titik akhir yang spesifik untuk identitas, seperti AWS IAM Identity Center dan AWS Builder ID.

### AWS IAM Identity Center

Untuk detail tentang titik akhir yang diperlukan untuk Pusat Identitas IAM, lihat topik [Aktifkan Pusat Identitas IAM](#) di Panduan Pengguna. AWS IAM Identity Center

## Pusat Identitas IAM Perusahaan

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

## AWS ID Pembangun

```
https://view.awsapps.com/start (must be blocked to disable individual tier)
https://codewhisperer.us-east-1.amazonaws.com and q.us-east-1.amazonaws.com (should be
permitted)
```

## Telemetri

Berikut ini adalah titik akhir khusus Telemetri yang perlu diizinkan terdaftar.

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

## Referensi

Berikut ini adalah daftar referensi endpoint.

```
idtoolkits-hostedfiles.amazonaws.com.
cognito-identity.us-east-1.amazonaws.com.
amazonwebservices.gallery.vsassets.io.
eu-west-1.prod.pr.analytics.console.aws.a2z.com.
prod.pa.cdn.uis.awsstatic.com.
portal.sso.eu-west-1.amazonaws.com.
log.sso-portal.eu-west-1.amazonaws.com.
```

```
prod.assets.shortbread.aws.dev.  
prod.tools.shortbread.aws.dev.  
prod.log.shortbread.aws.dev.  
a.b.cdn.console.awsstatic.com.  
assets.sso-portal.eu-west-1.amazonaws.com.  
oidc.eu-west-1.amazonaws.com.  
aws-toolkit-language-servers.amazonaws.com.  
aws-language-servers.us-east-1.amazonaws.com.  
idtoolkits.amazonwebservices.com.
```

# 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](#)
- [AWS Dokumen](#)
- [Amazon CodeCatalyst untuk Kode VS](#)
- [Bekerja dengan Amazon API Gateway](#)
- [Menggunakan AWS App Runner dengan AWS Toolkit for Visual Studio Code](#)
- [AWS Pembuat Aplikasi](#)
- [AWS Komposer Infrastruktur](#)
- [AWS CDK untuk VS Code](#)
- [Bekerja dengan AWS CloudFormation tumpukan](#)
- [Bekerja dengan CloudWatch Log dengan menggunakan AWS Toolkit for Visual Studio Code](#)
- [Amazon DocumentDB](#)
- [Amazon Elastic Compute Cloud](#)
- [Bekerja dengan Amazon Elastic Container Registry](#)
- [Bekerja dengan Amazon Elastic Container Service](#)
- [Bekerja dengan Amazon EventBridge](#)
- [AWS Penganalisis Akses IAM](#)
- [Bekerja dengan AWS IoT di AWS Toolkit for Visual Studio Code](#)
- [AWS Lambda Fungsi](#)
- [Amazon Redshift di Toolkit untuk VS Code](#)
- [Bekerja dengan Amazon S3](#)
- [Amazon SageMaker Unified Studio untuk VS Code](#)
- [Bekerja dengan aplikasi tanpa server](#)

- [Bekerja dengan dokumen Otomasi Systems Manager](#)
- [AWS Step Functions](#)
- [Bekerja dengan Threat Composer](#)
- [Bekerja dengan sumber daya](#)

## Bekerja dengan fitur eksperimental

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

### 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 AWS layanan tertentu di bagian AWS Toolkit pada panel Pengaturan di IDE Kode VS Anda.

1. Untuk mengedit AWS pengaturan di VS Code, pilih File, Preferences, Settings.
2. Di panel Pengaturan, perluas Ekstensi dan pilih AWS Toolkit.
3. Di bawah AWS: Eksperimen, 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

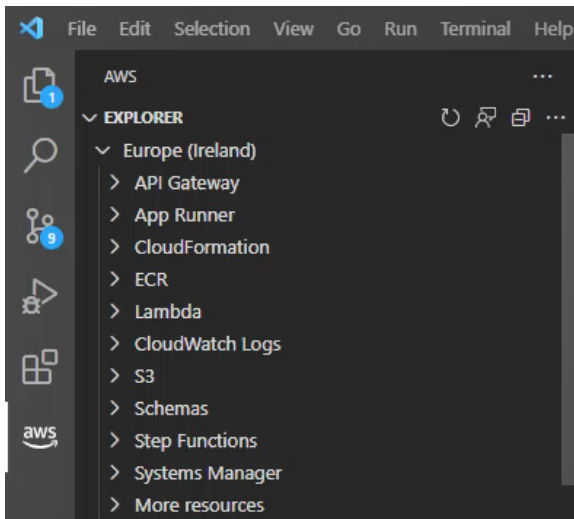
AWS Explorer memberi Anda pandangan tentang beberapa AWS layanan yang dapat Anda gunakan saat menggunakan AWS Toolkit for Visual Studio Code.

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

## AWS Dokumen

AWS Toolkit for Visual Studio Code Mendukung AWS Serverless Application Model JSON Schema forAWS SAM templates, meningkatkan pengalaman penulisan template dengan mengaktifkan definisi, pelengkapan otomatis, dan validasi langsung di VS Code. AWS Dokumen mendukung

semua AWS SAM dan CloudFormation sumber daya. Untuk detail tambahan, lihat sumber daya berikut:

- Untuk informasi spesifik tentang Skema JSON, lihat situs web JSON Schema [JSON-schema.org](https://json-schema.org).
- Untuk informasi tambahan tentang AWS SAM template, lihat topik [anatomi AWS SAM template](#) di Panduan AWS Serverless Application ModelPengembang.
- Untuk informasi tambahan tentang AWS sumber daya dan jenis properti, lihat topik [referensi jenis AWS sumber daya dan properti](#) di Panduan CloudFormationPengguna.
- Untuk informasi rinci tentang AWS SAM skema yang digunakan oleh AWS Toolkit, lihat [AWS Serverless Application Model](#) skema di repositori. AWS GitHub

## Memulai dengan AWS Dokumen

Untuk mulai bekerja dengan AWS Documents in VS Code, instal AWS Toolkit for Visual Studio Code ekstensi dari IDE Anda atau [VS Code Marketplace](#), dan buka AWS SAM template apa pun.

## Melihat dokumentasi, pelengkapan otomatis, dan validasi di VS Code

Melihat dokumentasi, pelengkapan otomatis, dan validasi adalah fitur yang disertakan dengan Toolkit. AWS Lihat gambar di bawah ini untuk contoh seperti apa fitur ini di VS Code.

- Untuk melihat dokumentasi dari AWS SAM template terbuka Anda, arahkan kursor ke entri baris dalam dokumen.
- Untuk pelengkapan otomatis, mulailah mengetik AWS SAM template Anda untuk mengaktifkan pop-up dengan saran berdasarkan masukan Anda.
- AWS SAM Template Anda secara otomatis dipindai untuk validasi dan kesalahan disorot oleh ikon bola lampu yang dapat Anda pilih untuk saran tambahan.

Lihat gambar di bawah ini untuk contoh seperti apa fitur ini di VS Code.

```

    AUTHORIZER: myCognitoAuthMultipleUserPools
  WithCognitoMultipleUserPoolsAuthorizerAnyMethod:

```

```

  Type: Api

```

```

  Pr

```

### RestApiId

Identifier of a RestApi resource, which must contain an operation with the given path and method. Typically, this is set to reference an `AWS::Serverless::Api` resource defined in this template.

**With Type Pr** If you don't define this property, AWS SAM creates a default `AWS::Serverless::Api` resource using a generated `OpenApi` document. That resource contains a union of all paths and methods defined by `Api` events in the same template that do not specify a `RestApiId`.

This cannot reference an `AWS::Serverless::Api` resource defined in another template.

Type: String

Required: No

*AWS CloudFormation compatibility:* This property is unique to AWS SAM and doesn't have an AWS CloudFormation equivalent.

**With Type Pr** Source: `schema.json`

```

  RestApiId: !Ref MyApi

```

```

  Path: /any/lambdatoken

```

```

  Method: any

```

## 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 menggunakan AWS Toolkit untuk meluncurkan, lingkungan komputasi virtual Dev Environments yang menjalankan VS Code. 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 Panduan Pengguna. CodeCatalyst

### 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 ikhtisar 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 penjelajah ALAT PENGEMBANG.
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 Repository 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 penjelajah ALAT PENGEMBANG.
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 secara 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 penjelajah ALAT PENGEMBANG.
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 penjelajah ALAT PENGEMBANG.
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 untuk YAMB yang dapat digunakan untuk menentukan konfigurasi untuk Lingkungan Dev. Setiap Dev Environment 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 IDE Anda. 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](https://devfile.io).

1. Dari Toolkit for VS Code, perluas penjelajah ALAT PENGEMBANG.
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 penjelajah ALAT PENGEMBANG.
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 penjelajah ALAT PENGEMBANG.
2. Perluas CodeCatalyst dan pilih Open Settings untuk membuka tampilan Dev Environment Settings, untuk Dev Environment 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 tersebut.
- 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 AWS akun tersambung menggunakan 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 Explorer, pilih API Gateway untuk memperluas menu. Sumber daya API Gateway jarak jauh terdaftar.
2. Temukan sumber daya API Gateway yang ingin Anda panggil, buka menu konteksnya (klik kanan), lalu pilih Invoke on. AWS
3. Dalam formulir parameter, tentukan parameter pemanggilan.
4. Untuk menjalankan resource API Gateway jarak jauh, pilih Invoke. Hasilnya di-deployed dalam tampilan VS Code Output.

# 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 dari 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 ECR) 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 `AWSServiceRoleForAppRunner` 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 Konsol Manajemen AWS 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 repositori.](#) <https://console.aws.amazon.com/ecr/>
- [Registri publik Amazon ECR: Salin URI untuk repositori publik yang menggunakan Galeri Publik Amazon ECR di](#) <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 ditautkan. GitHub 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, Peran, AppRunner ECRAccess 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 dalam 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.

## AWS Pembuat Aplikasi

AWS Application Builder for the AWS Toolkit for Visual Studio Code adalah panduan Anda untuk membangun proyek secara visual, mengulanginya secara lokal, dan menerapkan aplikasi Anda ke AWS.

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

Topik

- [Bekerja dengan AWS Application Builder](#)

## Bekerja dengan AWS Application Builder

Bagian berikut menjelaskan cara mengakses AWS Application Builder di AWS Toolkit for Visual Studio Code. Dengan Application Builder, Anda dapat membangun proyek secara visual, mengulanginya secara lokal, dan menerapkannya. AWS Untuk ikhtisar fitur dan kasus penggunaan potensial untuk Pembuat Aplikasi dan AWS Lambda pengalaman lokal Anda, lihat YouTube video [AWS Pengembang\\*New\\* Pengalaman IDE Lambda Lambda AWS Lambda!](#) .

### Bekerja dengan explorer AWS Application Builder

Untuk mengakses Application Builder di AWS Toolkit, buka AWS Toolkit di VS Code, lalu perluas AWS Application Builder explorer. Penjelajah AWS Application Builder berisi tautan untuk membuka Walkthrough of Application Builder di tab editor VS Code, dan menampilkan folder dalam ruang kerja VS Code Anda saat ini yang berisi sumber daya terkait Pembuat AWS Aplikasi.

Dari explorer Application Builder di AWS Toolkit, ada 4 project-folder-level tindakan yang dapat diakses dari ikon tombol yang terletak di sebelah folder proyek Anda atau dengan membuka menu konteks untuk (mengklik kanan) folder proyek:

- **Buka File Template:** Membuka file template Anda di penjelajah VS Code.
- **Buka dengan Infrastructure Composer:** Membuka file template Anda dengan AWS Infrastructure Composer di editor VS Code. Untuk informasi rinci tentang bekerja dengan AWS Infrastructure Composer, lihat topik [Apa itu AWS Infrastructure Composer](#) di Panduan Pengembang AWS Infrastructure Composer.
- **Build SAM Template:** Membuka dialog Specify parameters for build di AWS Toolkit. Anda dapat memilih untuk Menentukan flag build untuk build atau Gunakan nilai default dari samconfig. Untuk informasi rinci tentang AWS SAM template, lihat topik [anatomy Template](#) di Panduan AWS Serverless Application ModelPengembang.
- **Menyebarkan Aplikasi SAM:** Membuka dialog perintah Select deployment di VS Code di mana Anda dapat memilih untuk Menyebarkan aplikasi atau Sinkronisasi, untuk memperbarui aplikasi yang telah Anda gunakan. Untuk informasi terperinci tentang penerapan AWS SAM aplikasi, lihat topik [Menerapkan aplikasi dan sumber daya Anda](#) di Panduan AWS Serverless Application ModelPengembang.

Ada 2 tindakan yang dapat diakses dari ikon tombol yang terletak di sebelah AWS Lambda fungsi di folder proyek Anda atau dengan mengklik kanan AWS Lambda fungsi:

- Konfigurasi Pemanggilan dan Debug Lokal: Membuka formulir Local Invoke dan Debug Configuration di editor VS Code Anda. Dengan formulir ini Anda dapat membuat, mengedit, dan menjalankan launch-configs tipe: `aws-sam` Untuk informasi tambahan tentang konfigurasi Debug SAM, lihat topik [Opsi konfigurasi untuk men-debug aplikasi tanpa server](#) di Panduan Pengguna ini.

#### Note

Saat ini, men-debug aplikasi.NET Core pada ARM64 arsitektur tidak didukung oleh VS Code. Jika Anda mencoba men-debug aplikasi.NET Core, kesalahan berikut akan ditampilkan:

```
The vsdbg debugger does not currently support the arm64 architecture. Function will run locally without debug.
```

Untuk detail tambahan tentang masalah ini, lihat masalah [VSCode-csharp](#) ini di repositori. DotNet GitHub

- Open Function Handler: Membuka file proyek Anda yang berisi fungsi handler.

Ada 2 tindakan tambahan yang tersedia untuk AWS Lambda fungsi yang diterapkan.

- Pemanggilan jarak jauh: membuka menu konfigurasi pemanggilan jarak jauh di editor VS Code.
- Log pencarian: Membuka dialog Log pencarian di VS Code.

## Panduan Pembuat Aplikasi

Walkthrough of Application Builder adalah panduan step-by-step interaktif yang membawa Anda melalui proses membangun aplikasi baru dengan AWS Application Builder. Anda dapat mengakses Walkthrough of Application Builder dari dua tempat: Application Builder explorer di AWS Toolkit for Visual Studio Code dan tab VS Code Welcome. Ketika Anda memilih Walkthrough of Application Builder dari Application Builder explorer di AWS Toolkit, itu akan membuka Walkthrough of Application Builder di tab VS Code Welcome di jendela VS Code Editor.

Panduan Pembuat Aplikasi terdiri dari 5 bagian utama:

### 1. Peningkatan

Bagian Instalasi memeriksa untuk melihat apakah Anda telah menginstal AWS CLI alat yang diperlukan oleh Application Builder dan alat opsional lainnya. Jika Anda tidak memiliki alat yang diperlukan atau alat Anda kedaluwarsa, Anda dipandu melalui proses menginstal versi yang benar.

Untuk melihat apakah Anda memiliki alat yang benar AWS CLI dan opsional diinstal, pilih tombol untuk AWS CLI atau alat lain yang ingin Anda uji. Setelah memilih tombol, pembaruan Log AWS Toolkit Anda dan Kode VS menampilkan pesan peringatan dengan status alat Anda. Jika Anda perlu menginstal atau memperbarui alat Anda, Panduan Pembuat Aplikasi diperbarui dengan instruksi dan sumber daya yang Anda butuhkan untuk melanjutkan.

Untuk informasi rinci tentang menginstal AWS CLI, lihat [Menginstal atau memperbarui ke versi terbaru dari AWS CLI](#) topik dalam Panduan AWS CLIPengembang. Untuk informasi rinci tentang menginstal AWS SAM CLI, lihat topik Instal [CLI di Panduan AWS SAM Pengembang](#) CLI AWS SAM .

## 2. Pilih template aplikasi Anda

Bagian Choose your application template memandu Anda melalui proses membangun aplikasi baru dari template.

Untuk memilih template dan menginisialisasi aplikasi Anda, selesaikan langkah-langkah berikut.

1. Dari Walkthrough of Application Builder, pilih bagian Pilih template aplikasi Anda untuk menampilkan daftar opsi template di layar Anda.
2. Pilih template dari daftar, lalu pilih tombol Inisialisasi proyek Anda untuk membuka dialog VS Code.
3. Selesaikan langkah-langkah dalam dialog VS Code untuk menginisialisasi aplikasi baru Anda.
4. AWS Toolkit log update dengan status aplikasi Anda selama proses inisialisasi.
5. Untuk melihat aplikasi Anda di explorer Application Builder, pilih ikon Refresh Application Builder Explorer untuk memperbarui explorer dengan perubahan Anda.

## 3. Iterasi secara lokal

Bagian Iterate secara lokal berisi contoh gambar yang menunjukkan bagaimana Anda dapat melakukan iterasi dengan fitur Pembuat Aplikasi yang tersedia di penjelajah Kode VS dan AWS Toolkit.

Untuk informasi tambahan tentang semua fitur Pembuat Aplikasi yang tersedia di penjelajah Kode VS dan AWS Toolkit, lihat bagian Bekerja dengan Application Builder explorer, yang terletak di topik Panduan Pengguna ini.

## 4. Menyebarkan ke AWS

AWS Bagian Deploy to berisi informasi tentang cara mengonfigurasi kredensi Anda untuk terhubung dengan AWS tujuan penerapan aplikasi Anda dan contoh cara menerapkan aplikasi Anda dengan Application Builder.

Untuk terhubung AWS dengan kredensi Anda yang ada dari Walkthrough of Application Builder, lengkapi salah satu prosedur berikut.

Tenaga Kerja: Masuk AWS dengan sistem masuk tunggal.

1. Dari AWS bagian Deploy to di Walkthrough of Application Builder, pilih tombol Configure credentials untuk membuka menu AWS: LOGIN di penjelajah Toolkit. AWS
2. Dari menu AWS: LOGIN, pilih Workforce, lalu pilih tombol Continue untuk melanjutkan.
3. Masukkan URL Mulai Anda ke bidang yang disediakan, pilih AWS Wilayah Anda dari menu tarik-turun, lalu pilih tombol Lanjutkan untuk melanjutkan.
4. Dari jendela pop-up VS Code, konfirmasi bahwa Anda ingin membuka situs AWS Otentikasi di browser default Anda.
5. Dari browser default Anda, selesaikan langkah-langkah otentikasi, Anda diberi tahu saat otentikasi selesai dan aman untuk menutup jendela browser Anda.

IAM Credentials: Simpan kunci untuk digunakan dengan alat. AWS CLI

1. Dari AWS bagian Deploy to di Walkthrough of Application Builder, pilih tombol Configure credentials untuk membuka menu AWS: LOGIN di penjelajah Toolkit. AWS
2. Dari menu AWS: LOGIN, pilih IAM Credentials, lalu pilih tombol Continue untuk melanjutkan.
3. Masukkan Nama Profil ke bidang yang disediakan, lalu masukkan nama Anda **Access Key** dan **Secret Key**, lalu pilih tombol Lanjutkan untuk melanjutkan.
4. VS Code menampilkan status otentikasi Anda, memberi tahu Anda jika otentikasi selesai atau kredensialnya tidak valid.

Untuk informasi mendetail tentang mengonfigurasi kredensi Anda untuk penerapan dengan AWS CLI, lihat [Mengonfigurasi AWS CLI topik di Panduan Pengembang](#). AWS CLI Untuk informasi tambahan tentang menghubungkan ke AWS dari AWS Toolkit menggunakan kredensial yang ada, lihat AWS topik [Menyambung ke](#) dalam Panduan Pengguna ini.

# AWS Komposer Infrastruktur

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

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

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

Topik

- [Bekerja dengan AWS Infrastructure Composer di Toolkit](#)

## Bekerja dengan AWS Infrastructure Composer di Toolkit

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

Untuk informasi rinci tentang AWS Infrastructure Composer, memulai informasi, dan tutorial, lihat Panduan Pengguna [AWS Infrastructure Composer](#).

Bagian berikut menjelaskan cara mengakses AWS Infrastructure Composer dari. AWS Toolkit for Visual Studio Code

## Mengakses AWS Infrastructure Composer dari Toolkit

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

Mengakses AWS Infrastructure 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 Infrastructure Composer yang terletak di sudut kanan atas jendela editor.
3. AWS Infrastructure Composer membuka dan memvisualisasikan file template Anda di jendela editor VS Code.

Mengakses AWS Infrastructure Composer dari menu konteks (klik kanan)

1. Dari VS Code klik kanan file template yang ingin Anda buka dengan AWS Infrastructure Composer.
2. Di menu konteks, pilih opsi Buka dengan App Composer.
3. AWS Infrastructure Composer membuka dan memvisualisasikan file template Anda di jendela editor VS Code baru.

Mengakses AWS Infrastructure 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 Infrastructure Composer** dan pilih AWS Infrastructure Composer saat mengisi hasil.
3. Pilih file template yang ingin Anda buka, AWS Infrastructure 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. Dokumentasi dapat berubah.

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

AWS CDK aplikasi terdiri dari blok bangunan yang dikenal sebagai [konstruksi](#), yang mencakup definisi untuk CloudFormation tumpukan Anda dan AWS sumber daya di dalamnya. Menggunakan AWS CDK Explorer, Anda dapat memvisualisasikan [tumpukan](#) dan [sumber daya](#) yang didefinisikan

dalam AWS CDK konstruksi. Visualisasi ini disediakan dalam tampilan pohon di panel Alat Pengembang dalam editor Visual Studio Code (VS Code).

Bagian ini memberikan informasi tentang cara mengakses dan menggunakan AWS CDK editor VS Code. Ini mengasumsikan bahwa Anda telah [menginstal dan mengonfigurasi](#) 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 dalam beberapa bagian pertama [Memulai dengan AWS CDK](#) di Panduan AWS Cloud Development Kit (AWS CDK) Pengembang.

#### 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 aplikasi CDK.

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

 Important

Ketika Anda tiba di langkah tutorial Deploying the 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 yang diperlukan dari tutorial CDK, buka konten CDK yang Anda buat di editor VS Code.
3. Dari panel AWS navigasi, perluas judul CDK (Pratinjau). Aplikasi CDK Anda dan sumber daya terkaitnya sekarang ditampilkan di tampilan pohon CDK Explorer.

### Catatan penting

- Saat memuat aplikasi CDK ke editor VS Code, Anda dapat memuat beberapa folder sekaligus. Setiap folder dapat berisi beberapa aplikasi CDK, 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 aplikasi CDK, 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 VS Code untuk melakukan operasi lain pada aplikasi CDK, 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

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

### Topik

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

## Menghapus tumpukan CloudFormation

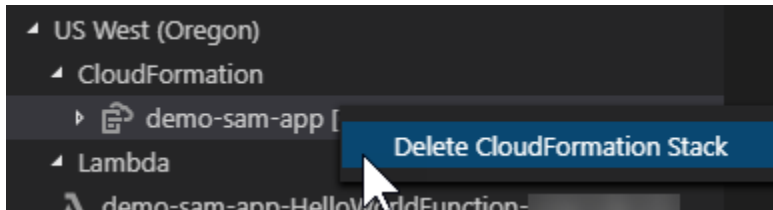
Anda dapat menggunakan AWS Toolkit for Visual Studio Code untuk menghapus CloudFormation tumpukan.

### Prasyarat

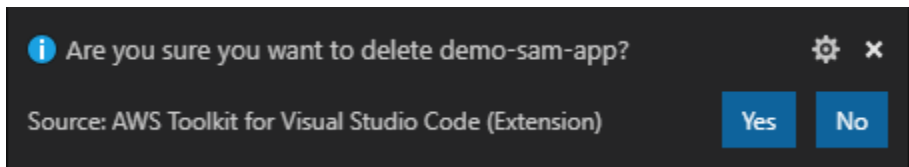
- Pastikan sistem Anda memenuhi prasyarat yang ditentukan dalam [Menginstal Toolkit for VS Code](#).
- Pastikan bahwa kredensial yang Anda konfigurasi [Otentikasi dan akses](#) menyertakan akses baca/tulis yang sesuai ke layanan. CloudFormation Jika di AWS Explorer, di bawah CloudFormation, Anda melihat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

## Hapus CloudFormation tumpukan

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



2. Pilih Hapus CloudFormation Tumpukan.
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 read/write akses 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, 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 CloudWatch Log dengan menggunakan AWS Toolkit for Visual Studio Code

Amazon CloudWatch Logs memungkinkan Anda 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 selengkapnya, lihat [Apa itu Amazon CloudWatch Logs?](#) di Panduan CloudWatch Pengguna Amazon.

Topik berikut menjelaskan cara menggunakan AWS Toolkit for Visual Studio Code untuk bekerja dengan CloudWatch Log di AWS akun.

### Topik

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

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

Pengaliran log adalah urutan log acara yang berbagi sumber yang sama. Setiap sumber log yang terpisah ke dalam CloudWatch Log membentuk aliran 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 log stream untuk dimasukkan ke dalam setiap grup. Tidak ada batas jumlah log stream yang dapat bergabung dalam satu grup log.

Untuk informasi selengkapnya, lihat [Bekerja dengan Grup Log dan Aliran Log](#) di Panduan CloudWatch Pengguna Amazon.

## Topik

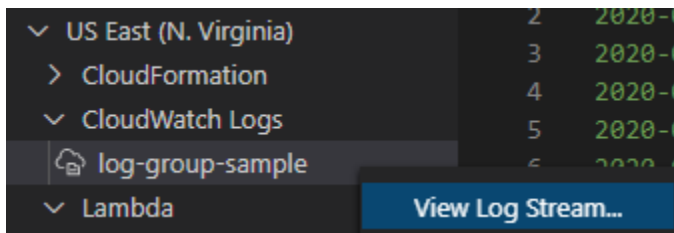
- [Melihat grup log dan aliran log dengan node CloudWatch Log](#)

## Melihat grup log dan aliran log dengan node CloudWatch Log

1. Di VS Code, pilih View, Explorer untuk membuka AWS Explorer.
2. Klik node CloudWatch Log untuk memperluas daftar grup log.

Grup log untuk AWS Wilayah saat ini ditampilkan di bawah node CloudWatch Log.

3. Untuk melihat aliran log dalam grup log, klik kanan nama grup log, lalu pilih Lihat Aliran Log.



4. Dari Command Palette, pilih aliran log dari grup yang akan dilihat.

### Note

Command Palette menampilkan stempel waktu untuk acara terakhir di setiap aliran.

[Editor Log Stream](#) diluncurkan untuk menampilkan peristiwa log aliran.

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

Setelah Anda membuka editor Log Steam, 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, editor Log Stream menampilkan urutan peristiwa log streaming tersebut.

1. Untuk menemukan aliran log untuk dilihat, buka editor Aliran Log (lihat [Melihat grup CloudWatch log 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 peristiwa berdasarkan waktu: Tampilkan peristiwa log terbaru dan yang lebih lama dengan memilih Muat peristiwa yang lebih baru atau Muat peristiwa 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 editor Aliran Log dan pilih Salin Nama Aliran Log.

### Note

Anda juga dapat menggunakan Command Palette untuk menjalankan AWS Toolkit Copy Log Stream Name.

## Menyimpan konten dari editor aliran log ke file lokal

Anda dapat mengunduh konten editor aliran CloudWatch log ke log file di mesin lokal Anda.

**Note**

Dengan opsi ini, Anda menyimpan untuk mengajukan 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 untuk disalin, buka editor Aliran Log (lihat [Melihat grup CloudWatch log dan aliran log](#)).
2. Pilih ikon Simpan di samping tab yang menampilkan nama aliran log.

**Note**

Anda juga dapat menggunakan Command Palette untuk menjalankan AWS Toolkit Save Current Log Stream Content.

3. 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 Log, lihat bagian [Filter dan sintaks pola](#) di Panduan CloudWatch Pengguna Amazon.

1. Dari VS Code, buka Command Palette dengan menekan **cmd+shift+p** (windows:**ctrl+shift+p**).

2. 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.

3. Dari prompt Pilih Grup Log (1/3), pilih grup log yang ingin Anda cari.
4. Dari prompt Select Time Filter (2/3), pilih filter waktu yang akan diterapkan pada pencarian Anda.
5. 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.

## Amazon CloudWatch Log Ekor Langsung

Streaming langsung peristiwa CloudWatch log Anda saat dimasukkan ke dalam Grup Log tertentu dengan Amazon CloudWatch Logs Live Tail.

Untuk informasi lebih lanjut tentang fitur Live Tail, lihat topik [Pemecahan Masalah dengan CloudWatch Log Live Tail](#) di Panduan Pengguna Amazon CloudWatch Logs.

Sesi Live Tail dikenakan biaya berdasarkan waktu penggunaan sesi, per menit. Untuk informasi tentang harga, lihat tab Log di bagian Tingkat Berbayar dari panduan [CloudWatch Harga Amazon](#).

### Memulai sesi Live Tail dari VS Code Command Palette

Untuk memulai sesi Live Tail dari VS Code Command Palette, selesaikan langkah-langkah berikut.

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

Memulai sesi tailing dari Command Palette

1. Dari VS Code, buka Command Palette dengan menekan **cmd+shift+p** (Windows:**ctrl+shift+p**).
2. Dari Command Palette, masukkan perintah **AWS: Tail Log Group**, lalu pilih untuk membuka dialog Tail log group di VS Code dan ikuti petunjuk untuk melanjutkan.

#### Note

Pada prompt pertama Anda memiliki opsi untuk beralih AWS Wilayah Anda sebelum melanjutkan ke langkah berikutnya.

3. Dari prompt Tail Log Group (1/3), pilih grup log yang ingin Anda ekor.
4. Dari prompt Sertakan peristiwa log dari... (2/3), pilih filter aliran log untuk diterapkan ke sesi tailing Anda.
5. Dari prompt Berikan pola filter peristiwa log... (3/3), masukkan pola filter Anda di bidang yang disediakan, lalu tekan **Enter** tombol untuk melanjutkan atau **ESC** tombol untuk membatalkan pencarian.
6. Setelah selesai, hasil Anda mengalir ke editor VS Code

**Note**

Jika sesi Live Tail yang berjalan di jendela VS Code cocok dengan konfigurasi perintah Tail Log Group yang baru dikirimkan, sesi baru tidak akan dimulai. Sebaliknya, sesi Anda yang ada menjadi editor teks aktif.

## Memulai sesi Live Tail dari AWS Explorer

Untuk memulai sesi Live Tail dari AWS Toolkit Explorer, selesaikan langkah-langkah berikut.

### Memulai sesi tailing dari Explorer AWS

1. Dari AWS Toolkit Explorer, perluas CloudWatch.
2. Buka menu konteks untuk (klik kanan) Grup Log yang ingin Anda ekor, lalu pilih Grup Log Ekor untuk membuka prompt tailing.
3. Ikuti petunjuk untuk melanjutkan.
4. Hasil Anda akan mengalir ke editor VS Code.

## Menghentikan sesi Live Tail

Ada 2 cara untuk menghentikan sesi Tailing yang sedang berjalan.

### Menghentikan sesi tailing

1. Klik Stop tailing CodeLens di bagian bawah dokumen teks sesi tailing.
2. Tutup semua editor yang berisi dokumen teks sesi tailing.

## Amazon DocumentDB

Anda dapat mengelola cluster dan instance Amazon DocumentDB langsung di VS Code dengan file. AWS Toolkit for Visual Studio Code Amazon DocumentDB (dengan kompatibilitas MongoDB) adalah layanan database yang cepat, andal, dan dikelola sepenuhnya yang menyederhanakan pengaturan, pengoperasian, dan penskalaan database yang kompatibel dengan MongoDB di cloud. Untuk informasi rinci tentang layanan Amazon DocumentDB, lihat [Panduan Pengembang Amazon DocumentDB](#).

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

Topik

- [Bekerja dengan Amazon DocumentDB di Toolkit](#)

## Bekerja dengan Amazon DocumentDB di Toolkit

Amazon DocumentDB (dengan kompatibilitas MongoDB) adalah layanan database yang cepat, andal, dan dikelola sepenuhnya yang menyederhanakan pengaturan, pengoperasian, dan penskalaan database yang kompatibel dengan MongoDB di cloud.

Untuk informasi terperinci tentang Amazon DocumentDB, memulai informasi, dan tutorial, lihat Panduan Pengembang Amazon [DocumentDB](#).

Bagian berikut menjelaskan cara bekerja dengan Amazon DocumentDB dengan file. AWS Toolkit for Visual Studio Code

### Mengakses Amazon AWS DocumentDB dari Toolkit

Untuk mengakses Amazon DocumentDB dengan Toolkit, AWS selesaikan prosedur berikut.

#### Mengakses Amazon DocumentDB di Toolkit AWS

1. Dari VS Code, buka AWS Toolkit for Visual Studio Code.
2. Dari AWS Toolkit, perluas Explorer.
3. Dari Explorer, perluas Amazon DocumentDB untuk menampilkan sumber daya Amazon DocumentDB yang ada.


### Membuat cluster berbasis instance.

Untuk mulai bekerja dengan Amazon DocumentDB, buat cluster dengan menyelesaikan prosedur berikut.

#### Membuat cluster berbasis instance

1. Dari AWS Toolkit for Visual Studio Code, buka menu konteks untuk (klik kanan) Amazon DocumentDB, lalu pilih Create Cluster untuk membuka dialog Create Amazon DocumentDB Cluster di VS Code.

2. Dari layar tipe Cluster, pilih Instance Based Cluster.
3. Dari nama Cluster layar, tentukan nama untuk cluster baru Anda.
4. Dari layar Select engine version, pilih versi mesin Amazon DocumentDB pilihan Anda.
5. Dari layar nama pengguna dan kata sandi Admin, tentukan nama pengguna dan kata sandi admin untuk melindungi klaster Anda.
6. Dari layar Tentukan enkripsi Penyimpanan, pilih apakah akan mengenkripsi klaster Anda atau tidak.
7. Dari layar Jumlah instans, konfigurasi jumlah instans pilihan Anda.
8. Dari layar Select instance class, pilih kelas instans pilihan Anda lalu lanjutkan untuk membuat cluster baru Anda.

 Note

Ini bisa memakan waktu beberapa menit untuk spin-up cluster Anda.

## Menyalin titik akhir cluster

Untuk menyalin titik akhir klaster Amazon DocumentDB Anda, selesaikan prosedur berikut.

### Menyalin titik akhir cluster

1. Dari AWS Toolkit for Visual Studio Code, perluas Amazon DocumentDB untuk menampilkan cluster Amazon DocumentDB Anda.
2. Klik kanan klaster tempat Anda ingin menyalin detail koneksi, lalu pilih Salin Titik Akhir untuk menyalin informasi titik akhir cluster ke clipboard Anda.
3. Endpoint cluster Anda sekarang dapat disisipkan ke dalam dokumen Anda.

## Buka di browser

Buka kluster Amazon DocumentDB Anda di Konsol untuk fitur AWS pengelolaan klaster lainnya. Untuk membuka AWS Konsol ke cluster Amazon DocumentDB Anda di browser web default Anda, selesaikan prosedur berikut.

## Membuka klaster Anda di AWS Konsol

1. Dari AWS Toolkit for Visual Studio Code, perluas Amazon DocumentDB untuk menampilkan cluster Amazon DocumentDB Anda.
2. Klik kanan klaster yang ingin Anda lihat di AWS Konsol, lalu pilih Buka di Browser.
3. AWS Konsol terbuka ke cluster Amazon DocumentDB di browser web default Anda.

## Memperluas cluster yang ada

Untuk menskalakan klaster Amazon DocumentDB Anda dengan menambahkan instance, selesaikan prosedur berikut.

### Menambahkan instance untuk memperluas klaster Anda

1. Dari AWS Toolkit for Visual Studio Code, perluas Amazon DocumentDB untuk menampilkan cluster Amazon DocumentDB Anda.
2. Klik kanan cluster yang ingin Anda kembangkan dan pilih Tambahkan Instance untuk membuka dialog Add an Instance di VS Code.
3. Saat diminta, masukkan nama untuk instance baru Anda ke dalam bidang teks, lalu tekan **Enter** tombol untuk melanjutkan.
4. Saat diminta, pilih kelas instance dari daftar untuk melanjutkan.
5. AWS Explorer menampilkan status pembuatan dan pembaruan saat instance baru siap.

## Menghentikan klaster

Hentikan klaster Amazon DocumentDB Anda dengan menyelesaikan prosedur berikut.

### Note

Saat klaster Anda dihentikan, sebagian besar fitur manajemen klaster tidak akan tersedia.

## Menghentikan klaster Amazon DocumentDB

1. Dari AWS Toolkit for Visual Studio Code, perluas Amazon DocumentDB untuk menampilkan cluster Amazon DocumentDB Anda.

2. Pilih tombol Stop Cluster yang terletak di sebelah cluster yang ingin Anda hentikan atau klik kanan cluster dan pilih Stop Cluster.
3. Saat diminta, pilih Ya untuk menghentikan klaster Anda atau Batal untuk membatalkan proses penghentian dan membiarkan klaster Anda berjalan.
4. AWS Explorer menampilkan status klaster Anda dan pembaruan saat klaster berhenti.

## Mem-boot ulang sebuah instance

Mem-boot ulang instance berguna untuk pemecahan masalah dan membuat perubahan kecil tanpa memengaruhi seluruh klaster Anda. Untuk me-reboot instans Amazon DocumentDB, selesaikan prosedur berikut.

### Melakukan reboot instans klaster

1. Dari AWS Toolkit for Visual Studio Code, perluas Amazon DocumentDB untuk menampilkan cluster Amazon DocumentDB Anda.
2. Klik kanan instance cluster yang ingin Anda reboot, lalu pilih Reboot Instance.
3. Saat diminta, pilih Ya untuk me-reboot instance Anda atau Batal untuk membatalkan proses reboot dan biarkan instance Anda berhenti.
4. AWS Explorer menampilkan status klaster dan pembaruan saat instans Anda telah di-boot ulang.

## Menghapus sebuah instans

Untuk menghapus instance klaster Amazon DocumentDB, selesaikan prosedur berikut.

### Note

Menghapus instance tidak memengaruhi data di klaster Anda. Jika Anda menghapus instance utama, salah satu instance replika mengambil alih sebagai instance yang dapat ditulis.

### Menghapus instance cluster

1. Dari AWS Toolkit for Visual Studio Code, perluas Amazon DocumentDB untuk menampilkan cluster Amazon DocumentDB Anda.

2. Klik kanan instance cluster yang ingin Anda hapus, lalu pilih Hapus untuk membuka dialog delete-cluster-instance konfirmasi di VS Code.
3. Ikuti prompt konfirmasi, lalu tekan **Enter** tombol untuk menghapus instance cluster Anda.
4. AWS Explorer menampilkan status instans cluster Anda dan update saat instans Anda telah dihapus.

## Melihat, menambah, dan menghapus tag

Tag digunakan untuk mengatur dan melacak sumber daya dalam lingkungan Anda. Untuk melihat atau mengedit tag yang terkait dengan klaster Amazon DocumentDB Anda, lengkapi salah satu prosedur berikut.

### Melihat tag cluster

1. Dari AWS Toolkit for Visual Studio Code, perluas Amazon DocumentDB untuk menampilkan cluster Amazon DocumentDB Anda.
2. Klik kanan cluster yang ingin Anda lihat tag, lalu pilih Tag... untuk membuka Tag untuk **your cluster name** dialog.
3. Tag Anda ditampilkan di jendela dialog, jika tidak ada tag yang terkait dengan klaster Anda maka pesan [Tidak ada tag yang ditetapkan] ditampilkan.

### Menambahkan tag ke klaster Anda

1. Dari AWS Toolkit for Visual Studio Code, perluas Amazon DocumentDB untuk menampilkan cluster Amazon DocumentDB Anda.
2. Klik kanan cluster yang ingin Anda tambahkan tag, lalu pilih Tag... untuk membuka Tag untuk **your cluster name** dialog.
3. Pilih Add tag... tombol untuk membuka dialog Add Tag di VS Code.
4. Masukkan tag baru ke dalam bidang teks, lalu tekan tombol Enter untuk melanjutkan.
5. Masukkan nilai ke dalam bidang teks, lalu tekan Enter untuk menambahkan key/value pasangan ke cluster Anda.

## Menghapus tag dari kluster

1. Dari AWS Toolkit for Visual Studio Code, perluas Amazon DocumentDB untuk menampilkan cluster Amazon DocumentDB Anda.
2. Klik kanan cluster tempat Anda ingin menghapus tag, lalu pilih Tag... untuk membuka Tag untuk **your cluster name** dialog.
3. Pilih tag Hapus... tombol untuk membuka Hapus tag dari **your cluster name** dialog di VS Code.
4. Pilih tag yang ingin Anda hapus dari daftar yang disediakan untuk menghapus tag dari kluster Anda.

## Memodifikasi kelas instance

Untuk memodifikasi kelas instance kluster Amazon DocumentDB, selesaikan prosedur berikut.

### Memodifikasi kelas instance

1. Dari AWS Toolkit for Visual Studio Code, perluas Amazon DocumentDB untuk menampilkan cluster Amazon DocumentDB Anda.
2. Klik kanan instance cluster yang ingin Anda modifikasi, lalu pilih Modify Class... untuk membuka dialog Select instance class di VS Code.
3. Pilih kelas baru untuk instance Anda dari daftar untuk memperbarui kelas.
4. AWS Explorer menampilkan status instans cluster Anda dan update saat kelas instans Anda telah diperbarui.

## Amazon Elastic Compute Cloud

Amazon Elastic Compute Cloud for the AWS Toolkit for Visual Studio Code memungkinkan Anda meluncurkan dan terhubung ke EC2 instans Amazon dari VS Code. Untuk informasi rinci tentang Amazon EC2, lihat [Apa itu Amazon EC2?](#) topik di Panduan Pengguna Amazon Elastic Compute Cloud.

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

### Topik

- [Bekerja dengan Amazon Elastic Compute Cloud](#)
- [Memecahkan Masalah Amazon Elastic Compute Cloud](#)

## Bekerja dengan Amazon Elastic Compute Cloud

Bagian berikut menjelaskan cara bekerja dengan Amazon Elastic Compute Cloud di. AWS Toolkit for Visual Studio Code

### Prasyarat

Fitur yang dijelaskan dalam topik panduan pengguna ini telah diuji pada EC2 instans Amazon dengan sistem operasi berikut:

- Windows 2016+

#### Note

OS ini hanya berfungsi saat menghubungkan terminal VS Code. Itu tidak berfungsi saat menghubungkan instance jarak jauh VS Code lengkap. Untuk informasi tambahan tentang terminal VS Code dan instance jarak jauh, lihat topik [Memulai dengan terminal](#) dan [VS Code Remote Development](#) dalam dokumentasi VS Code.

- Amazon Linux 2023
- Ubuntu, 22.04

SSH yang diinstal secara lokal diperlukan untuk membuka koneksi jarak jauh ke EC2 instans Amazon, tetapi tidak diperlukan untuk membuka terminal ke instans Amazon EC2 .

Profil EC2 instans Amazon Anda harus menyertakan izin AWS Identity and Access Management (IAM) berikut.

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

**Note**

Izin yang diperlukan disertakan dalam kebijakan AWS terkelola berikut.

- AmazonSSMManagedInstanceCore
- AmazonSSMManagedEC2InstanceDefaultPolicy

## Melihat EC2 instans Amazon yang ada

Untuk melihat EC2 instans Amazon yang ada dari AWS Toolkit, selesaikan langkah-langkah berikut.

1. Dari AWS Toolkit, perluas AWS Toolkit Explorer.
2. Perluas wilayah yang berisi EC2 instans Amazon yang ingin Anda lihat.
3. Perluas EC2 judul untuk menampilkan EC2 instans Amazon yang ada.

## Meluncurkan EC2 instans Amazon baru

Ada 3 cara untuk membuat EC2 instance Amazon baru dengan AWS Toolkit.

Setiap alur kerja membuka wizard Launch an instance di AWS konsol. Untuk informasi mendetail tentang meluncurkan EC2 instans Amazon baru dari panduan Luncurkan instans, lihat [Peluncuran EC2 instance menggunakan panduan instance peluncuran dalam topik konsol di](#) Panduan Pengguna Amazon Elastic Compute Cloud. Untuk meluncurkan EC2 instans Amazon baru, selesaikan salah satu prosedur berikut.

Meluncurkan EC2 instans Amazon baru dari VS Code Command Palette

1. Dari VS Code, buka VS Code Command Palette dengan menekan **command + shift + P (Windows: ctrl + shift + P)**
2. Dari Palet Perintah Kode VS, cari **AWS: Launch EC2** perintah dan pilih saat terisi dalam daftar untuk membuka prompt Launch EC2 instance Select Region di VS Code.
3. Dari prompt Launch EC2 instance Select Region, pilih wilayah tempat Anda ingin meluncurkan instans baru, lalu konfirmasi bahwa Anda ingin membuka AWS Konsol di browser web default Anda.

4. Dari AWS Konsol di browser web default Anda, selesaikan proses otentikasi untuk melanjutkan ke panduan Luncurkan instans.
5. Dari panduan Luncurkan instans, lengkapi bagian yang diperlukan, lalu pilih tombol Launch instance untuk meluncurkan EC2 instans Amazon baru Anda.
6. AWS Explorer memperbarui untuk menampilkan EC2 instans Amazon baru Anda.

#### Meluncurkan EC2 instans Amazon baru dari AWS Explorer

1. Perluas AWS Toolkit Explorer, lalu perluas wilayah tempat Anda ingin membuat EC2 instance Amazon baru.
2. Perluas atau arahkan kursor ke EC2 judul, lalu pilih ikon + (Launch EC2 instance).
3. Saat diminta, konfirmasikan bahwa Anda ingin membuka AWS Konsol di browser web default Anda.
4. Dari AWS Konsol di browser web Anda, selesaikan proses otentikasi untuk melanjutkan ke panduan Luncurkan instans.
5. Dari panduan Luncurkan instans, lengkapi bagian yang diperlukan, lalu pilih tombol Launch instance untuk meluncurkan EC2 instans Amazon baru Anda.
6. AWS Explorer memperbarui untuk menampilkan EC2 instans Amazon baru Anda.

#### Meluncurkan EC2 instance Amazon baru dari menu konteks (klik kanan)

1. Perluas AWS Toolkit Explorer, lalu perluas wilayah tempat Anda ingin membuat EC2 instance Amazon baru.
2. Klik kanan EC2 judul, lalu pilih Launch EC2 instance.
3. Saat diminta, konfirmasikan bahwa Anda ingin membuka AWS Konsol di browser web default Anda.
4. Dari AWS Konsol di browser web Anda, selesaikan proses otentikasi untuk melanjutkan ke panduan Luncurkan instans.
5. Dari panduan Luncurkan instans, lengkapi bagian yang diperlukan, lalu pilih tombol Launch instance untuk meluncurkan EC2 instans Amazon baru Anda.
6. AWS Explorer memperbarui untuk menampilkan EC2 instans Amazon baru Anda.

## Menghubungkan Kode VS ke EC2 instans Amazon

Ada 3 cara untuk terhubung ke EC2 instance Amazon dari VS Code. Untuk menghubungkan VS Code ke EC2 instans Anda, selesaikan salah satu prosedur berikut.

### Menghubungkan VS Code ke EC2 instans Amazon dari Command Palette

1. Dari VS Code, buka VS Code Command Palette dengan menekan **command + shift + P** (**Windows: ctrl + shift + P**)
2. Dari Palet Perintah Kode VS, cari **AWS: Connect VS Code to EC2 instance...** perintah dan pilih saat terisi dalam daftar untuk membuka prompt Select EC2 Instance di VS Code.
3. Dari prompt Select EC2 Instance, pilih wilayah yang berisi instance yang ingin Anda sambungkan, lalu pilih instance yang ingin Anda sambungkan.
4. VS Code menampilkan status saat koneksi sedang dibuat.
5. Jendela baru terbuka untuk menampilkan EC2 instance Amazon Anda saat koneksi selesai.

### Menghubungkan VS Code ke EC2 instans Amazon dari AWS Explorer.

1. Perluas AWS Toolkit Explorer, lalu perluas wilayah yang berisi EC2 instance Amazon yang ingin Anda sambungkan.
2. Arahkan kursor ke EC2 instance Amazon, lalu pilih ikon (Connect VS Code to EC2 instance).


#### Note

Anda juga dapat memilih ikon (Connect VS Code to EC2 instance) dari judul EC2 layanan di AWS Explorer.

3. VS Code menampilkan status saat koneksi sedang dibuat.
4. Jendela baru terbuka untuk menampilkan EC2 instance Amazon Anda saat koneksi selesai.

### Menghubungkan VS Code ke EC2 instans Amazon dari menu klik kanan

1. Perluas AWS Toolkit Explorer, lalu perluas wilayah yang berisi EC2 instance Amazon yang ingin Anda sambungkan.
2. Klik kanan EC2 instans Amazon yang ingin Anda sambungkan, lalu pilih Connect VS Code to EC2 instance.

 Note

Anda juga dapat mengklik kanan judul EC2 layanan di AWS Explorer dan memilih Connect VS Code to EC2 instance.

3. VS Code menampilkan status saat koneksi sedang dibuat.
4. Jendela baru terbuka untuk menampilkan EC2 instance Amazon Anda saat koneksi selesai.

## Membuka terminal ke EC2 instans Amazon.


Ada 3 cara untuk terhubung ke EC2 instance Amazon dari terminal VS Code.

Menghubungkan VS Code ke EC2 instans Amazon dari Command Palette

1. Dari VS Code, buka VS Code Command Palette dengan menekan **command + shift + P** (**Windows: ctrl + shift + P**)
2. Dari Palet Perintah Kode VS, cari **AWS:Open terminal to EC2 instance...** perintah dan pilih saat terisi dalam daftar untuk membuka prompt Select EC2 Instance di VS Code.
3. Dari prompt Select EC2 Instance, pilih wilayah yang berisi instance yang ingin Anda buka di terminal, lalu pilih instance.
4. VS Code menampilkan status saat koneksi sedang dibuat.
5. Terminal VS Code terbuka untuk menampilkan sesi baru Anda saat koneksi selesai.

Membuka EC2 instance Amazon di terminal VS Code dari AWS Explorer.

1. Perluas AWS Toolkit Explorer, lalu perluas wilayah yang berisi EC2 instance Amazon yang ingin Anda sambungkan.
2. Arahkan kursor ke EC2 instance Amazon, lalu pilih (Buka terminal ke EC2 instance...) ikon.

 Note

Anda juga dapat memilih (Buka terminal ke EC2 instance...) ikon dari judul EC2 layanan di AWS Explorer.

3. VS Code menampilkan status saat koneksi sedang dibuat.
4. Terminal VS Code terbuka untuk menampilkan sesi baru Anda saat koneksi selesai.

## Membuka EC2 instance Amazon di terminal VS Code dari menu klik kanan

1. Perluas AWS Toolkit Explorer, lalu perluas wilayah yang berisi EC2 instance Amazon yang ingin Anda buka di terminal VS Code.
2. Klik kanan EC2 instance Amazon yang ingin Anda buka di terminal, lalu pilih Open terminal to EC2 instance... .

### Note

Anda juga dapat mengklik kanan judul EC2 layanan di AWS Explorer dan memilih terminal Open untuk EC2... .

3. VS Code menampilkan status saat koneksi sedang dibuat.
4. Terminal VS Code terbuka untuk menampilkan sesi baru Anda saat koneksi selesai.

## Memulai atau me-reboot instans Amazon EC2

Ada 3 cara untuk memulai atau me-reboot EC2 instance Amazon.

### Mem-boot ulang EC2 instance Amazon dari Command Palette

1. Dari VS Code, buka VS Code Command Palette dengan menekan **command + shift + P** (**Windows: ctrl + shift + P**)
2. Dari Palet Perintah Kode VS, cari **AWS: Reboot EC2 instance** perintah dan pilih saat terisi dalam daftar untuk membuka prompt Select EC2 Instance di VS Code.

### Note

Untuk memulai instance yang tidak berjalan, Anda harus memilih **AWS: Start EC2 instance** perintah. **AWS: Reboot EC2 instance** Perintah hanya me-reboot instance yang sedang berjalan.

3. Dari prompt Select EC2 Instance, pilih wilayah yang berisi instance yang ingin Anda mulai atau reboot.
4. VS Code menampilkan status saat instance reboot.
5. AWS Explorer memperbarui untuk menunjukkan bahwa instans Anda berjalan setelah selesai reboot.

## Memulai atau me-reboot EC2 instans Amazon dari Explorer AWS

1. Perluas AWS Toolkit Explorer, lalu perluas wilayah yang berisi EC2 instance Amazon yang ingin Anda mulai atau reboot.
2. Arahkan kursor ke EC2 instance Amazon, lalu pilih ikon (Reboot EC2 instance).

### Note

Jika instance dihentikan, maka satu-satunya opsi adalah ikon (Mulai EC2 instance)

3. VS Code menampilkan status saat instance reboot.
4. AWS Explorer memperbarui untuk menunjukkan bahwa instans Anda berjalan setelah selesai reboot.

## Memulai atau me-reboot EC2 instance Amazon dari menu klik kanan

1. Perluas AWS Toolkit Explorer, lalu perluas wilayah yang berisi EC2 instance Amazon yang ingin Anda mulai atau reboot.
2. Klik kanan EC2 instans Amazon yang ingin Anda sambungkan, lalu pilih Reboot EC2 instance.

### Note

Jika instance dihentikan, maka satu-satunya pilihan adalah EC2 instance Start.

3. VS Code menampilkan status saat instance reboot.
4. AWS Explorer memperbarui untuk menunjukkan bahwa instans Anda berjalan setelah selesai reboot.

## Menghentikan EC2 instans Amazon

Ada 3 cara untuk menghentikan EC2 instance Amazon.

### Menghentikan EC2 instans Amazon dari Command Palette

1. Dari VS Code, buka VS Code Command Palette dengan menekan **command + shift + P** (**Windows: ctrl + shift + P**)

2. Dari Palet Perintah Kode VS, cari **AWS: Stop EC2 instance** perintah dan pilih saat terisi dalam daftar untuk membuka prompt Select EC2 Instance di VS Code.
3. Dari prompt Select EC2 Instance, pilih wilayah yang berisi instance yang ingin Anda hentikan.
4. VS Code menampilkan status saat instance berhenti.
5. AWS Explorer memperbarui untuk menunjukkan bahwa instans Anda dihentikan.

### Menghentikan EC2 instans Amazon dari AWS Explorer

1. Perluas AWS Toolkit Explorer, lalu perluas wilayah yang berisi EC2 instance Amazon yang ingin Anda hentikan.
2. Arahkan kursor ke EC2 instance Amazon, lalu pilih ikon (Stop EC2 instance).
3. VS Code menampilkan status saat instance berhenti.
4. AWS Explorer memperbarui untuk menunjukkan bahwa instans Anda telah berhenti.

### Menghentikan EC2 instans Amazon dari menu klik kanan

1. Perluas AWS Toolkit Explorer, lalu perluas wilayah yang berisi EC2 instance Amazon yang ingin Anda hentikan.
2. Klik kanan EC2 instans Amazon yang ingin Anda sambungkan, lalu pilih Reboot EC2 instance.
3. VS Code menampilkan status saat instance berhenti.
4. AWS Explorer memperbarui untuk menunjukkan bahwa instans Anda telah berhenti.

### Salin ID Instance

Untuk menyalin ID instance, selesaikan langkah-langkah berikut.

1. Klik kanan instance yang ingin Anda salin ID dari.
2. Pilih Salin ID Instance.
3. ID instance disalin ke clipboard lokal Anda.

### Salin Nama

Untuk menyalin nama instance, selesaikan langkah-langkah berikut.

1. Klik kanan instance yang ingin Anda salin namanya.

2. Pilih Salin Nama Instance.
3. Nama instance disalin ke clipboard lokal Anda.

## Salin ARN

Untuk menyalin ARN instance, selesaikan langkah-langkah berikut.

1. Klik kanan instance yang ingin Anda salin ARN.
2. Pilih Salin Instance ARN.
3. ARN instance disalin ke clipboard lokal Anda.

## Memecahkan Masalah Amazon Elastic Compute Cloud

Bagian berikut menjelaskan cara memecahkan masalah yang diketahui yang dapat terjadi saat bekerja dengan Amazon Elastic Compute Cloud di AWS Toolkit for Visual Studio Code. Untuk informasi mendetail tentang masalah pemecahan masalah khusus untuk layanan Amazon EC2, lihat topik [Memecahkan masalah dengan instans Amazon EC2 di Panduan Pengguna Amazon Elastic Compute Cloud](#).

### Debugging Umum

Jika Anda mengalami masalah koneksi jarak jauh karena alasan apa pun, mulailah dengan memeriksa untuk melihat apakah AWS Systems Manager koneksi dapat dibuat dari AWS Konsol.

Untuk menyambung ke instans Amazon EC2 melalui Systems Manager dari AWS Konsol, selesaikan langkah-langkah berikut.

1. Dari browser web Anda, navigasikan ke [AWS Konsol](#).
2. Otentikasi lengkap untuk melanjutkan ke pendaratan AWS Console EC2.
3. Dari panel navigasi Amazon EC2, pilih Instans.
4. Pilih kotak yang terletak di sebelah instance yang ingin Anda sambungkan.
5. Pilih tombol Connect untuk membuka layar Connect to instance di tab browser baru.

**Note**

Anda hanya dapat terhubung ke sebuah instance jika sedang berjalan. Jika Anda tidak dapat memilih tombol Connect, periksa untuk memastikan instans Anda berjalan.

6. Dari layar Connect to instance, pilih tab Session Manager, lalu pilih tombol Connect untuk membuka koneksi Systems Manager di tab browser Anda saat ini.

**Note**

Jika Anda baru saja memulai instans dan opsi tidak tersedia bagi Anda untuk menghubungkan Systems Manager, Anda mungkin perlu menunggu beberapa menit lagi sebelum opsi tersedia.

## Instance target tidak berjalan

Untuk menyambung ke instans Amazon EC2 dari terminal atau koneksi jarak jauh, instans harus berjalan. Sebelum Anda mencoba untuk terhubung ke instans Anda dari AWS Toolkit, mulai dari AWS Explorer, Konsol Manajemen AWS, atau AWS Command Line Interface.

## Instance target tidak memiliki peran IAM atau memiliki peran IAM dengan izin yang tidak tepat

Untuk terhubung ke instans Amazon EC2 Anda, instans harus memiliki peran IAM dengan izin yang benar dilampirkan. Jika Anda mencoba terhubung ke instance yang tidak memiliki peran IAM yang dilampirkan, Anda akan diberi tahu oleh VS Code.

Jika Anda mencoba menyambung ke instans yang memiliki peran IAM tetapi tidak memiliki izin yang diperlukan, Anda diminta untuk menambahkan tindakan minimum yang diperlukan sebagai kebijakan sebaris ke peran IAM yang ada. Setelah memperbarui kebijakan inline, Anda terhubung ke instans Anda. Untuk informasi mendetail tentang peran IAM, izin, dan melampirkan peran ke instans, lihat topik peran [IAM untuk Amazon EC2 di Panduan Pengguna Amazon Elastic Compute Cloud](#) dan [Langkah 2: Verifikasi atau tambahkan izin instans untuk topik Session Manager di Panduan Pengguna Systems Manager](#).AWS

Contoh berikut berisi tindakan minimum yang diperlukan.

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

### Note

Izin yang diperlukan disertakan dalam kebijakan AWS terkelola berikut.

- AmazonSSMManagedEC2InstanceDefaultPolicy
- AmazonSSMManagedInstanceCore

## Instance target tidak memiliki agen Systems Manager yang berjalan

Anda mungkin mengalami masalah ini karena sejumlah alasan berbeda. Untuk memperbaiki masalah, mulailah dengan me-reboot instance dan membuat upaya koneksi lain. Atau, secara manual memulai koneksi awal melalui metode koneksi non-ssm. Untuk informasi selengkapnya tentang Systems Manager, lihat topik [Bekerja dengan Agen Systems Manager](#) di AWS Systems Manager.

Saat start-up, status Amazon EC2 menunjukkan itu berjalan, tetapi koneksi tidak melalui

Jika Anda baru saja memulai atau membuat peran IAM baru untuk sebuah instans dan tidak dapat membuat koneksi, tunggu beberapa menit lagi sebelum melakukan upaya lain untuk membuat koneksi.

## Bekerja dengan Amazon Elastic Container Registry

Amazon Elastic Container Registry (Amazon ECR) adalah layanan registri kontainer terkelola AWS 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 atau gambar yang ditandai.
- Mengakses tag gambar dan repositori URIs atau. ARNs
- 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?](#) di Panduan Pengguna Registri Amazon Elastic Container.

### Topik

- [Bekerja dengan Amazon Elastic Container Registry](#)
- [Membuat layanan App Runner melalui Amazon ECR](#)

## Bekerja dengan Amazon Elastic Container Registry

Anda dapat mengakses layanan Amazon Elastic Container Registry (Amazon ECR) langsung dari Explorer AWS di VS Code dan menggunakannya 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 itu dan beri tag gambar untuk diproses.
3. Buat repositori di dalam instans Amazon ECR Anda.
4. Dorong gambar yang ditandai ke repositori Anda.

### 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 kredensial. Ini agar layanan dapat menentukan apakah Anda memiliki izin untuk mengakses sumber dayanya. Kami tidak menyarankan Anda mengakses AWS langsung melalui kredensial untuk akun

root AWS Anda. Sebagai gantinya, gunakan AWS Identity and Access Management (IAM) untuk membuat pengguna IAM dan kemudian menambahkan pengguna tersebut ke grup IAM dengan izin administratif. Anda kemudian dapat mengakses AWS menggunakan URL khusus dan kredensial untuk pengguna IAM.

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

Untuk membuat pengguna administrator, pilih salah satu opsi berikut.

Pilih salah satu cara untuk mengelola administrator Anda	Untuk	Oleh	Anda juga bisa
Di Pusat Identitas IAM  (Direkomendasikan)	Gunakan kredensial jangka pendek untuk mengakses. AWS  Ini sejalan dengan praktik terbaik keamanan. Untuk informasi tentang praktik terbaik, lihat <a href="#">Praktik terbaik keamanan di IAM</a> di Panduan Pengguna IAM.	Mengikuti petunjuk di <a href="#">Memulai</a> di Panduan AWS IAM Identity Center Pengguna.	Konfigurasi akses terprogram dengan <a href="#">Mengonfigurasi AWS CLI yang akan digunakan AWS IAM Identity Center</a> dalam AWS Command Line Interface Panduan Pengguna.
Di IAM	Gunakan kredensial jangka panjang untuk mengakses. AWS	Mengikuti petunjuk di <a href="#">Buat pengguna IAM untuk akses darurat</a> di Panduan Pengguna IAM.	Konfigurasi akses terprogram dengan <a href="#">Mengelola kunci akses untuk pengguna</a>

Pilih salah satu cara untuk mengelola administrator Anda	Untuk	Oleh	Anda juga bisa
(Tidak direkomendasikan)			<a href="#">IAM di Panduan Pengguna IAM</a> .

Untuk masuk sebagai pengguna IAM baru ini, keluar dari AWS konsol, lalu gunakan URL berikut. Di URL berikut, di mana `your_aws_account_id` adalah nomor AWS akun Anda tanpa tanda hubung (misalnya, jika nomor akun Anda, ID akun Anda AWS adalah): 1234-5678-9012 AWS 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 untuk halaman login Anda berisi ID AWS akun Anda, Anda dapat membuat alias akun. Dari dasbor IAM, pilih Kustomisasi 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 [Menginstal, memperbarui, dan menghapus instalasi panduan pengguna AWS CLI](#) versi 2.

## 1. Membuat Dockerfile

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

### Membuat Dockerfile

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

#### Note

VS Code dapat meminta Anda untuk memilih jenis file atau ekstensi file. Jika ini terjadi, pilih plaintext. Vs Code memiliki ekstensi "dockerfile". Namun, kami tidak menyarankan Anda menggunakannya. Ini karena ekstensi dapat menyebabkan konflik dengan versi tertentu dari Docker atau aplikasi terkait lainnya.

### Edit Dockerfile Anda menggunakan VS Code

Jika Dockerfile Anda memiliki ekstensi file, buka menu konteks (klik kanan) untuk file tersebut 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 gambar Ubuntu 18.04. Petunjuk RUN memperbarui cache paket. Instal paket perangkat lunak untuk server web, lalu tulis “Hello World!” konten ke root dokumen server web. Instruksi EXPOSE mengekspos port 80 pada wadah, dan instruksi CMD memulai server web.

### 3. Simpan Dockerfile Anda.

#### Important

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

## 2. Bangun gambar Anda dari Dockerfile Anda

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

Buat gambar dari Dockerfile Anda

1. Gunakan CLI Docker atau CLI yang terintegrasi dengan instance Docker Anda untuk menavigasi ke direktori yang berisi Dockerfile Anda.
2. Jalankan perintah Docker build untuk membangun gambar yang ditentukan di Dockerfile Anda.

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

3. Jalankan perintah gambar Docker untuk memverifikasi bahwa gambar telah dibuat dengan benar.

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

Example contoh keluaran:

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 saat dijalankan.

Untuk menjalankan image yang baru dibangun, gunakan perintah Docker run.

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

Opsi `-p` yang ditentukan dalam contoh sebelumnya memetakan port 80 yang terbuka pada kontainer ke port 80 dari sistem host. Jika Anda menjalankan Docker secara lokal, navigasikan ke <http://localhost:80> menggunakan browser web Anda. Jika program berjalan dengan benar, “Hello World!” pernyataan ditampilkan.

Untuk informasi selengkapnya tentang perintah Docker run, lihat [referensi Docker run](#) di situs web Docker.

### 3. Buat repositori baru

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

Buat repositori Amazon ECR baru

1. Dari VS Code Activity Bar, pilih ikon AWS Toolkit.
2. Perluas menu AWS Explorer.
3. Temukan AWS Wilayah default yang terkait dengan AWS akun Anda. Kemudian, pilih untuk melihat daftar layanan yang melalui Toolkit for VS Code.
4. Pilih opsi ECR + untuk memulai proses Create new repository.
5. Ikuti petunjuk untuk menyelesaikan proses.
6. Setelah selesai, Anda dapat mengakses repositori baru Anda dari bagian ECR pada menu Explorer. AWS

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

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

- Tarik gambar dari repositori Anda.
- Hapus gambar yang disimpan di repositori Anda.
- Hapus repositori Anda.

## Otentikasi Docker dengan registri default Anda

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 AWS CLI Anda.
2. Gunakan `get-login-password` metode ini untuk mengautentikasi 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

Pada perintah sebelumnya, Anda harus memperbarui **region** dan **AWS\_account\_id** ke informasi yang spesifik untuk akun Anda AWS .

## Tag dan dorong gambar ke repositori Anda

Setelah Anda mengautentikasi Docker dengan instance Anda AWS, dorong gambar ke repositori Anda.

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

```
docker images
```

Example contoh keluaran:

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

2. Tandai gambar Anda dengan perintah tag Docker.

```
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 perintah tag Docker.

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

Example contoh keluaran:

```
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 berhasil diunggah ke repositori Anda, gambar tersebut terlihat di menu Explorer. AWS

Tarik gambar dari Amazon ECR

- Anda dapat menarik gambar ke instance lokal Anda dari perintah tag Docker.

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

Example contoh keluaran:

```
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
```

## Hapus gambar dari repositori Amazon ECR Anda

Ada dua metode untuk menghapus gambar dari VS Code. Metode pertama adalah menggunakan AWS Explorer.

1. Dari AWS Explorer, perluas menu ECR
2. Perluas repositori yang ingin Anda hapus gambarnya
3. Pilih tag gambar yang terkait dengan gambar yang ingin Anda hapus, dengan membuka menu konteks (klik kanan)
4. Pilih Delete Tag... pilihan untuk menghapus semua gambar yang disimpan terkait dengan tag itu

## Hapus gambar menggunakan AWS CLI

- Anda juga dapat menghapus gambar dari repositori Anda dengan perintah AWS `batch-delete-imageecr`.

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

## Example contoh keluaran:

```
{  
  "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 Explorer.

1. Dari AWS Explorer, perluas menu ECR
2. Pilih repositori yang ingin Anda hapus dengan membuka menu konteks (klik kanan)
3. Pilih Delete Repository... opsi ke repositori yang dipilih

## Hapus repositori Amazon ECR dari CLI AWS

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

### Note

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

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

## Example contoh keluaran:

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

```
}
```

## Membuat layanan App Runner melalui Amazon ECR

Topik berikut menjelaskan cara membuat dan meluncurkan AWS App Runner layanan dari simpul Amazon Elastic Container Registry (Amazon ECR) Registry ECR), di. AWS Toolkit for Visual Studio Code Untuk informasi terperinci tentang layanan ECR Amazon AWS App Runner dan Amazon, lihat Panduan Pengguna [ECR Amazon AWS App Runner dan Amazon](#).

### Prasyarat

Sebelum Anda dapat membuat dan meluncurkan AWS App Runner dari Amazon ECR di AWS Toolkit, Anda harus menyelesaikan yang berikut ini. Untuk panduan terperinci tentang cara menyelesaikan prosedur ini, lihat topik [Bekerja dengan Amazon Elastic Container Registry](#) di Panduan Pengguna ini.

1. Buat `adockerfile`.
2. Bangun gambar dari `Andadockerfile`.
3. Buat repositori baru.
4. Tandai dan dorong gambar ke repositori Anda.

## Membuat AWS App Runner layanan dari repositori Amazon ECR yang ada

Prosedur berikut menjelaskan cara membuat AWS App Runner layanan dari repositori Amazon ECR yang ada, di Toolkit. AWS

1. Dari AWS Explorer, perluas wilayah yang berisi repositori Amazon ECR tempat Anda ingin membuat AWS App Runner layanan.
2. Perluas node layanan Amazon ECR untuk melihat repositori Amazon ECR Anda.
3. Buka menu konteks untuk (klik kanan) repositori Amazon ECR atau gambar repositori yang ingin Anda buat layanan. AWS App Runner
4. Dari menu konteks, pilih **Create App Runner Service** untuk membuka wizard AWS App Runner pembuatan di VS Code
5. Dari Masukkan port untuk layanan baru (1/5), masukkan nomor port yang ingin Anda gunakan, lalu tekan **Enter** untuk melanjutkan.

6. Dari Konfigurasi variabel lingkungan (2/5), pilih Gunakan file... untuk menelusuri untuk memilih jelajahi file lokal Anda atau pilih Lewati untuk melewati langkah ini.
7. Dari Pilih peran yang akan ditarik dari ECR (3/5), pilih peran IAM yang ada dari daftar.

#### Note

AppRunnerECRAccessPeran akses peran diperlukan untuk membuat AWS App Runner layanan dari registri pribadi Amazon ECR. Jika peran yang valid tidak tersedia dari daftar, pilih + (Buat Peran...) ikon untuk secara otomatis membuat dan menetapkan AppRunnerECRAccessPeran ke registri Anda.

8. Dari Nama layanan Anda (4/5), masukkan nama untuk layanan baru Anda, lalu tekan **Enter** untuk melanjutkan.
9. Dari Pilih konfigurasi instans (5/5) pilih **Memory** konfigurasi **vCPU** dan dari daftar untuk membuat layanan baru Anda.
10. Dari AWS Explorer, perluas node layanan App Runner untuk melihat AWS App Runner sumber daya Anda. Ketika layanan baru Anda telah berhasil dibuat, status secara otomatis diperbarui ke Running.

## Bekerja dengan Amazon Elastic Container Service

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

### Topik

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

## Menggunakan IntelliSense file definisi tugas 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 Layanan Kontainer Elastis Amazon. Saat Anda menginstal AWS Toolkit for Visual Studio Code, instalasi menyertakan IntelliSense fungsionalitas untuk file definisi tugas Amazon ECS.

## Prasyarat

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

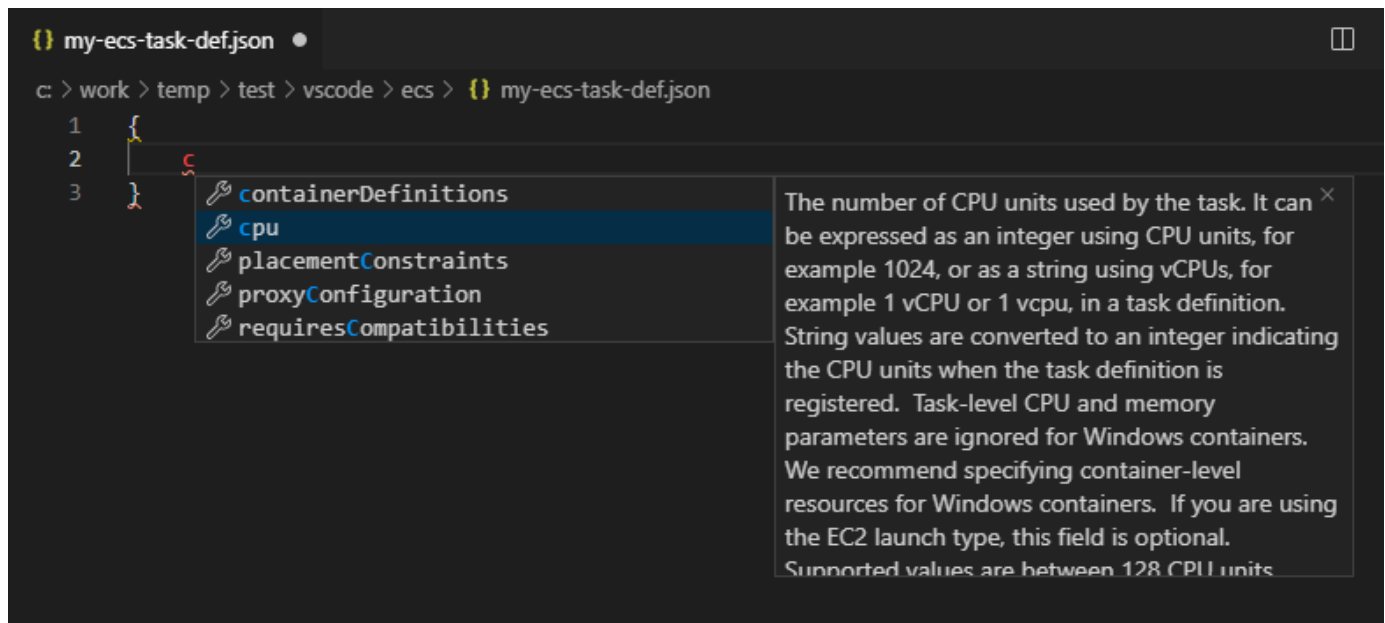
## Gunakan IntelliSense dalam file definisi tugas Amazon ECS

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

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

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

2. Buka file di editor VS Code dan masukkan kurawal kurawal kurawal.
3. Masukkan huruf “c” seolah-olah Anda cpu ingin menambahkan definisi. Amati IntelliSense dialog yang terbuka, yang mirip 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 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 [Amazon ECS dioptimalkan AMIs](#).
- 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

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

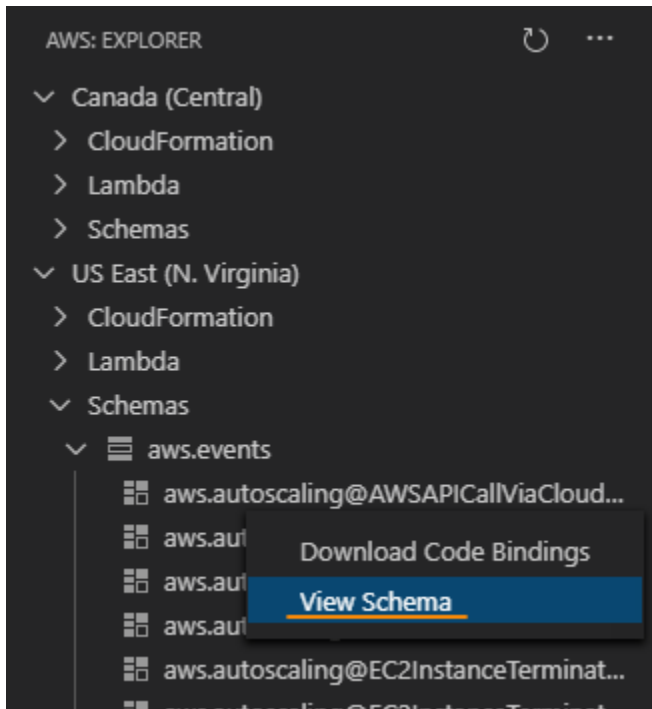
### Prasyarat

- 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.

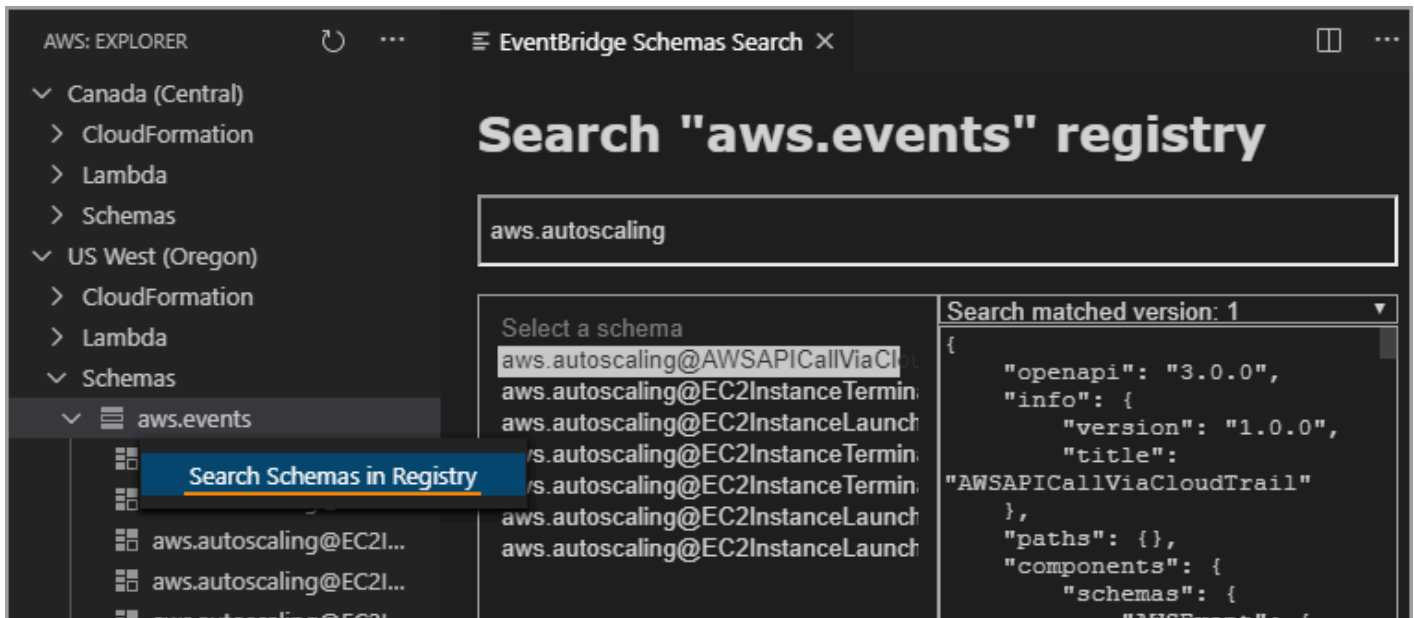


## Temukan Skema yang Tersedia

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 Penganalisis Akses IAM

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

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

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

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 kebijakan IAM dan pemeriksaan kebijakan kustom di 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 khusus IAM Access Analyzer](#).

### Prasyarat

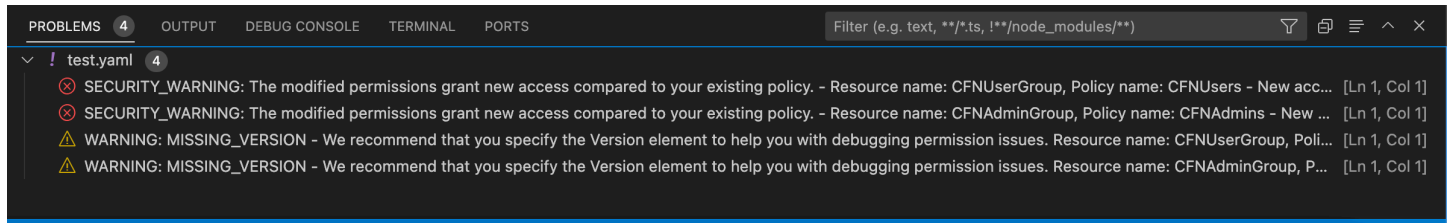
Prasyarat berikut harus dipenuhi sebelum Anda dapat bekerja dengan pemeriksaan kebijakan IAM Access Analyzer dari Toolkit.

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

### Pemeriksaan kebijakan IAM Access Analyzer

Anda dapat melakukan pemeriksaan kebijakan untuk CloudFormation templat, paket Terraform, dan dokumen Kebijakan JSON, menggunakan AWS Toolkit for Visual Studio Code Temuan pemeriksaan

Anda dapat dilihat di Panel Masalah Kode VS. Gambar berikut menunjukkan Panel Masalah Kode VS.



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

## Menjalankan Kebijakan Validasi

Pemeriksaan Kebijakan Validasi, juga dikenal sebagai validasi kebijakan, memvalidasi kebijakan Anda terhadap tata bahasa kebijakan IAM dan praktik terbaik. AWS Untuk informasi tambahan, lihat [Tata Bahasa bahasa kebijakan IAM JSON](#) dan [praktik terbaik AWS Keamanan dalam topik IAM](#), yang terdapat di Panduan Pengguna. AWS Identity and Access Management

1. Dari VS Code, buka file yang didukung yang berisi Kebijakan AWS IAM, 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 Kebijakan IAM di editor Kode VS.
3. Dari panel Pemeriksaan Kebijakan IAM, 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 tindakan IAM 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 URI bucket Amazon S3. Untuk menggunakan URI bucket Amazon S3, 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 Kebijakan AWS IAM, 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 Kebijakan IAM di editor Kode VS.
3. Dari panel Pemeriksaan Kebijakan IAM, 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 kembaliPASS.

### 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 Kebijakan AWS IAM, di editor Kode VS.
2. Untuk membuka pemeriksaan kebijakan IAM Access Analyzer, buka Palet Perintah Kode VS dengan menekan **CRTL+Shift+P**, cari **IAM Policy Checks**, lalu klik untuk membuka panel Pemeriksaan Kebijakan IAM di editor Kode VS.
3. Dari panel Pemeriksaan Kebijakan IAM, pilih jenis dokumen Anda dari menu tarik-turun.
4. Dari bagian Pemeriksaan Kebijakan Kustom, pilih CheckNoNewAccess.
5. Masukkan dokumen kebijakan JSON referensi. Atau, Anda dapat menyediakan jalur file yang mereferensikan dokumen kebijakan JSON.
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 kembaliPASS.

### 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 Kebijakan AWS IAM, di editor Kode VS.
2. Untuk membuka pemeriksaan kebijakan IAM Access Analyzer, buka Palet Perintah Kode VS dengan menekan **CRTL+Shift+P**, cari **IAM Policy Checks**, lalu klik untuk membuka panel Pemeriksaan Kebijakan IAM di editor Kode VS.
3. Dari panel Pemeriksaan Kebijakan IAM, 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 in AWS Toolkit for Visual Studio Code memungkinkan Anda untuk berinteraksi dengan AWS IoT layanan, sambil meminimalkan gangguan pada alur kerja Anda di VS Code. Panduan pengguna ini dimaksudkan untuk membantu Anda mulai menggunakan fitur AWS IoT layanan yang tersedia di AWS Toolkit for Visual Studio Code. Untuk informasi tambahan tentang AWS IoT layanan, lihat panduan pengembang [Apa itu AWS IoT?](#)

### AWS IoT prasyarat

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

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

### AWS IoT Hal-hal

AWS IoT menghubungkan perangkat ke layanan AWS cloud dan sumber daya. Anda dapat menghubungkan perangkat Anda AWS IoT dengan menggunakan objek yang disebut benda.

Sesuatu 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 berbagai hal, lihat panduan pengembang [Mengelola perangkat dengan AWS IoT](#).

## Mengelola AWS IoT hal-hal

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

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

Untuk menciptakan sesuatu

1. Dari AWS Explorer, perluas judul layanan IoT, dan pilih konteks (klik kanan) Things.
2. Pilih Create Thing dari menu konteks untuk membuka kotak dialog.
3. Ikuti prompt dengan memasukkan nama untuk hal IoT Anda ke dalam bidang Thing Name.
4. Ketika ini selesai, ikon benda yang diikuti dengan nama yang Anda tentukan akan terlihat di bagian Thing.

Untuk melampirkan sertifikat pada suatu hal

1. Dari AWS Explorer, perluas bagian layanan IoT.
2. Di bawah subbagian Things, temukan hal di mana Anda melampirkan sertifikat.
3. Pilih konteks (klik kanan) benda itu dan pilih Lampirkan Sertifikat dari menu konteks, untuk membuka pemilih input 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 di AWS explorer, sebagai item dari benda yang Anda lampirkan.

Untuk melepaskan sertifikat dari suatu hal

1. Dari AWS Explorer, perluas bagian layanan IoT
2. Di subbagian Things, temukan hal yang ingin Anda lepaskan sertifikatnya.
3. Pilih konteks (klik kanan) benda itu dan pilih Lepaskan Sertifikat dari menu konteks.
4. Ketika ini selesai, sertifikat terpisah tidak akan lagi ditampilkan di bawah benda itu di AWS Explorer, tetapi masih dapat diakses dari subbagian Sertifikat.

Untuk menghapus sesuatu

1. Dari AWS Explorer, perluas bagian layanan IoT.
2. Di subbagian Things, cari hal yang ingin Anda hapus.
3. Konteks-pilih (klik kanan) benda itu dan pilih Delete Thing dari menu konteks untuk menghapusnya.
4. Ketika ini selesai, hal yang dihapus tidak akan lagi tersedia dari subbagian Things.

#### Note

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

## AWS IoT sertifikat

Sertifikat adalah cara umum untuk membuat koneksi aman antara AWS IoT layanan dan perangkat Anda. Sertifikat X.509 adalah sertifikat digital yang menggunakan standar infrastruktur kunci publik X.509 untuk mengaitkan kunci publik dengan identitas yang terkandung dalam sertifikat. Untuk informasi tambahan tentang AWS IoT sertifikat, lihat panduan pengembang [Otentikasi \(IoT\)](#).

## Mengelola sertifikat

Toolkit VS Code menawarkan berbagai cara bagi Anda untuk mengelola AWS IoT sertifikat Anda, langsung dari AWS Explorer.


- [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 Anda. AWS IoT


1. Dari AWS Explorer, perluas bagian layanan IoT, dan pilih konteks (klik kanan) Sertifikat.
2. Pilih Buat 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 Anda.

 Note

- Nama file default berisi ID sertifikat sebagai awalan.
- Hanya sertifikat X.509 yang disimpan dengan AWS akun Anda, melalui layanan. AWS IoT
- Key pair RSA Anda hanya dapat diterbitkan sekali, 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 akun Anda AWS .

Untuk mengubah status sertifikat

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

 Note

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

- Sertifikat yang telah dicabut tidak dapat diaktifkan kembali.

1. Dari AWS Explorer, perluas bagian layanan IoT.
  2. Di sub-bagian Sertifikat, 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 menjadi aktif.
  - Jika sertifikat memiliki status aktif, pilih nonaktifkan untuk mengubah status menjadi tidak aktif.
  - Jika sertifikat memiliki status aktif atau tidak aktif, pilih cabut untuk mengubah status yang dicabut.

#### Note

Masing-masing tindakan perubahan status ini juga tersedia jika Anda memilih sertifikat yang dilampirkan pada suatu benda saat ditampilkan di subbagian Things.

Untuk melampirkan kebijakan IoT ke sertifikat

1. Dari AWS Explorer, perluas bagian layanan IoT.
2. Di sub-bagian Sertifikat, cari sertifikat yang ingin Anda ubah.
3. Pilih konteks (klik kanan) sertifikat dan pilih Lampirkan Kebijakan dari menu konteks, untuk membuka pemilih input dengan daftar kebijakan yang tersedia.
4. Pilih kebijakan yang ingin Anda lampirkan ke sertifikat.
5. Ketika ini selesai, kebijakan yang Anda pilih akan ditambahkan ke sertifikat sebagai item sub-menu.

Untuk melepaskan kebijakan IoT dari sertifikat

1. Dari AWS Explorer, perluas bagian layanan IoT.
2. Di sub-bagian Sertifikat, 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 sub-bagian Kebijakan.

Untuk menghapus sertifikat

1. Dari AWS Explorer, perluas judul layanan IoT.
2. Di sub-bagian Sertifikat, cari sertifikat yang ingin Anda hapus.
3. Konteks-pilih (klik kanan) sertifikat dan pilih Hapus Sertifikat dari menu konteks.

#### Note

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

## AWS IoT kebijakan

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

#### Note


Kebijakan bernama diberi versi sehingga Anda dapat memutarnya kembali. Di The AWS Explorer, kebijakan IoT Anda tercantum di bawah subbagian Kebijakan, di 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 kebijakan layanan AWS IoT Anda. Ini adalah cara Anda dapat mengelola atau mengubah kebijakan Anda langsung dari AWS Explorer di VS Code:

- [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 di sistem file Anda.

1. Dari AWS Explorer, perluas bagian layanan IoT.
2. Pilih konteks (klik kanan) subbagian Kebijakan dan pilih Buat Kebijakan dari Dokumen, untuk membuka kolom input Nama Kebijakan.
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 saat ini selesai.

Untuk mengunggah versi AWS IoT kebijakan baru

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, perluas bagian layanan IoT.
2. Perluas subbagian Kebijakan untuk melihat kebijakan Anda AWS IoT

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

#### Untuk mengedit versi AWS IoT kebijakan

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

1. Dari AWS Explorer, perluas bagian layanan IoT.
2. Perluas subbagian Kebijakan dan temukan kebijakan yang ingin Anda perbarui. Buat Kebijakan dari Dokumen untuk membuka kolom input Nama Kebijakan.
3. Perluas kebijakan yang ingin Anda perbarui, lalu pilih Konteks (klik kanan) versi kebijakan yang ingin Anda edit.
4. Pilih Lihat dari menu konteks untuk membuka versi kebijakan di VS Code
5. Saat 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 AWS Explorer, ulangi langkah-langkah yang dijelaskan dalam [Upload a new policy version](#) prosedur.

#### Untuk memilih default versi kebijakan baru

1. Dari AWS Explorer, perluas bagian layanan IoT.
2. Perluas subbagian Kebijakan dan temukan kebijakan yang ingin Anda perbarui.
3. Perluas kebijakan yang ingin Anda perbarui, lalu pilih Konteks (klik kanan) versi kebijakan yang ingin disetel, lalu pilih Setel 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 kondisi yang harus dipenuhi.

- Anda tidak dapat menghapus kebijakan jika dilampirkan 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 AWS Explorer, perluas bagian layanan IoT.
2. Perluas subbagian Kebijakan dan temukan kebijakan yang ingin Anda perbarui.
3. Perluas kebijakan yang ingin Anda perbarui, lalu pilih Konteks (klik kanan) versi kebijakan yang ingin dihapus, lalu pilih Hapus.
4. Ketika versi dihapus, itu tidak akan lagi terlihat dari Explorer.
5. Jika satu-satunya versi yang tersisa untuk kebijakan adalah default, Anda dapat memilih konteks (klik kanan) kebijakan induk dan memilih Hapus untuk menghapusnya.

## AWS Lambda Fungsi

AWS Toolkit for Visual Studio Code Ini memberikan dukungan komprehensif untuk AWS Lambda fungsi, memungkinkan Anda untuk membangun, menguji, dan menyebarkan langsung dari VS Code.

Lambda adalah layanan komputasi berbasis peristiwa yang dikelola sepenuhnya yang secara otomatis menjalankan kode Anda sebagai respons terhadap peristiwa dari lebih dari 200 layanan AWS dan aplikasi (software-as-a-serviceSaaS). Untuk informasi rinci tentang AWS Lambda layanan, lihat Panduan [AWS Lambda](#) Pengembang.

Topik-topik berikut menjelaskan cara bekerja dengan AWS Lambda di. AWS Toolkit for Visual Studio Code

## Topik

- [Bekerja dengan AWS Lambda Fungsi](#)
- [AWS Lambda console ke IDE](#)
- [AWS Lambda dengan LocalStack dukungan](#)
- [AWS Lambda debugging jarak jauh](#)

## Bekerja dengan AWS Lambda Fungsi

AWS Toolkit for Visual Studio Code Ini memungkinkan Anda untuk bekerja dengan AWS Lambda fungsi Anda di lingkungan VS Code lokal Anda. Dengan AWS Toolkit, Anda dapat membuat, mengedit, menguji, men-debug, dan menerapkan fungsi Lambda Anda, tanpa harus meninggalkan IDE. Untuk informasi rinci tentang AWS Lambda layanan, lihat Panduan [AWS Lambda](#) Pengembang.

Bagian berikut menjelaskan cara memulai bekerja dengan fungsi Lambda di. AWS Toolkit for Visual Studio Code

### Note

Jika Anda telah membuat fungsi Lambda dengan menggunakan Konsol Manajemen AWS, maka Anda dapat memanggilnya dari Toolkit. Selain itu, Anda dapat membuka fungsi Lambda Anda ke Kode VS dari AWS Lambda console, untuk informasi tambahan, lihat [AWS Lambda console ke IDE](#) topik di panduan pengguna ini. Untuk membuat fungsi Lambda baru di VS Code, ikuti langkah-langkah yang diuraikan dalam topik [Membuat aplikasi tanpa server baru \(lokal\)](#) dalam panduan pengguna ini.

## Prasyarat

Ketentuan berikut harus dipenuhi untuk bekerja dengan AWS Lambda layanan di AWS Toolkit.

- Versi terbaru diinstal dan diatur dengan AWS kredensi Anda. AWS Toolkit for Visual Studio Code
- Izin dan kebijakan terkelola AWS Identity and Access Management (IAM) Anda dikonfigurasi untuk bekerja dengan layanan. AWS Lambda Untuk informasi terperinci tentang cara mengonfigurasi izin dan membuat kebijakan AWS terkelola yang kompatibel, lihat AWS Lambda topik [AWS Identity and Access Management untuk](#) di Panduan AWS Lambda Pengembang.

- Anda memiliki AWS Lambda fungsi yang ada atau terbiasa dengan cara membuatnya. Untuk petunjuk tentang cara membuat fungsi Lambda, lihat topik [Membuat fungsi Lambda pertama Anda di Panduan Pengembang](#).AWS Lambda

## Memanggil Fungsi Lambda

Untuk menjalankan fungsi Lambda dari akun AWS Anda ke VS Code, selesaikan langkah-langkah berikut.

1. Dari AWS Toolkit for Visual Studio Code, perluas AWS penjelajah.
2. Dari AWS penjelajah, perluas Lambda untuk melihat sumber daya Lambda Anda.
3. Buka menu konteks untuk (klik kanan) fungsi Lambda yang ingin Anda panggil, lalu pilih Panggil di cloud atau pilih ikon Panggil di cloud untuk membuka menu konfigurasi pemanggilan jarak jauh di VS Code.
4. Dari menu konfigurasi pemanggilan jarak jauh, tentukan pengaturan Payload Anda dan tambahkan informasi tambahan apa pun yang diperlukan untuk acara tersebut.

### Note

Proses pemanggilan pertama mungkin mulai berjalan segera setelah Anda memilih Invoke di cloud di AWS explorer. Output ditampilkan di tab OUTPUT dari terminal VS Code.

5. Pilih tombol Remote Invoke untuk menjalankan fungsi Anda, Output ditampilkan di tab OUTPUT dari terminal VS Code.

## Menghapus fungsi Lambda

Untuk menghapus fungsi Lambda, selesaikan prosedur berikut.

### Warning

Jangan gunakan prosedur ini untuk menghapus fungsi Lambda yang terkait dengannya. [CloudFormation](#) Fungsi-fungsi ini harus dihapus melalui CloudFormation tumpukan Anda.

1. Dari AWS Toolkit for Visual Studio Code, perluas AWS penjelajah.

2. Dari AWS penjelajah, perluas Lambda untuk melihat sumber daya Lambda Anda.
3. Klik kanan fungsi Lambda yang ingin Anda hapus, lalu pilih Hapus.
4. Saat diminta, konfirmasikan bahwa Anda ingin menghapus fungsi Anda.

Setelah fungsi dihapus, itu tidak lagi terdaftar di AWS explorer.

## Mengunduh fungsi Lambda

Anda dapat mengunduh kode dari fungsi Lambda jarak jauh ke ruang kerja VS Code Anda untuk mengedit dan men-debug.


### Note

Untuk mengunduh fungsi Lambda Anda, Anda harus bekerja di ruang kerja VS Code dengan folder yang dapat diakses dan AWS Toolkit hanya mendukung fitur ini dengan fungsi Lambda menggunakan runtime Node.js dan Python.

1. Dari AWS Toolkit for Visual Studio Code, perluas AWS penjelajah.
2. Dari AWS penjelajah, perluas Lambda untuk melihat sumber daya Lambda Anda.
3. Klik kanan fungsi Lambda yang ingin Anda unduh, lalu pilih Unduh.
4. Fungsi Lambda Anda terbuka di editor VS Code dan ditampilkan di AWS explorer saat unduhan selesai. AWS Toolkit juga membuat konfigurasi peluncuran di panel run VS Code yang memungkinkan Anda menjalankan dan men-debug fungsi Lambda secara lokal. AWS Serverless Application Model Untuk informasi selengkapnya tentang penggunaan AWS SAM, lihat [the section called “Menjalankan dan men-debug aplikasi tanpa server dari template \(lokal\)”](#).

## Menyebarkan pembaruan untuk fungsi Lambda baru

Anda dapat menerapkan pembaruan ke fungsi Lambda baru dari lokasi sementara yang tidak ditentukan di komputer lokal Anda.

 Note

Ketika ada perubahan yang tidak diterapkan pada file lambda Anda, Anda akan diberi tahu oleh ikon M yang terletak di sebelah file yang dimodifikasi di editor Kode VS dan di explorer. AWS

### Menerapkan dari editor VS Code


1. Buka file dari fungsi Lambda Anda di editor VS Code, lalu buat perubahan pada file tersebut.
2. Simpan secara manual dari menu utama Kode VS atau menekan **option+s** (Mac) **ctrl+s** (Windows).
3. VS Code secara otomatis meminta Anda tentang penerapan perubahan Anda ke cloud, pilih tombol Deploy untuk mengonfirmasi penerapan.
4. VS Code memperbarui status penerapan Anda dan memberi tahu Anda saat proses selesai.

### Menyebarkan dari Explorer AWS

1. Buka file dari fungsi Lambda Anda di editor VS Code, lalu buat perubahan pada file tersebut.
2. Dari AWS Toolkit, perluas AWS explorer.
3. Dari AWS penjelajah, perluas AWS wilayah dengan fungsi Lambda yang ingin Anda gunakan perubahannya.
4. Dari AWS wilayah tersebut, perluas Lambda dan navigasikan fungsi yang ingin Anda gunakan untuk perubahannya.
5. Dari menu cepat di sebelah fungsi Anda, pilih ikon Simpan dan terapkan kode Anda.
6. VS Code memperbarui status penerapan Anda dan memberi tahu Anda saat proses selesai.

### Mengunggah pembaruan untuk fungsi Lambda yang ada

Prosedur berikut menjelaskan cara mengunggah perubahan lokal yang dibuat ke fungsi Lambda Anda yang ada. Fitur ini mendukung unggahan dengan runtime yang didukung Lambda apa pun.

 Warning

Sebelum mengunggah fungsi lambda Anda, perhatikan hal-hal berikut:

- Memperbarui kode dengan cara ini tidak menggunakan AWS SAM CLI untuk penerapan atau membuat tumpukan CloudFormation
- AWS Toolkit tidak memvalidasi kode. Validasi kode Anda dan uji fungsi Anda sebelum mengunggah perubahan apa pun ke cloud.

### Mengunggah Arsip Zip

1. Dari AWS Toolkit for Visual Studio Code, perluas AWS penjelajah.
2. Dari AWS penjelajah, perluas Lambda untuk melihat sumber daya Lambda Anda.
3. Klik kanan fungsi Lambda yang ingin Anda unggah perubahan, lalu pilih Unggah Lambda... untuk membuka menu Pilih Jenis Unggah.
4. Pilih Arsip ZIP untuk menemukan ZIP Archive di direktori lokal Anda.
5. Saat diminta, konfirmasi unggahan untuk memulai unggahan yang dipilih ZIP Archive.
6. Status unggahan Anda ditampilkan dalam Kode VS dan Anda akan diberi tahu saat proses pengunggahan selesai.

### Mengunggah direktori tanpa membangun

1. Dari AWS Toolkit for Visual Studio Code, perluas AWS penjelajah.
2. Dari AWS penjelajah, perluas Lambda untuk melihat sumber daya Lambda Anda.
3. Klik kanan fungsi Lambda yang ingin Anda unggah perubahan, lalu pilih Unggah Lambda... untuk membuka menu Pilih Jenis Unggah.
4. Pilih Direktori untuk melanjutkan ke direktori Build layar.
5. Dari layar Membangun direktori, pilih Tidak untuk memilih direktori lokal untuk diunggah.
6. Saat diminta, konfirmasi unggahan untuk mengunggah direktori yang dipilih.
7. Status unggahan Anda ditampilkan dalam Kode VS dan Anda akan diberi tahu saat proses pengunggahan selesai.

### Mengunggah direktori dengan build

#### Note

Ketahui hal-hal berikut:

- Prosedur ini membutuhkan AWS Serverless Application Model CLI.
- AWS Toolkit memberi tahu Anda bahwa handler yang cocok tidak dapat dideteksi sebelum mengunggah.
- Untuk mengubah handler yang terpasang pada fungsi Lambda Anda, gunakan tombol atau AWS Lambda console . AWS Command Line Interface

1. Dari AWS Toolkit for Visual Studio Code, perluas AWS penjelajah.
2. Dari AWS penjelajah, perluas Lambda untuk melihat sumber daya Lambda Anda.
3. Klik kanan fungsi Lambda yang ingin Anda unggah perubahan, lalu pilih Unggah Lambda... untuk membuka menu Pilih Jenis Unggah.
4. Pilih Direktori untuk melanjutkan ke direktori Build layar.
5. Dari layar Build directory, pilih Ya, lalu pilih direktori lokal untuk diunggah.
6. Saat diminta, konfirmasi unggahan untuk mulai membangun dan mengunggah direktori yang dipilih.
7. Status unggahan Anda ditampilkan dalam Kode VS dan Anda akan diberi tahu saat proses pengunggahan selesai.

## Mengonversi fungsi Lambda Anda menjadi proyek AWS SAM

Untuk mengonversi fungsi Lambda Anda menjadi AWS SAM tumpukan, selesaikan langkah-langkah berikut.

### Warning

Saat ini, hanya sebagian sumber daya yang didukung saat mengonversi fungsi Lambda menjadi proyek. AWS SAM Untuk menemukan sumber daya yang hilang setelah konversi, periksa konsol Lambda dan tambahkan secara manual ke template Anda AWS SAM . Untuk detail tambahan tentang sumber daya yang didukung dan tidak didukung, lihat topik [dukungan jenis sumber daya](#) di Panduan AWS CloudFormation Pengembang.

1. Dari AWS Toolkit, perluas AWS explorer.
2. Dari AWS penjelajah, perluas AWS wilayah dengan fungsi Lambda yang ingin Anda ubah menjadi AWS SAM proyek.

3. Dari AWS wilayah tersebut, perluas Lambda dan navigasikan fungsi yang ingin Anda ubah menjadi AWS SAM tumpukan.
4. Dari menu cepat di sebelah fungsi Lambda Anda, pilih ikon Konversi ke Aplikasi SAM untuk menelusuri sistem file lokal Anda dan menentukan lokasi untuk proyek baru AWS SAM Anda.
5. Setelah menentukan lokasi, AWS Toolkit mulai mengubah fungsi Lambda Anda menjadi AWS SAM proyek, VS Code memberikan pembaruan tentang status proses.

#### Note

Proses ini mungkin memerlukan waktu beberapa menit.

6. Saat diminta oleh VS Code, masukkan nama tumpukan, lalu tekan **Enter** tombol untuk melanjutkan.
7. VS Code terus memperbarui Anda dengan status proyek Anda, kemudian memberi tahu Anda saat proses selesai dan membuka AWS SAM proyek baru Anda sebagai ruang kerja VS Code.

## AWS Lambda console ke IDE

Fitur AWS Lambda console to IDE memungkinkan Anda mengunduh AWS Lambda fungsi Anda dari Kode VS AWS Lambda console ke dalam. Bekerja dengan fungsi Lambda Anda di VS Code memberi Anda akses ke opsi pengembangan lokal lainnya seperti AWS Serverless Application Model (AWS SAM) dan file. AWS Cloud Development Kit (AWS CDK)

Untuk informasi selengkapnya AWS Lambda, lihat Panduan [AWS Lambda](#) Pengembang. Untuk mulai bekerja dengan fungsi Lambda Anda di AWS Toolkit, lihat topik [Bekerja dengan AWS Lambda Fungsi](#) dalam panduan pengguna ini. Bagian berikut menjelaskan cara memindahkan alur kerja Anda dari konsol Lambda ke VS Code. Untuk informasi mendetail tentang memindahkan fungsi Lambda Anda dari konsol Lambda ke VS Code, termasuk cara mulai bekerja dengan konsol Lambda, lihat topik Mengembangkan [fungsi Lambda secara lokal dengan VS Code di Panduan](#) Pengembang. AWS Lambda

### Pindah dari konsol ke pengembangan lokal

Untuk membuka fungsi Lambda dari konsol Lambda di VS Code, selesaikan langkah-langkah berikut:

1. Dari browser web Anda, buka konsol [Lambda](#).
2. Dari konsol Lambda, pilih fungsi yang ingin Anda buka di VS Code.

3. Dari tampilan fungsi, arahkan ke tab Sumber kode.
4. Dari tab Sumber kode, pilih Buka di VS Code.

## Bekerja dengan fungsi Lambda Anda di VS Code

Saat fungsi Lambda Anda terbuka di VS Code melalui konsol Lambda:

- VS Code secara otomatis diluncurkan di komputer lokal Anda.
- Fungsi Lambda Anda terbuka sebagai ruang kerja VS Code.
- Lambda Anda handler file terbuka di editor Kode VS.

### Note

Jika tidak ada yang dikonfigurasi dengan benar handler file di ruang kerja, tidak ada file yang terbuka di editor Kode VS.

Membuka fungsi Lambda Anda di VS Code melalui konsol Lambda memungkinkan Anda mengakses semua fitur AWS Lambda Toolkit yang ada, termasuk kemampuan untuk mengedit kode fungsi dengan dukungan bahasa penuh, pengujian lokal, debugging jarak jauh, dukungan penerapan, dan manajemen ketergantungan. Untuk informasi selengkapnya tentang fitur Lambda yang didukung di AWS Toolkit, lihat daftar isi [AWS Lambda](#) layanan dalam panduan pengguna ini.

## AWS Lambda dengan LocalStack dukungan

Buat, uji, dan debug aplikasi tanpa server Anda dengan LocalStack dukungan di. AWS Toolkit for Visual Studio Code LocalStack adalah emulator AWS Cloud yang memungkinkan pengujian lokal aplikasi tanpa server.

Untuk informasi selengkapnya AWS Lambda, lihat Panduan [AWS Lambda](#) Pengembang. Untuk mempelajari lebih lanjut LocalStack, kunjungi situs web mereka [LocalStack](#).

## Prasyarat

Berikut ini adalah prasyarat untuk bekerja dengan LocalStack di VS Code.

**Note**

LocalStack CLI diinstal selama proses pengaturan, tetapi jika Anda lebih suka versi LocalStack CLI yang berbeda, versi minimum yang diperlukan adalah 4.8.0.

- Akun Aplikasi LocalStack Web diperlukan untuk akses ke semua fitur yang tersedia untuk LocalStack tingkatan gratis dan berbayar. LocalStack edisi komunitas tersedia tanpa akun.
- Docker diperlukan untuk bekerja dengan LocalStack di VS Code. Untuk informasi selengkapnya tentang LocalStack persyaratan untuk Docker, lihat topik LocalStack [Docker Images](#) dalam dokumentasi. LocalStack
- Direkomendasikan: The AWS Command Line Interface (AWS CLI) membantu Anda dalam bekerja dengan layanan di lingkungan cloud simulasi Anda.

## Instalasi LocalStack

Untuk menginstal versi berjenjang LocalStack gratis dan berbayar, selesaikan langkah-langkah berikut.

**Note**

Untuk petunjuk tentang cara mengatur edisi LocalStack Komunitas, lihat Konten LocalStackkomunitas di LocalStack bagian Menyiapkan topik ini.

1. Dari AWS Toolkit, perluas penjelajah APPLICATION BUILDER.
2. Pilih tombol Buka Walkthrough untuk membuka tab Panduan Mulai membangun aplikasi Anda di editor Kode VS.
3. Dari panduan, pilih Instal LocalStack untuk memulai proses LocalStack instalasi di VS Code.

## Menyiapkan LocalStack

Setelah Anda menginstal LocalStack ekstensi untuk VS Code, Anda mungkin melihat salah satu indikator berikut saat penyiapan diperlukan:

- Di VS Code Status Bar, yang terletak di sudut kiri bawah IDE secara default, LocalStack statusnya berwarna merah.

- VS Code meminta Anda untuk mengatur LocalStack.

Ada dua jenis pengaturan dan konfigurasi untuk LocalStack, tergantung pada versi yang LocalStack Anda gunakan. Bagian tab berikut menjelaskan setiap proses LocalStack penyiapan.

#### Note

LocalStack token autentikasi diperlukan untuk versi tingkat gratis dan berbayar. LocalStack Untuk informasi spesifik tentang LocalStack harga, lihat panduan harga [Pilih paket Anda](#).

### LocalStack tingkatan gratis dan berbayar

Ada 2 cara untuk mengatur LocalStack.

- Dari VS Code Setup LocalStack untuk memulai prompt, pilih tombol Setup.
- Dari bilah status VS Code, pilih ikon LocalStack status untuk membuka prompt Setup LocalStack to get start, lalu pilih tombol Setup.

Selama pengaturan, sistem melewati langkah-langkah berikut:

1. Menginstal LocalStack CLI.
2. Periksa untuk melihat apakah Anda memiliki LocalStack akun.
3. Jika Anda memiliki LocalStack akun, sistem memandu Anda melalui proses otentikasi di browser web default Anda. Demikian pula, jika Anda tidak memiliki LocalStack akun, sistem memandu Anda melalui pengaturan akun sebelum proses otentikasi.

Setelah LocalStack diatur, LocalStack status diperbarui di bilah status Kode VS.

#### Note

Jika Anda belum membuat AWS profil untuk LocalStack, maka yang baru secara otomatis dibuat untuk Anda sebagai bagian dari proses LocalStack penyiapan.

## LocalStack Komunitas

Edisi Komunitas gratis untuk digunakan dan tidak mengharuskan Anda untuk mendaftar akun, itu berjalan dari gambar Docker yang tidak memerlukan lisensi. LocalStack Untuk detail tambahan tentang Edisi LocalStack Komunitas, lihat dokumentasi [gambar LocalStack Komunitas](#). Bagian berikut menjelaskan prasyarat dan pengaturan dasar yang diperlukan untuk bekerja dengan edisi LocalStack komunitas di VS Code.

### Meluncurkan instance baru

Untuk meluncurkan instance baru LocalStack Komunitas, selesaikan prosedur berikut.

#### Note

Contoh berikut memulai instance kontainer LocalStack pada port 4566. Jika Anda menentukan nilai port yang berbeda, Anda harus memperbarui nilai port yang ditentukan dalam prosedur yang terletak di bagian Configuring the AWS CLI and AWS Toolkit.

1. Dari VS Code, buka terminal VS Code dengan menekan **ctrl + ` (backtick)**.
2. Masukkan yang berikut ini ke terminal.

Mac:

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

Windows:

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

3. Terminal diperbarui dengan status instance Docker Anda saat proses selesai.

Instance containerized ini LocalStack memberi Anda akses ke AWS layanan yang Anda tentukan selama proses pengunduhan.

## Mengkonfigurasi CLI LocalStack untuk dan Docker.

Untuk mengonfigurasi AWS CLI dan AWS Toolkit agar berfungsi LocalStack di Docker, siapkan profil baru dengan menyelesaikan langkah-langkah berikut:

1. Dari VS Code, buka terminal VS Code dengan menekan **ctrl + ` (backtick)**.
2. Masukkan yang berikut ini ke terminal.

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

3. AWS Toolkit mendeteksi LocalStack profil Anda dan memperbarui menu status koneksi.

Setelah penyiapan, memilih LocalStack profil Anda dari bagian AWS profil bilah status membuat LocalStack sumber daya Anda terlihat di AWS penjelajah. Selain itu, Anda dapat melihat LocalStack log Anda di tab Output dari terminal VS Code.

## Mulai LocalStack dari VS Code

Anda dapat mulai LocalStack menggunakan salah satu metode berikut:

### Mulai LocalStack dari VS Code Status Bar

1. Dari VS Code, navigasikan ke bilah status, lalu pilih LocalStack tombol Mulai untuk diluncurkan LocalStack.
2. Bilah Status Kode VS diperbarui ketika LocalStack telah berhasil diluncurkan.

### Mulai LocalStack dari Palet Perintah Kode VS

1. Dari VS Code, buka Command Palette dengan menekan **Cmd + Shift + P** (Mac) atau **Control + Shift + P** (Windows).
2. Dari Command Palette, masukkan **Start LocalStack** di bilah pencarian dan pilih dari daftar saat diisi dalam hasil.

3. Bilah Status Kode VS diperbarui ketika LocalStack telah berhasil diluncurkan.

### Mulai LocalStack dari terminal VS Code

1. Dari VS Code, buka terminal VS Code dengan menekan **ctrl + ` (backtick)**.
2. Dari terminal VS Code, masukkan perintah **localstack start CLI**.
3. Bilah Status Kode VS diperbarui ketika LocalStack telah berhasil diluncurkan.

### Membangun contoh aplikasi tanpa server

Untuk mulai bekerja dengan LocalStack di VS Code, Anda memerlukan contoh aplikasi tanpa server. Jika Anda sudah memiliki aplikasi yang ada di AWS akun Anda, Anda dapat menerapkannya secara lokal menggunakan LocalStack atau Anda dapat membuat aplikasi baru dengan AWS Serverless Land.

Untuk informasi tambahan tentang membuat aplikasi dengan Tanah Tanpa Server di AWS Toolkit, lihat topik [Bekerja dengan Tanah AWS Tanpa Server](#) di Panduan Pengguna ini. Untuk informasi rinci tentang Serverless Land, lihat halaman landing utama aplikasi web [Serverless Land](#).

### Menguji dan men-debug fungsi Lambda dengan LocalStack

Menguji dan men-debug fungsi Lambda Anda LocalStack di ekstensi VS Code mirip dengan bekerja dengan fungsi Anda yang diterapkan ke cloud. AWS Perbedaan utamanya adalah instans AWS Toolkit Anda harus diautentikasi dengan LocalStack akun Anda untuk menyebarkan dan men-debug fungsi Anda. LocalStack

#### Note

Fitur pengujian dan debugging yang dijelaskan dalam bagian ini tidak tersedia untuk edisi LocalStack Komunitas.

Untuk bekerja dengan LocalStack di VS Code, sambungkan ke LocalStack profil Anda di AWS Toolkit. Saat LocalStack profil Anda aktif, bilah status VS Code menampilkan AWS: profile:localstack (titik akhir khusus) dengan tanda centang.

Untuk informasi rinci tentang bekerja dengan fungsi Lambda Anda di AWS Toolkit, lihat topik [Bekerja dengan AWS Lambda fungsi](#) dalam panduan pengguna ini.

## AWS Lambda debugging jarak jauh

AWS Toolkit for Visual Studio Code Ini memungkinkan Anda untuk men-debug AWS Lambda fungsi Anda yang berjalan di cloud, langsung di VS Code. Dengan debugging AWS Lambda jarak jauh, Anda dapat memeriksa fungsi yang sedang berjalan, mengatur breakpoint, memeriksa variabel, dan debugging step-through tanpa memodifikasi alur kerja pengembangan yang ada.

Bagian berikut menjelaskan cara bekerja dengan debugging jarak jauh Lambda di file. AWS Toolkit for Visual Studio Code

### Cara kerja debugging jarak jauh Lambda

AWS Toolkit memungkinkan debugging jarak jauh dengan memodifikasi sementara fungsi Lambda Anda dengan lapisan debugging Lambda tambahan dan memperpanjang batas waktu pemanggilan Lambda hingga 900 detik. Koneksi aman dibuat antara debugger lokal Anda dan lingkungan runtime Lambda menggunakan Tunneling Aman. AWS IoT Koneksi ini memungkinkan Anda untuk menggunakan breakpoint kode lokal Anda untuk melangkah melalui fungsi saat dijalankan dari jarak jauh. Setelah sesi debugging Anda selesai, semua modifikasi sementara secara otomatis dikembalikan ke pengaturan aslinya.

### Memulai

Waktu aktif yang didukung

Runtime berikut didukung oleh debugging jarak jauh Lambda.

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

#### Note

Instans terkelola Lambda dan tipe fungsi gambar OCI tidak didukung oleh debugging jarak jauh Lambda.

### Prasyarat

Sebelum Anda mulai, prasyarat berikut harus dipenuhi.

- Anda harus memiliki AWS kredensial yang valid yang dikonfigurasi di Toolkit. AWS Untuk detail tambahan tentang menginstal AWS Toolkit dan mengonfigurasi kredensial Anda, lihat topik [Memulai](#) di panduan pengguna ini.
- Fungsi Lambda telah diterapkan ke akun Anda. AWS Untuk detail tentang penerapan fungsi Lambda, lihat topik [Membuat fungsi Lambda pertama](#) Anda di Panduan Pengembang. AWS Lambda
- Anda harus memiliki kebijakan dan izin AWS Identity and Access Management (IAM) yang sesuai untuk men-debug fungsi Anda. Untuk detail tambahan tentang izin Lambda, lihat [kebijakan AWS terkelola untuk AWS Lambda](#) topik di Panduan Pengembang. AWS Lambda Berikut ini adalah contoh kebijakan yang berisi izin minimum yang diperlukan untuk bekerja dengan debugging jarak jauh Lambda di Toolkit. AWS

#### Note

Debugging jarak jauh diaktifkan melalui Tunneling AWS AWS IoT Aman. Ini memungkinkan debugger lokal Anda untuk membuat koneksi aman ke lingkungan runtime Lambda.

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

```
}
```

## Mengakses debugging jarak jauh Lambda

Ada dua jalur utama untuk mengakses Lambda remote debugging di AWS Toolkit: AWS explorer atau Application Builder explorer. Dari AWS penjelajah, Anda dapat mengakses debugging jarak jauh Lambda melalui node Anda. AWS Lambda Dari penjelajah Application Builder, Anda dapat mengakses debugging jarak jauh Lambda melalui proyek lokal Anda. AWS SAM

### Mengakses debugging jarak jauh Lambda dari penjelajah AWS

1. Dari VS Code, buka ekstensi AWS Toolkit.
2. Dari AWS Toolkit, perluas AWS penjelajah.
3. Dari penjelajah, perluas simpul Lambda.
4. Arahkan ke fungsi yang ingin Anda debug, lalu pilih ikon Panggil jarak jauh dari menu konteks untuk membuka layar konfigurasi pemanggilan jarak jauh.

### Mengakses debugging jarak jauh Lambda dari penjelajah Application Builder.

1. Dari VS Code, buka ekstensi AWS Toolkit.
2. Dari AWS Toolkit, perluas explorer pembuat aplikasi.
3. Dari penjelajah, perluas AWS SAM proyek yang berisi proyek Lambda yang ingin Anda debug.
4. Perluas Lambda fungsi yang digunakan yang ingin Anda debug.
5. Arahkan ke fungsi remote, lalu pilih ikon Invoke remote dari menu konteks untuk membuka layar konfigurasi pemanggilan jarak jauh.

## Bekerja dengan debugging jarak jauh Lambda

Bagian berikut menjelaskan cara bekerja dengan debugging jarak jauh Lambda di file. AWS Toolkit for Visual Studio Code

**Note**

Fungsi Lambda memiliki batas 5 lapisan dan batas gabungan 250MB untuk kode fungsi dan semua lapisan terlampir. Debugging jarak jauh Lambda membutuhkan setidaknya 1 lapisan gratis untuk dijalankan.

## Menyiapkan sesi debugging

Sebelum memulai, konfigurasi sesi debugging Anda dengan menyelesaikan prosedur berikut.

1. Buka menu konfigurasi panggilan Jarak Jauh dengan menyelesaikan proses debug jarak jauh Accessing Lambda dari explorer AWS atau debugging jarak jauh Accessing Lambda dari prosedur explorer Application Builder, yang terletak di bagian sebelumnya.
2. Dari menu konfigurasi panggilan jarak jauh, pilih kotak centang Remote Debugging untuk menampilkan properti debugging jarak jauh.
3. Tentukan Local Root Path ke file handler lokal Anda.

**Note**

Jalur root lokal adalah lokasi kode sumber Anda yang cocok dengan fungsi Lambda yang diterapkan. Jika Anda bekerja dari fungsi yang di-deploy di Application Builder explorer, jalur root lokal Anda akan terdeteksi secara otomatis. Jika Anda tidak memiliki kode sumber yang disimpan secara lokal, pilih tombol Unduh kode jarak jauh untuk mengambil kode sumber fungsi Lambda Anda. Ini akan membuka Anda `handler file` di editor VS Code.

4. Dari bagian Payload, tentukan di mana data ujian-acara Anda diperoleh.

## Mengatur breakpoint dan debugging

Atur breakpoint dan mulai debugging dengan menyelesaikan prosedur berikut.

1. Dari editor VS Code `handler file`, klik margin selokan untuk mengatur breakpoint pada nomor baris tempat Anda ingin menjeda debugging.
2. Ketika Anda puas dengan breakpoint, kembali ke menu konfigurasi pemanggilan jarak jauh untuk memverifikasi bahwa pengaturan Anda dikonfigurasi dengan benar, lalu pilih tombol Remote invoke untuk memulai debugging.

3. AWS Toolkit memperbarui fungsi Lambda Anda dengan kemampuan debugging, membuat terowongan aman untuk sesi debugging, memanggil fungsi Anda dengan muatan yang ditentukan, lalu menjeda proses saat mencapai breakpoint.
4. Pada jeda breakpoint, gunakan panel RUN AND DEBUG untuk melihat VARIABLES, CALL STACK, dan BREAKPOINTS Anda.

### Memperbarui dan menguji fungsi Anda

Untuk memodifikasi kode Anda dan menguji perubahan dengan penerapan cepat, selesaikan prosedur berikut.

1. Dengan sesi debugging Anda aktif, buat perubahan handler file pada editor VS Code Anda.
2. Simpan perubahan Anda (**Command+S on macOS, Ctrl+S on Windows**)
3. Saat diminta, konfirmasi bahwa Anda ingin melanjutkan untuk menerapkan perubahan Anda. AWS Toolkit akan memperbarui fungsi Lambda Anda dengan kode yang dimodifikasi.
4. Lanjutkan debugging dan uji perubahan Anda dengan mengatur breakpoint baru dan memilih tombol Remote invoke lagi.

#### Note

Atau, Anda dapat membatalkan pilihan opsi Attach debugger di kontrol debugging VS Code dan memilih tombol Remote invoke untuk menjalankan fungsi Anda tanpa debugging.

### Mengakhiri sesi debugging

Masing-masing opsi berikut mengakhiri sesi debugging jarak jauh Anda dan menghapus lapisan debug dari proyek Anda.

- Memilih opsi Hapus Pengaturan Debug dari layar konfigurasi pemanggilan jarak jauh.
- Memilih ikon putus sambungan dari kontrol debugging VS Code.
- Menutup handler file di editor Kode VS.

#### Note

Perhatikan hal-hal berikut ini:

- Lapisan debug Lambda secara otomatis dihapus setelah 60 detik tidak aktif. Hitungan dimulai saat pemanggilan terakhir Anda selesai.
- Jika Anda membuat perubahan kode pada fungsi infrastructure-as-code (IaC) yang dikelola (AWS SAM, AWS CDK, Terraform) selama proses debugging, simpan ke proyek lokal Anda dan pertimbangkan untuk memperbarui repositori kontrol sumber Anda. Perubahan yang belum disimpan akan ditimpa saat fungsi IaC Anda di-deploy ulang.
- Jika Anda membuat perubahan sementara hanya untuk tujuan debugging, Anda mungkin ingin menerapkan ulang fungsi Anda dari kontrol sumber untuk memastikannya cocok dengan kode produksi Anda.

## Mendebug fungsi TypeScript Lambda dengan peta sumber

Bagian berikut menjelaskan cara men-debug fungsi TypeScript Lambda Anda dengan peta sumber.

### Prasyarat


Untuk men-debug fungsi TypeScript Lambda Anda, prasyarat berikut harus dipenuhi.

- Anda TypeScript harus dikompilasi dengan opsi peta sumber diaktifkan. Untuk informasi tambahan, lihat topik [dukungan peta JavaScript sumber](#) dalam dokumentasi VS Code.
- Peta sumber in-line tidak didukung. Anda harus menggunakan `.js.map` file terpisah untuk menyimpan peta sumber.

### Konfigurasi

Untuk mengonfigurasi debugging jarak jauh Lambda untuk fungsi TypeScript Lambda di AWS Toolkit, selesaikan langkah-langkah berikut.

1. Dari AWS Toolkit, perluas AWS penjelajah.
2. Dari penjelajah, perluas simpul Lambda.
3. Arahkan ke fungsi yang ingin Anda konfigurasi TypeScript, lalu pilih ikon Panggil jarak jauh dari menu konteks untuk membuka layar konfigurasi pemanggilan jarak jauh.
4. Aktifkan debugging jarak jauh dengan memilih kotak centang Remote debugging.
5. Konfigurasi Jalur Root Lokal Anda dengan menunjuk ke direktori yang berisi `AndaTypeScript handler file`.


 Note

Di TypeScript handler file sinilah Anda mengatur breakpoint debugging Anda.

6. Perluas pengaturan konfigurasi tambahan debug jarak jauh.
7. Aktifkan pemetaan sumber dengan memilih kotak centang Peta Sumber.
8. Atur bidang Keluar file ke direktori lokal salinan fungsi Lambda Anda.

## Example

Jika `app.js` dan `app.map` berada di `.aws-sam/build/HelloWorldFunction`, maka buatlah lokasi file Keluar/Users/**user/project**/aws-sam/build/**HelloWorldFunction**/\*.

 Note

Jalur file Out harus menjadi jalur absolut.

Untuk AWS SAM dan AWS CDK proyek, AWS Toolkit mendukung deteksi peta sumber otomatis. Jika bidang Out files dibiarkan kosong untuk proyek-proyek ini, toolkit akan secara otomatis mencoba mendeteksi lokasi peta sumber.

9. Ketika Anda puas dengan pengaturan, pilih tombol Remote invoke untuk mulai men-debug fungsi Anda TypeScript .

## Pemecahan masalah dan kasus penggunaan lanjutan

Jika sesi debug Anda gagal, mulailah proses pemecahan masalah dengan menyelesaikan langkah-langkah ini.

1. Perbarui AWS Toolkit ke versi terbaru.
2. Segarkan tampilan web dengan menutup tampilan web konfigurasi panggilan Remote dan membukanya kembali.
3. Mulai ulang VS Code dengan menutupnya sepenuhnya dan membukanya kembali.
4. Buka Palet Perintah Kode VS dan masukkan perintah **AWS: Reset Lambda Remote Debugging Snapshot**, pilih saat diisi dalam hasil untuk mengatur ulang snapshot debugging jarak jauh Lambda Anda.

## 5. [Jika Anda tidak dapat memecahkan masalah, kirimkan masalah ke AWS Toolkit for Visual Studio Code GitHub Masalah.](#)

Kasus penggunaan lanjutan: konfigurasi penandatanganan kode

Debugging jarak jauh membutuhkan melampirkan lapisan debug ke fungsi Lambda Anda. Jika fungsi Anda memiliki konfigurasi penandatanganan kode diaktifkan dan diberlakukan, AWS Toolkit tidak dapat secara otomatis melampirkan lapisan debug ke fungsi Anda.

Ada dua opsi untuk mengatasi masalah konfigurasi penandatanganan kode.

- Hapus penandatanganan kode untuk sementara.
- Gunakan layer debug yang ditandatangani.

Menghapus penandatanganan kode untuk sementara

Perbarui konfigurasi penandatanganan kode dengan menyetel `UntrustedArtifactOnDeployment` : Warn, lalu aktifkan kembali Enforced setelah proses debugging selesai.

Untuk informasi selengkapnya, lihat [UpdateCodeSigningConfig](#) referensi di Referensi AWS Lambda API.

Menggunakan lapisan debug yang ditandatangani

1. Dari debugging jarak jauh Lambda di AWS Toolkit, perluas bagian konfigurasi tambahan debug jarak jauh.
2. Dari bagian konfigurasi tambahan debug jarak jauh, salin ARN layer Region Anda dari bidang penggantian Layer.
3. Dari AWS CLI, gunakan perintah berikut untuk men-download versi layeraws `lambda get-layer-version-by-arn --arn layer-arn`, menggantikan layer-arn dengan layer ARN Anda. Untuk petunjuk rinci tentang cara mengunduh layer debug yang ditandatangani, lihat referensi [get-layer-version-by-arn di Command Reference](#).AWS CLI
4. Tanda tangani layer dengan konfigurasi penandatanganan kode Anda dan publikasikan ke akun Anda. Untuk panduan penandatanganan dan penerbitan, lihat [Menyiapkan penandatanganan kode untuk topik AWS SAM aplikasi Anda](#) di Panduan AWS Serverless Application Model Pengembang.

5. Setelah layer ditandatangani dan dipublikasikan ke akun Anda, kembali ke bagian konfigurasi tambahan debug jarak jauh Lambda dari debugging jarak jauh Lambda, lalu masukkan ARN layer baru ke bidang penggantian Layer. Ketika proses selesai, debugging jarak jauh Lambda menggunakan lapisan yang ditandatangani Anda alih-alih lapisan default.

Kasus penggunaan lanjutan: fungsi debugging dengan SnapStart atau konkurensi yang disediakan

Untuk fungsi Lambda yang dikonfigurasi dengan SnapStart atau disediakan konkurensi, penerbitan versi baru membutuhkan waktu yang jauh lebih lama. Untuk mempercepat alur kerja debugging, Anda dapat mengonfigurasi debugging jarak jauh Lambda untuk memperbarui hanya \$LATEST versi fungsi Anda alih-alih menerbitkan versi baru.

1. Dari layar konfigurasi pemanggilan jarak jauh, perluas pengaturan konfigurasi tambahan Remote debug.
2. Hapus pilihan versi Publikasikan.
3. AWS Toolkit sekarang hanya akan memperbarui \$LATEST versi fungsi Anda dan men-debug menggunakannya.

#### Note

Sebagai efek samping dari debugging dengan \$LATEST versi, Anda harus menghindari lalu lintas lain yang mungkin memanggil \$LATEST versi Anda untuk memastikan lingkungan debugging yang tidak terganggu.

### Wilayah yang didukung

Kesalahan berikut terjadi ketika suatu wilayah tidak mendukung debugging jarak jauh.

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

Berikut ini adalah daftar wilayah yang didukung.

- ap-east-1
- ap-northeast-1
- ap-northeast-2

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

### Lambda RequestEntityTooLargeException

Fungsi Lambda memiliki batas 5 lapisan dan batas gabungan 250MB untuk kode fungsi dan semua lapisan terlampir. Lapisan debugging jarak jauh sekitar 40MB, yang dapat menyebabkan fungsi Anda melebihi batas ini jika Anda memiliki paket fungsi besar atau beberapa lapisan. Untuk detail tambahan, lihat bagian [Lambda: InvalidParameterValueException atau RequestEntityTooLargeException](#) topik di Panduan AWS Lambda Pengembang.

Daftar berikut menjelaskan cara untuk memecahkan masalah dan memperbaiki kesalahan ini.

- Kurangi ukuran fungsi: Optimalkan kode fungsi Anda dan hapus dependensi yang tidak perlu.
- Hapus lapisan yang tidak digunakan: Hapus sementara lapisan yang tidak penting selama debugging.
- Gunakan dependensi eksternal: Pindahkan dependensi besar ke penyimpanan eksternal, seperti Amazon S3, dan muat saat runtime.

## Memecahkan masalah debugging Java

Untuk men-debug fungsi Java Lambda, Anda harus menginstal versi Java yang sama secara lokal yang cocok dengan versi runtime fungsi Lambda Anda.

Misalnya, saat men-debug fungsi Java 25, Anda harus menginstal Java 25 di lingkungan lokal tempat AWS Toolkit berjalan. Jika Anda mencoba men-debug fungsi Java 25 dengan Java 21 atau versi sebelumnya yang diinstal secara lokal, debugging jarak jauh tidak akan dapat berhenti di breakpoint yang Anda tetapkan.

Pastikan versi Java lokal Anda cocok dengan versi runtime fungsi Lambda Anda sebelum memulai sesi debugging.

### Kuota tunneling aman IoT terlampaui

Berikut ini adalah contoh kesalahan terlampaui kuota terowongan yang terjadi ketika Anda telah mencapai batas harian untuk koneksi tunneling AWS IoT aman di debugging jarak jauh Lambda.

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

AWS IoT Koneksi Tunneling yang aman memiliki kuota berikut:

- Tunneling aman IoT tingkat bebas dialokasikan 10 koneksi per hari.
- Setiap terowongan mendukung satu instance VS Code hingga 12 jam.
- Kuota berlaku per AWS akun, per hari.

Jika Anda mengalami kesalahan tunneling AWS IoT aman, tunggu pengaturan ulang kuota harian atau hubungi AWS dukungan untuk meminta peningkatan batas kuota. Untuk info kontak AWS dukungan, lihat [portal kontak AWS dukungan](#). Untuk informasi rinci tentang tunneling AWS IoT aman, lihat topik [tunneling AWS IoT aman](#) di Panduan Pengembang.AWS IoT

## 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](#).

### Memulai

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/>
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/>
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
- Kredensial sementara

Untuk menyambung ke database yang terletak di klaster 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 AWS Region 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

**ℹ 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 AWS Region 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 AWS Region 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 AWS Region 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.

# Amazon SageMaker Unified Studio untuk VS Code

Sebagai bagian dari generasi Amazon berikutnya SageMaker, Amazon SageMaker Unified Studio adalah pengalaman pengembangan terpadu yang menyatukan AWS data, analitik, kecerdasan buatan (AI), dan layanan pembelajaran mesin (ML). Ini menyediakan tempat untuk membangun, menyebarkan, mengeksekusi, dan memantau alur kerja dari satu antarmuka. Untuk informasi selengkapnya tentang menyiapkan integrasi Amazon SageMaker Unified Studio dengan VS Code IDE, lihat [Menyiapkan integrasi Amazon SageMaker Unified Studio dalam Kode VS di Panduan Pengguna Amazon SageMaker Unified Studio](#).

## Bekerja dengan aplikasi tanpa server

AWS Toolkit for Visual Studio Code Memberikan dukungan untuk [AWS Serverless Application](#). Topik berikut menjelaskan cara 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](#)
- [Bekerja dengan AWS Serverless Land](#)
- [Menjalankan dan men-debug fungsi Lambda langsung dari kode](#)
- [Menjalankan dan men-debug sumber daya Amazon API Gateway lokal](#)
- [Opsi konfigurasi untuk men-debug aplikasi nirserver](#)
- [Memecahkan masalah aplikasi tanpa server](#)

## 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 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 mengonfigurasi 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 read/write akses yang sesuai ke layanan berikut: CloudFormation, IAM, Lambda, Amazon API Gateway, Amazon Simple Storage Service (Amazon S3) Simple Storage Service (Amazon S3), dan Amazon Elastic Container Registry (Amazon ECR).

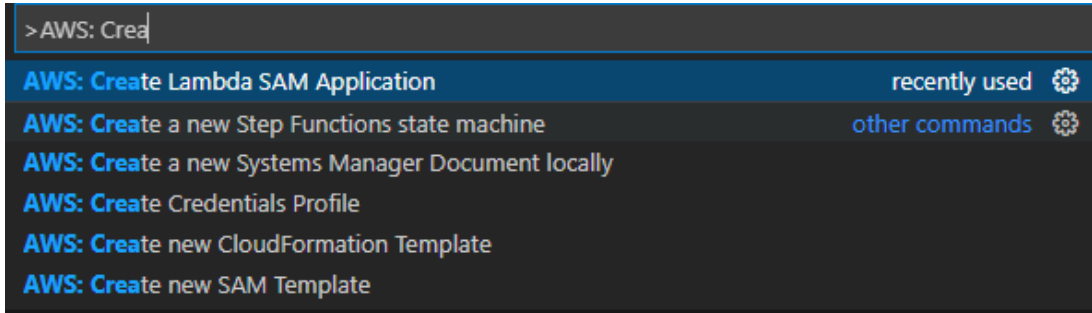
Untuk informasi tambahan tentang menyiapkan autentikasi yang diperlukan untuk menyebarkan dan menjalankan aplikasi tanpa server, lihat [Mengelola akses dan izin sumber daya](#) di Panduan Pengembang. AWS Serverless Application Model Untuk informasi tentang cara mengatur kredensial Anda, lihat [AWS Kredensi IAM](#) 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 AWS Toolkit 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 ARM64 berbasis berbasis 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 ARM64 berbasis. 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 folder tersebut. 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 langsung dari kode](#).

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. Menggunakan CodeLens indikator dalam `template.yaml` file untuk menambahkan 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`.

Mengkonfigurasi `launch.json` file untuk debugging berbasis template.

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.

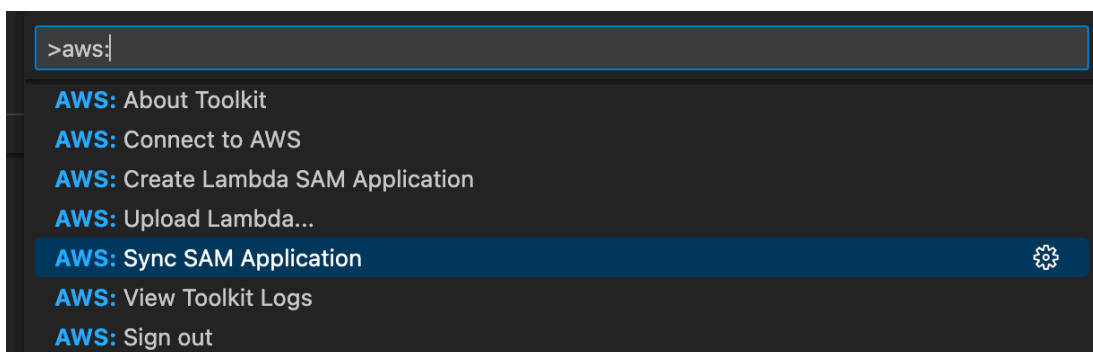
Ketika sesi debug dimulai, panel KONSOL DEBUG menampilkan output debug dan menampilkan nilai apa pun yang dikembalikan oleh fungsi Lambda. (Saat men-debug AWS SAM aplikasi, AWS Toolkit dipilih sebagai saluran Output di panel Output.)

## 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 CloudUntuk 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.


- 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.


- 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.
- 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.

- 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 AWS Explorer. Untuk informasi selengkapnya tentang cara menjalankan fungsi Lambda yang dibuat sebagai bagian dari aplikasi, lihat topik [Bekerja dengan AWS Lambda Fungsi](#) di Panduan Pengguna ini.

## Menghapus aplikasi tanpa server dari AWS Cloud

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

1. Buka [AWS Penjelajah](#).
2. Di jendela AWS Toolkit Explorer, perluas Wilayah yang berisi aplikasi yang digunakan yang ingin Anda hapus, lalu perluas CloudFormation
3. Buka menu konteks (klik kanan) untuk nama CloudFormation tumpukan yang sesuai dengan aplikasi tanpa server yang ingin Anda hapus, lalu pilih Hapus Tumpukan 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 CloudFormation AWS

## Bekerja dengan AWS Serverless Land

AWS Serverless Land in the AWS Toolkit for Visual Studio Code adalah kumpulan fitur yang membantu Anda membangun arsitektur berbasis acara. Bagian topik berikut menjelaskan cara bekerja dengan Serverless Land di Toolkit. AWS Untuk informasi rinci tentang Serverless Land, lihat aplikasi web [Serverless Land](#).

## Mengakses Tanah Tanpa Server

Ada 3 titik masuk utama untuk mengakses Serverless Land di Toolkit: AWS

- Palet Perintah Kode VS

- AWS Penjelajah Toolkit
- AWS Penjelajah Pembuat Aplikasi Toolkit

### Membuka Tanah Tanpa Server dari Palet Perintah Kode VS

Untuk membuka Serverless Land dari VS Code Command Palette, selesaikan langkah-langkah berikut.

1. Dari VS Code, buka Command Palette dengan menekan **option+shift+p** (Mac) atau **control+shift+p** (Windows).
2. Dari Palet Perintah Kode VS, **AWS Create application with Serverless template** masuk ke bilah pencarian.
3. Pilih AWS: Buat aplikasi dengan template Tanpa Server saat terisi dalam daftar.
4. Wizard Land Tanpa Server terbuka ke layar Pilih Pola untuk aplikasi Anda (1/5) di VS Code saat proses selesai.

### Membuka Tanah Tanpa Server dari AWS Toolkit Explorer.

Untuk membuka Serverless Land dari AWS Toolkit Explorer, selesaikan langkah-langkah berikut.

1. Dari AWS Toolkit Explorer, perluas wilayah tempat Anda ingin membuka Serverless Land.
2. Buka menu konteks untuk (klik kanan) simpul Lambda.
3. Pilih Buat aplikasi dengan template Tanpa Server dari menu konteks.
4. Wizard Land Tanpa Server terbuka ke layar Pilih Pola untuk aplikasi Anda (1/5) di VS Code saat proses selesai.

### Membuka Tanah Tanpa Server dari penjelajah Application Builder

Untuk membuka Serverless Land dari penjelajah AWS Toolkit Application Builder, selesaikan langkah-langkah berikut.

1. Dari AWS Toolkit Explorer, arahkan ke explorer Application Builder.
2. Klik kanan Application Builder explorer dan pilih Create application with Serverless template dari menu konteks.
3. Wizard Land Tanpa Server terbuka ke layar Pilih Pola untuk aplikasi Anda (1/5) di VS Code saat proses selesai.

## Membuat aplikasi dengan template Serverless

Untuk membuat aplikasi dengan template Tanpa Server, selesaikan langkah-langkah berikut.

1. Dari layar Serverless Land wizard Pilih Pola untuk aplikasi Anda (1/5), pilih Pola untuk dasar aplikasi Anda.

### Note

Untuk melihat pratinjau dan detail lebih lanjut tentang Pola tertentu, pilih ikon Buka di Tanah Tanpa Server yang terletak di sebelah Pola yang ingin Anda lihat. Pola Tanah Tanpa Server terbuka di browser web default Anda.

2. Dari layar Select Runtime (2/5), pilih runtime untuk proyek Anda.
3. Dari layar Pilih IAc (3/5), pilih opsi IAc untuk proyek Anda.
4. Dari layar Pilih lokasi proyek (4/5), pilih lokasi untuk menyimpan proyek Anda.
5. Dari layar Enter Project Name (5/5), masukkan nama untuk aplikasi baru Anda.
6. Aplikasi baru Anda ditampilkan di penjelajah VS Code dan proyek Anda `readme.md` terbuka di editor VS Code, ketika prosedur selesai.

### Note

Setelah aplikasi baru Anda dibuat, tindakan tambahan yang spesifik untuk jenis aplikasi Anda dapat ditemukan di `readme.md` file. Selain itu, aplikasi AWS Serverless Application Model (AWS SAM) Anda dapat dibuka dengan AWS Application Builder untuk pengujian lokal, debugging, dan banyak lagi.

Untuk detail tentang bekerja dengan Application Builder di AWS Toolkit, lihat topik [Working with the AWS Application Builder explorer](#) di Panduan Pengguna ini.

## Menjalankan dan men-debug fungsi Lambda langsung dari kode

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; .NET 5.0)	<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> <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>

**Bahasa/runtime**

Kriteria fungsi Lambda harus diidentifikasi oleh indikator CodeLens

Ekstensi [ms-python.python](#) (atau ekstensi apa pun yang menyediakan simbol bahasa untuk Python) diinstal dan diaktifkan.

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> dan A 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`  
Akses opsi Tambahkan Konfigurasi Debug di CodeLens indikator untuk penanganan fungsi Lambda.
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": Pasangan nilai kunci berformat JSON yang menentukan muatan peristiwa.
  - "path": Jalur ke file yang digunakan sebagai muatan peristiwa.

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

Mengkonfigurasi `launch.json` file untuk langsung menjalankan fungsi Lambda.

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 sebelah RUN.

Saat sesi debugging dimulai, panel DEBUG CONSOLE 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 sumber daya lokal AWS SAM API Gateway, yang ditentukan dalam `template.yaml`, dengan menjalankan konfigurasi peluncuran VS Code `type=aws-sam` dengan file `invokeTarget.target=api`

### Note

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

## 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 handler (.js, .cs, atau file.py) di AWS SAM proyek Anda, arahkan kursor ke handler Lambda, dan pilih Tambahkan Konfigurasi Debug. CodeLens Kemudian, di menu, pilih item yang ditandai API Event.
  - Opsi 2: Edit `launch.json` dan 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. Di panel VS Code Run, pilih konfigurasi peluncuran (dinamai `myConfig` dalam contoh di atas).
3. (Opsional) Tambahkan titik henti ke kode proyek Lambda Anda.
4. Ketik F5 atau pilih Play di panel Run.
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 AWS koneksi disediakan. Untuk informasi selengkapnya, lihat tabel properti AWS koneksi ("aws") di [Ops konfigurasi untuk men-debug aplikasi nirserver](#) bagian.

## sam

Bagaimana AWS SAM CLI membangun aplikasi. Untuk informasi lebih lanjut, lihat tabel properti AWS SAM CLI ("sam") di bagian ini [Ops konfigurasi untuk men-debug aplikasi nirserver](#).

## Opsi konfigurasi untuk men-debug aplikasi nirserver

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

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

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

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

Properti	Deskripsi
type	Menentukan ekstensi mana yang mengelola konfigurasi peluncuran. Selalu atur <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 jenis 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 di mana aplikasi SAM Lambda lokal Anda berjalan. Untuk runtime tertentu, Anda dapat memilih <code>arm64</code> alih-alih <code>x86_64</code> arsitektur default. Untuk informasi selengkapnya, lihat <a href="#">Membuat aplikasi tanpa server baru (lokal)</a>.</li> </ul> <p>Untuk menjalankan resource Lambda dengan AWS SAM template, tetapkan nilai untuk bidang berikut <code>invokeTarget</code> :</p> <ul style="list-style-type: none"> <li>• <code>target</code> – Atur ke template.</li> <li>• <code>templatePath</code> — Jalur ke file AWS SAM template.</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 template yang diformat YAML AWS SAM . <a href="#">Perhatikan bahwa secara AWS Toolkit implisit mengenali fungsi yang didefinisikan <code>PackageType: Image</code> dalam AWS SAM template sebagai fungsi Lambda berbasis Gambar</a>. 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 menulis ke bucket Amazon S3, alih-alih mengkode keras 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 AWS SAM template (<code>template.yaml</code>) dan file <code>launch.json</code></p> <p>Contoh pemformatan untuk variabel lingkungan dalam AWS SAM template:</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 dalam <code>launch.json</code> file:</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 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
<code>pathMappings</code>	<p>Menentukan di mana kode lokal dalam kaitannya dengan di mana ia berjalan dalam wadah.</p> <p>Secara default, Toolkit for VS Code <code>localRoot</code> disetel ke root kode fungsi Lambda di ruang kerja lokal, <code>remoteRoot</code> dan <code>/var/task</code> to, yang merupakan direktori kerja default untuk kode yang berjalan di Lambda. Jika direktori kerja diubah di Dockerfile atau dengan <code>WorkingDirectory</code> parameter dalam file CloudFormation template, setidaknya satu <code>pathMapping</code> entri harus ditentukan sehingga debugger dapat berhasil memetakan breakpoint yang disetel secara lokal ke kode yang berjalan di wadah Lambda.</p> <p>Contoh pemformatan untuk <code>pathMappings</code> dalam <code>launch.json</code> file:</p> <pre>"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 satu entri pemetaan jalur.</li></ul>

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

AWS SAM Properti 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 wadah 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 template Anda menggunakan parameter untuk memasukkan nilai pelanggan. Untuk	<code>"parameters": {}</code>

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

### AWS properti koneksi ("**aws**")

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

### Contoh: Konfigurasi peluncuran template

Berikut adalah contoh peluncuran file konfigurasi untuk target AWS SAM template:

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

## Memecahkan masalah aplikasi tanpa server

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

### Topik

- [Bagaimana saya bisa menggunakan samconfig.toml dengan konfigurasi peluncuran SAM?](#)
- [Kesalahan: "RuntimeError: Wadah tidak ada"](#)
- [Kesalahan: "docker.errors.APIError: 500 Server Kesalahan... Anda telah mencapai batas tingkat tarik Anda."](#)
- [Kesalahan: "500 Server Error: Mounting C:\Users\..."](#)

- [Menggunakan WSL, tampilan web \(misalnya, formulir “Invoke on AWS”\) rusak](#)
- [Mendebug TypeScript aplikasi, tetapi breakpoint tidak berfungsi](#)

Bagaimana saya bisa menggunakan `samconfig.toml` dengan konfigurasi peluncuran SAM?

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

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

**Kesalahan: "RuntimeError: Wadah tidak ada"**

`sam buildPerintah` dapat menunjukkan kesalahan ini jika sistem Anda tidak memiliki cukup ruang disk untuk wadah 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 build dimulai. Untuk informasi lebih lanjut, lihat [GitHub masalah ini](#).

**Kesalahan: “docker.errors.APIError: 500 Server Kesalahan... Anda telah mencapai batas tingkat tarik Anda.”**

Docker Hub membatasi permintaan yang dapat dibuat oleh pengguna anonim. Jika sistem Anda mencapai batas, Docker gagal dan kesalahan ini muncul di 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 layanan Docker sistem Anda telah diautentikasi dengan kredensial Docker Hub Anda.

## Kesalahan: “500 Server Error: Mounting C:\Users\...”

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

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

Coba segrakan kredensial untuk drive bersama Anda (di pengaturan Docker).

Menggunakan WSL, tampilan web (misalnya, formulir “Invoke on AWS”) rusak

Ini adalah masalah VS Code yang diketahui untuk pengguna Cisco VPN. Untuk informasi lebih lanjut, lihat [GitHubmasalah ini](#).

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

### Mendebug TypeScript aplikasi, tetapi breakpoint tidak berfungsi

Ini akan terjadi jika tidak ada peta sumber untuk menautkan JavaScript file yang dikompilasi ke TypeScript file sumber. Untuk memperbaikinya, buka `tsconfig.json` file Anda dan pastikan opsi dan nilai berikut ditetapkan: `"inlineSourceMap": true`.

## Bekerja dengan dokumen Otomasi Systems Manager

AWS Systems Manager memberi Anda visibilitas dan kontrol infrastruktur Anda. AWS Systems Manager menyediakan antarmuka pengguna terpadu sehingga Anda dapat melihat data operasional dari beberapa AWS layanan dan mengotomatiskan tugas operasional di seluruh sumber daya Anda AWS .

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

## Topik

- [Asumsi dan prasyarat](#)
- [Izin IAM untuk dokumen Otomasi Systems Manager](#)
- [Membuat dokumen Systems Manager Automation baru](#)
- [Membuka dokumen Systems Manager Automation yang sudah ada](#)
- [Mengedit dokumen Otomasi Systems Manager](#)
- [Menerbitkan dokumen Otomasi Systems Manager](#)
- [Menghapus dokumen Otomasi Systems Manager](#)
- [Menjalankan dokumen Otomasi Systems Manager](#)
- [Memecahkan masalah dokumen Otomasi Systems Manager di Toolkit for VS Code](#)

## Asumsi dan prasyarat

Sebelum Anda mulai, pastikan:

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

## Izin IAM untuk dokumen Otomasi Systems Manager

Dalam Toolkit for VS Code Anda harus memiliki profil kredensial yang berisi AWS Identity and Access Management izin (IAM) 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:

JSON

```
{
```

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

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

## Membuat dokumen Systems Manager Automation baru

Anda dapat membuat dokumen Otomasi baru di JSON atau YAML menggunakan Visual Studio Code. 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.

Untuk membuat dokumen Otomasi baru

1. Buka Kode VS.
2. Pada menu View, pilih Command Palette untuk membuka Command Palette.
3. Di Command Palette, masukkan AWS Toolkit Create a new Systems Manager Document Locally.
4. Pilih salah satu template pemula 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 Systems Manager Automation yang sudah ada

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 Kode VS.
2. Dari navigasi sebelah kiri, pilih AWS untuk membuka Explorer. AWS
3. Di AWS Explorer, untuk Systems Manager, pilih ikon unduh pada dokumen yang ingin Anda buka lalu pilih versi dokumen. File akan terbuka dalam format untuk versi itu. Jika tidak, pilih Unduh sebagai JSON atau Unduh sebagai YAMAL.

**Note**

Menyimpan dokumen Otomasi secara lokal sebagai file di VS Code tidak membuatnya muncul di AWS. Itu perlu dipublikasikan AWS sebelum dieksekusi.

## Mengedit dokumen Otomasi Systems Manager

Jika Anda memiliki dokumen Otomasi apa pun, dokumen tersebut akan muncul dalam kategori dokumen Systems Manager yang dimiliki oleh saya di AWS Explorer. Anda dapat memiliki dokumen Otomasi yang sudah ada AWS, dan Anda dapat memiliki dokumen baru atau yang diperbarui yang sebelumnya Anda terbitkan AWS dari VS Code.

Saat Anda membuka dokumen Otomasi untuk diedit di VS Code, Anda dapat melakukan lebih banyak hal dengannya daripada yang Anda bisa di Konsol Manajemen AWS. Contoh:

- 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 pelengkapan otomatis pada berbagai opsi di JSON danYAML.

## Bekerja dengan versi

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

Untuk menyetel versi default

- Di AWS Explorer, arahkan ke dokumen yang ingin Anda atur versi default, buka menu konteks (klik kanan) untuk dokumen, dan pilih Setel versi default.

### Note

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

## Menerbitkan dokumen Otomasi Systems Manager

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

Untuk mempublikasikan dokumen Otomasi Anda

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

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

## Menghapus dokumen Otomasi 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 Otomasi yang telah dijalankan tidak menghapus AWS sumber daya yang dibuat atau dimodifikasi saat dimulai.

Untuk menghapus dokumen Otomasi

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

## Menjalankan dokumen Otomasi Systems Manager

Setelah dokumen Otomasi Anda diterbitkan AWS, Anda dapat menjalankannya untuk melakukan tugas atas nama Anda di AWS akun Anda. Untuk menjalankan dokumen Otomasi Anda Konsol Manajemen AWS, Anda menggunakan APIs, Systems Manager AWS CLI, atau Alat AWS untuk PowerShell. Untuk petunjuk tentang cara menjalankan dokumen Otomasi, lihat [Menjalankan otomatisasi sederhana](#) di Panduan AWS Systems Manager Pengguna.

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

**Note**

Menjalankan dokumen Otomasi dapat membuat sumber daya baru AWS dan dapat menimbulkan 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 Otomasi saya di VS Code, tetapi saya tidak melihatnya di Konsol Manajemen AWS.

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

Penerbitan dokumen Otomasi saya gagal dengan kesalahan izin.

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

Saya menerbitkan dokumen Otomasi saya ke AWS, tetapi saya tidak melihatnya di Konsol Manajemen AWS.

Pastikan Anda telah menerbitkan dokumen ke AWS Wilayah yang sama dengan yang Anda jelajahi Konsol Manajemen AWS.

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

Menghapus dokumen Otomasi tidak menghapus sumber daya yang dibuat atau dimodifikasi. Anda dapat mengidentifikasi AWS sumber daya yang telah dibuat dari [Konsol Manajemen AWS Penagihan](#), menjelajahi tagihan, dan memilih sumber daya yang akan dihapus dari sana.

## AWS Step Functions

Dengan AWS Step Functions, Anda dapat membuat alur kerja (juga disebut mesin status) untuk membangun aplikasi terdistribusi, mengotomatiskan proses, mengatur layanan mikro, dan membuat

pipeline data dan pembelajaran mesin. Topik-topik berikut menjelaskan cara bekerja dengan AWS Step Functions di AWS Toolkit for Visual Studio Code. Untuk informasi rinci tentang AWS Step Functions layanan, lihat Panduan [AWS Step Functions](#) Pengembang.

Topik

- [Bekerja dengan AWS Step Functions](#)
- [Bekerja dengan AWS Step Functions Workflow Studio](#)

## Bekerja dengan AWS Step Functions

Bagian berikut menjelaskan cara bekerja dengan AWS Step Functions Amazon State Language (ASL) file yang berisi definisi mesin status di AWS Toolkit. Untuk informasi rinci tentang mesin AWS Step Functions status, lihat topik [Pelajari tentang mesin status di Step Functions](#) di Panduan AWS Step Functions Pengembang.

### Melihat mesin status Step Functions

Untuk melihat ASL file yang ada yang berisi definisi mesin status di AWS Toolkit Explorer, selesaikan langkah-langkah berikut.

1. Dari AWS Toolkit Explorer, perluas wilayah yang berisi ASL file yang ingin Anda lihat.
2. Perluas heading Step Functions.
3. ASL File Anda ditampilkan di AWS Explorer.

### Membuat mesin status Step Functions

Di AWS Toolkit, Anda dapat membuat mesin status Step Functions baru dari file atau Anda dapat menggunakan template. Prosedur berikut menjelaskan cara membuat mesin status Step Functions dari file. Untuk detail tentang membuat SFN; state machine dari template, lihat bagian State machine templates yang terletak di bawah ini, dalam topik Panduan Pengguna ini.

#### Note

Untuk bekerja dengan Step Functions dalam VS Code, ekstensi file Amazon State Language (ASL) Anda yang berisi definisi mesin status Anda harus diakhiri dengan `asl.json`, `asl.yml`, atau `asl.yaml`.

Secara default, file Step Functions yang relevan terbuka di Workflow Studio. Untuk informasi rinci tentang bekerja di Workflow Studio melalui AWS Toolkit, lihat topik [Bekerja dengan Workflow Studio](#) di Panduan Pengguna ini.

1. Dari ruang kerja Anda di VS Code, buat file baru.
2. Beri nama file Anda dan tentukan ekstensi file sebagai `sl.json`, `asl.yaml`, atau `asl.yaml`.
3. Setelah pembuatan, AWS Toolkit membuka file baru di AWS Step Functions Workflow Studio.
4. Dari Workflow Studio pilih tombol Simpan dari menu utilitas untuk menyimpan ASL file baru Anda.

## Membuat mesin status Step Functions dari template

Di AWS Toolkit, Anda dapat membuat mesin status Step Functions dari template. Proses template membuat ASL file yang berisi definisi mesin negara, memberikan titik awal untuk proyek Anda. Prosedur berikut menjelaskan cara membuat mesin status Step Functions dari template di AWS Toolkit.

1. Dari AWS Toolkit Explorer, perluas wilayah tempat Anda ingin membuat mesin status Step Functions.
2. Buka menu konteks untuk (klik kanan) Step Functions dan pilih Create a new Step Functions state machine untuk membuka wizard Select a starter template (1/2) di VS Code.
3. Dari wizard Select a starter template (1/2), pilih jenis template untuk mesin status Step Functions Anda untuk melanjutkan.
4. Dari Pilih format template (2/2) layar, pilih YANG atau JSON untuk format template Anda.
5. ASL File baru yang berisi definisi mesin status Anda dibuka di editor VS Code.

## Mengunduh mesin status Step Functions

Untuk mengunduh mesin status Step Functions yang disimpan dari jarak jauh ke instance VS Code lokal Anda, selesaikan langkah-langkah berikut.

1. Dari AWS Toolkit Explorer, perluas wilayah yang berisi mesin status Step Functions yang ingin Anda unduh.
2. Perluas Step Functions, lalu klik kanan mesin status Step Functions yang ingin Anda unduh dan pilih Download Definition... .

3. Tentukan lokasi untuk menyimpan mesin status Step Functions Anda secara lokal untuk melanjutkan.
4. Mesin status Step Functions terbuka di Workflow Studio saat prosedur selesai.

## Menyimpan perubahan ke mesin status Step Functions

Prosedur berikut menjelaskan cara menyimpan perubahan yang dilakukan pada mesin status Step Functions Anda.

### Note

Pengeditan yang dibuat di Workflow Studio disinkronkan ke file lokal Anda, tetapi tetap tidak disimpan hingga pekerjaan Anda disimpan di editor VS Code atau Workflow Studio. Jika file lokal Anda diubah dan disimpan saat Workflow Studio terbuka dan tidak ada kesalahan yang terdeteksi dalam ASL file Anda, maka Anda menerima pemberitahuan Sukses di Workflow Studio, saat penyimpanan selesai. Namun, jika file lokal Anda berisi tidak valid JSON atau YAML dan Anda mencoba untuk menyimpan, maka file lokal Anda gagal disinkronkan dan Anda menerima pemberitahuan Peringatan di Workflow Studio.

1. Dari ASL file terbuka yang berisi definisi mesin status di Workflow Studio, navigasikan ke tombol Utility.
2. Pilih tombol Simpan.
3. VS Code memberi tahu Anda ketika file telah disimpan.

## Menjalankan mesin status Step Functions

Prosedur berikut menjelaskan cara menjalankan mesin status Step Functions di AWS Toolkit.

1. Dari AWS Toolkit Explorer, perluas wilayah yang berisi mesin status Step Functions yang ingin Anda jalankan.
2. Perluas Step Functions, lalu klik kanan mesin status Step Functions yang ingin Anda jalankan.
3. Dari menu konteks, pilih Mulai Eksekusi untuk memulai proses peluncuran.
4. Status peluncuran ditampilkan di jendela Output AWS Toolkit di VS Code.

## Bekerja dengan cuplikan kode

Cuplikan kode adalah saran otomatis yang dihasilkan berdasarkan kode yang sedang Anda kerjakan. Untuk bekerja dengan cuplikan kode dengan Step Functions di toolkit, selesaikan langkah-langkah berikut.

### Note

Untuk bekerja dengan cuplikan kode Step Functions di VS Code, ekstensi ASL file Anda yang berisi definisi mesin status Anda harus diakhiri dengan `.asl.json`, `.asl.yaml`, atau `.asl.yml`.

Secara default, file Step Functions Anda yang relevan terbuka di Workflow Studio.

1. Dari VS Code, buka ASL file yang berisi definisi mesin status yang ingin Anda ubah atau buat ASL file baru.
2. Dari Workflow Studio, beralih ke mode Kode jika Anda berada dalam mode Desain.
3. Dari editor kode Workflow Studio, tempatkan kursor Anda di properti. "States"
4. Tekan **control + space** untuk membuka menu cuplikan kode, properti tambahan dapat diakses dengan menekan **control + space** dan didasarkan pada. "State" "Type"
5. Pilih cuplikan kode yang Anda inginkan dari daftar.

## Validasi kode

Saat Anda mengerjakan Step Functions di Workflow Studio, validasi kode secara aktif mengidentifikasi kesalahan dan memberikan saran untuk hal-hal berikut:

- Properti yang hilang
- Nilai yang salah
- Status non terminal
- Tidak ada keadaan yang ditunjuk

## Bekerja dengan AWS Step Functions Workflow Studio

Bagian berikut menjelaskan cara bekerja dengan AWS Step Functions Workflow Studio di AWS Toolkit for Visual Studio Code Untuk informasi rinci tentang AWS Step Functions Workflow Studio, lihat topik [Mengembangkan alur kerja](#) di Panduan Pengembang AWS Step Functions

### Membuka Alur Kerja Studio

Daftar berikut menjelaskan berbagai jalur yang tersedia bagi Anda untuk membuka Workflow Studio di VS Code.

#### Note

Untuk bekerja dengan Workflow Studio di VS Code, ekstensi file Amazon State Language (ASL) Anda yang berisi definisi mesin status Anda, harus diakhiri dengan `asl.json`, `asl.yaml` atau `asl.yml` Untuk detail tentang mengunduh atau membuat definisi mesin status baru di AWS Toolkit, lihat bagian [Mengunduh mesin status](#) dan [Membuat mesin status](#) dalam AWS Step Functions topik [Bekerja dengan](#) Panduan Pengguna ini.

- Dari AWS Explorer, buka menu konteks untuk (klik kanan) ASL file yang berisi definisi mesin status, lalu pilih Buka di Workflow Studio.
- Dari ASL file terbuka yang berisi definisi mesin status, pilih ikon Buka dengan Workflow Studio yang terletak di sebelah tab di jendela editor VS Code.
- Dari ASL file terbuka yang berisi definisi mesin negara, pilih CodeLens perintah Buka dengan Workflow Studio, yang terletak di bagian atas file.
- Menutup dan membuka kembali ASL file yang berisi definisi mesin status secara otomatis membuka kembali file di Workflow Studio, kecuali Workflow Studio default dinonaktifkan secara manual.

### Mode desain dan mode Kode

Workflow Studio memiliki dua mode untuk bekerja dengan ASL file Anda yang berisi definisi mesin status: Mode desain dan mode Kode. Mode desain menyediakan antarmuka grafis untuk memvisualisasikan alur kerja Anda saat Anda membuat prototipe. Mode kode memiliki editor kode terintegrasi di mana Anda dapat melihat, menulis, dan mengedit ASL definisi dalam alur kerja Anda.

**Note**

Untuk informasi mendetail tentang setiap bagian UI dalam mode Desain dan Kode, lihat topik [Menggunakan Workflow Studio](#) di Panduan AWS Step FunctionsPengembang. Tidak semua fitur Workflow Studio tersedia di AWS Toolkit, seperti mode Config, misalnya.

UI mode Desain memiliki 7 bagian utama, seperti yang diberi label dan dijelaskan pada gambar berikut.

1. Tombol Mode: Tombol untuk beralih antara mode Desain dan Kode.
2. Tombol utilitas: Satu set tombol untuk melakukan tugas, seperti keluar dari Workflow Studio, menyimpan alur kerja Anda, atau mengekspor ASL definisi dalam file JSON atau YANG.
3. Toolbar desain: Toolbar yang berisi serangkaian tombol yang melakukan tindakan umum, seperti membatalkan, menghapus, dan memperbesar kontrol.
4. States Browser: Browser yang berisi drag-and-drop status untuk kanvas alur kerja Anda. Status diatur ke dalam tab dan didefinisikan sebagai Tindakan, Aliran, dan Pola.
5. Grafik Canvas dan alur kerja: Sebuah rendering visual dari alur kerja Anda di mana Anda dapat menghapus, mengatur ulang, dan memilih status untuk konfigurasi.
6. Inspector Panel: Lihat dan edit properti dari setiap status yang dipilih pada kanvas. Bergantung pada status yang dipilih dalam grafik alur kerja kanvas, tab diisi dengan opsi khusus status untuk Konfigurasi, Input/Output, Variabel, dan penanganan Kesalahan.
7. Tautan info: Membuka panel dengan informasi kontekstual saat Anda membutuhkan bantuan. Panel ini juga menyertakan tautan ke topik terkait di Panduan AWS Step FunctionsPengembang.

The screenshot displays the AWS Step Functions Workflow Studio interface within VS Code. The main canvas shows a workflow diagram with the following states:

- Start** (Yellow circle)
- Pass state: Set Variables and State Output** (Blue box)
- Choice state: Is Hello World Example?** (Blue box with a question mark)
- Wait state: Wait for X Seconds** (Blue box with a clock icon)
- Parallel state: Execute in Parallel** (Blue box with two arrows)
- Pass state: Format Execution Start Date** (Blue box)
- Pass state: Snapshot Execution Elapsed Time** (Blue box)
- Pass state: Set Checkpoint** (Blue box)
- Fail state: Fail the Execution** (Red box with a red X)
- Succeed state: Summarize the Execution** (Green box with a green checkmark)
- End** (Yellow circle)

The interface includes a top bar with 'Design' and 'Code' tabs, a search bar, and a sidebar with 'Actions', 'Flow', and 'Patterns' sections. A 'State query language' panel is open on the right, showing 'JSONata' as the selected language and a 'Set Variables and State Output' state selected in the workflow.

## Menggunakan tes single-state selama desain

Dari UI status uji Workflow Studio, Anda dapat menguji status individual mesin status Anda. Ini termasuk kemampuan untuk memberikan input status, mengatur variabel, dan membuat keduanya AWS SAM dan substitusi CloudFormation definisi.

Untuk mempelajari selengkapnya tentang infrastruktur sebagai kode (IaC), definisi sumber daya, dan data transformasi, lihat [Menggunakan AWS SAM untuk membangun alur kerja Step Functions](#) dan [Transforming data dengan topik JSONata Functions di Panduan Pengembang AWS Step Functions](#)

Prosedur berikut menjelaskan cara membuka UI status pengujian di Workflow Studio.

### Membuka UI status pengujian

1. Dari tab Mode Desain di Workflow Studio, navigasikan ke kanvas dan pilih status untuk membukanya di panel Inspector.
2. Dari panel Inspector, pilih tombol Test state.
3. UI status Uji terbuka di VS Code.

UI status uji memiliki 3 tab utama, input Uji, Argumen & Output, definisi Status. Tab masukan Uji memiliki 3 bidang tambahan yang memungkinkan Anda memberikan masukan Status, mengatur variabel, dan menentukan substitusi Definisi dari templat AWS SAM atau CloudFormation Anda. Di tab Definisi negara, Anda dapat menyesuaikan alur kerja dan menguji ulang. Setelah selesai menjalankan pengujian, Anda dapat menerapkan dan menyimpan perubahan pada definisi mesin status Anda.

Tangkapan layar berikut menunjukkan UI status uji, yang menyertakan definisi sumber daya topik.

**Test state**  
Test a state in isolation using the TestState API to ensure that it works correctly. [Learn more](#)

**Test input** | Arguments & Output | State definition

**Execution role**  
Testing a state requires an execution role. Enter an IAM role ARN, select an existing IAM role from the list, or [learn how to create a new IAM role with permissions for an optimized service integration](#)

Admin

**Definition substitutions**  
Enter values for any definition substitutions.

Key	Value
{topic}	arn:aws:sns:ca-central-1:652323157371:mySnsTr

**State input - optional**  
Enter input values for this state.

```
1 {
  "key": "value"
}
```

Must be in valid JSON format.

**Variables - optional**  
Enter values for any variables referenced.

```
1 {
  "variableName": "value"
}
```

Must be in valid JSON format.

**Start test**

**Basic** | **Advanced**

**Task state request**  
Request that will be sent to the Task state.

```
1 Start a test to view the output.
```

**Task state response**  
Response received from the Task state.

```
1 Start a test to view the output.
```

**State output**  
Output that will be passed to the next state.

```
1 Start a test to view the output.
```

**Variables**  
Variables at end of test.

```
1 Start a test to view the output.
```

[Copy TestState API response](#) [Apply changes and close](#)

## Menonaktifkan Workflow Studio secara default

Secara default, Workflow Studio adalah editor default untuk ASL file yang berisi definisi mesin status. Anda dapat menonaktifkan pengaturan default dengan memodifikasi `settings.json` file Anda di `.vscode` direktori lokal Anda. Jika Anda menonaktifkan Workflow Studio secara default, itu masih dapat diakses melalui metode yang tercantum di bagian Membuka Workflow Studio, yang terletak di topik ini.

Untuk mengedit `settings.json` file Anda dari VS Code, selesaikan langkah-langkah berikut.

1. Dari VS Code, buka Command Palette dengan menekan **option+shift+p** (Mac) atau **ctrl+shift+p** (Windows).
2. Dari Palet Perintah Kode VS, **Open User Settings (JSON)** masuk ke bidang pencarian dan pilih opsi saat terisi dalam daftar.
3. Dari `settings.json` editor Anda, tambahkan modifikasi berikut ke file Anda.

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

4. Simpan perubahan Anda `settings.json` dan segarkan atau mulai ulang VS Code.

## 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 Streams, peran IAM, Amazon CloudFront VPC, dan 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 dari file JSON yang menjelaskan sumber daya.
- Salin pengenalan sumber daya untuk sumber daya.
- Lihat AWS dokumentasi yang menjelaskan tujuan dari jenis sumber daya dan skema (dalam format JSON dan YAMAL) untuk pemodelan sumber daya.
- Buat sumber daya baru dengan mengedit dan menyimpan template berformat JSON 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 IDE Kode VS Anda, dan 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](#).

## Izin IAM untuk mengakses sumber daya

Anda memerlukan AWS Identity and Access Management izin khusus untuk mengakses sumber daya yang terkait dengan AWS layanan. Misalnya, entitas IAM, seperti pengguna atau peran, memerlukan izin Lambda untuk mengakses sumber daya. `AWS Toolkit:Lambda::Function`

Selain izin untuk sumber daya layanan, entitas IAM memerlukan izin untuk mengizinkan Toolkit for VS Code memanggil operasi AWS Cloud Control API atas namanya. Operasi Cloud Control API memungkinkan pengguna atau peran IAM untuk mengakses dan memperbarui sumber daya jarak jauh.

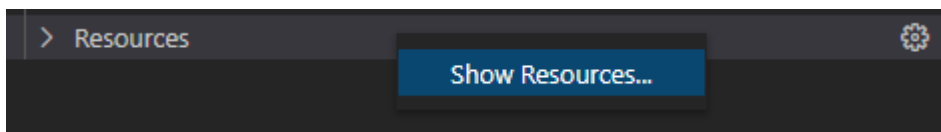
Cara termudah untuk memberikan izin adalah dengan melampirkan kebijakan AWS terkelola PowerUserAccess, ke entitas IAM yang memanggil operasi API ini menggunakan antarmuka Toolkit. [Kebijakan terkelola](#) ini memberikan berbagai izin untuk melakukan tugas pengembangan aplikasi, termasuk memanggil operasi API.

Untuk izin khusus yang menentukan operasi API yang diizinkan pada sumber daya jarak jauh, lihat [Panduan Pengguna AWS Cloud Control API](#).

## 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 untuk 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:

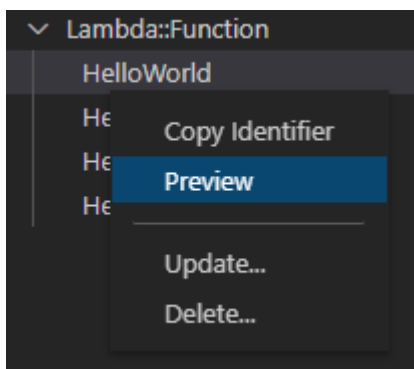
- Copy Resource Identifier: Salin pengenalan untuk sumber daya tertentu ke clipboard. (Misalnya, `AWS Toolkit::DynamoDB::Table` sumber daya dapat diidentifikasi menggunakan `TableName` properti.)
- Pratinjau: Lihat versi read-only dari template berformat JSON yang menjelaskan sumber daya.

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 berformat JSON untuk sumber daya di editor VS Code. 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 CloudFormation template tumpukan. Untuk informasi lebih lanjut, lihat [artikel Pusat Pengetahuan ini](#).



## Membuat dan mengedit sumber daya

### ⚠ Important

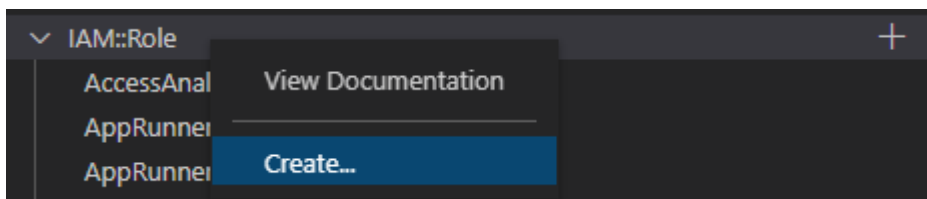
Pembuatan dan pembaruan sumber daya saat ini [???](#) dalam versi AWS Toolkit for Visual Studio Code.

Membuat sumber daya baru melibatkan penambahan jenis sumber daya ke daftar Sumber Daya dan kemudian mengedit template berformat JSON 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 AI 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.

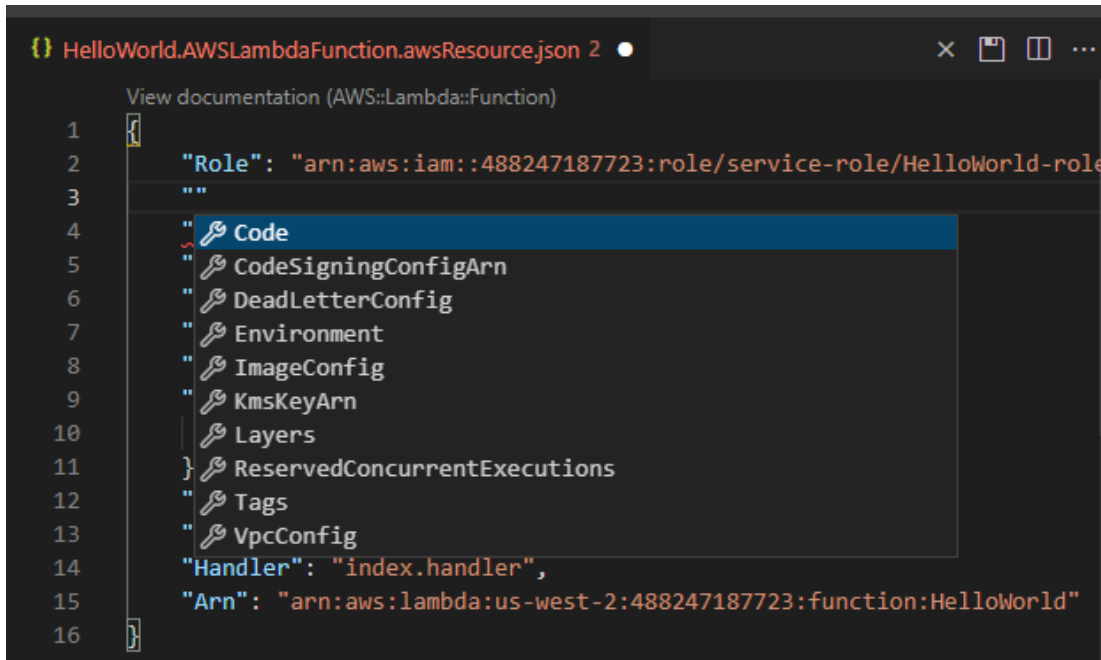
1. Tambahkan jenis sumber daya untuk sumber daya yang ingin Anda buat dengan mengklik kanan Resources dan memilih Show Resources.
2. Setelah jenis sumber daya ditambahkan di bawah Sumber daya, pilih ikon plus (“+”) untuk membuka file templat di editor baru.

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.



The screenshot shows a code editor window titled "HelloWorld.AWSLambdaFunction.awsResource.json 2". The editor displays a JSON configuration for an AWS Lambda function. A dropdown menu is open over the "Code" property, listing various configuration options: Code, CodeSigningConfigArn, DeadLetterConfig, Environment, ImageConfig, KmsKeyArn, Layers, ReservedConcurrentExecutions, Tags, and VpcConfig. The "Code" option is currently selected and highlighted in blue. The JSON content visible includes:

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

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 skema, 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.

# Memecahkan masalah AWS Toolkit for Visual Studio Code

Bagian berikut berisi informasi pemecahan masalah umum tentang AWS Toolkit for Visual Studio Code dan bekerja dengan AWS layanan dari toolkit. Untuk masalah yang secara khusus terkait dengan pemecahan masalah SAM di AWS Toolkit, lihat topik [Pemecahan Masalah aplikasi tanpa server](#) di Panduan Pengguna ini.

## Topik

- [Memecahkan masalah praktik terbaik](#)
- [Profil... tidak dapat ditemukan di file konfigurasi](#)
- [Skema SAM json: tidak dapat mengubah skema dalam file template.yaml](#)

## Memecahkan masalah praktik terbaik

Berikut ini adalah praktik terbaik yang disarankan saat memecahkan masalah AWS Toolkit for Visual Studio Code . Untuk informasi terperinci tentang bagaimana Anda dapat berkontribusi pada topik AWS Toolkit for Visual Studio Code, lihat [Berkontribusi](#) pada AWS Toolkit for Visual Studio Code topik di AWS Toolkit for Visual Studio Code GitHub repositori.

- Cobalah untuk membuat ulang masalah atau kesalahan Anda sebelum mengirim laporan.
- Buat catatan rinci dari setiap langkah, pengaturan, dan pesan kesalahan selama proses rekreasi.
- Kumpulkan AWS Log Debug Toolkit Anda. Untuk penjelasan rinci tentang cara menemukan log Debug AWS Toolkit Anda, lihat prosedur Cara menemukan AWS log Anda, yang terletak di topik panduan pengguna ini.
- Periksa permintaan terbuka, solusi yang diketahui, atau laporkan masalah Anda yang belum terselesaikan di bagian [AWS Toolkit for Visual Studio Code Masalah](#) di AWS Toolkit for Visual Studio Code GitHub repositori.

### Note

Prosedur berikut menjelaskan cara melihat log Debug AWS Toolkit Anda. Proses untuk melihat log Debug Amazon Q Anda identik kecuali Anda memilih Amazon Q: Lihat Log dari Palet Perintah Kode VS.

## Cara menemukan log AWS Toolkit for Visual Studio Code Debug Anda

1. Dari VS Code buka Command Palette dengan menekan **Cmd + Shift + P** atau **Ctrl + Shift + P** (Windows) dan **AWS View Logs** masuk ke bidang pencarian.
2. Pilih AWS Lihat Log untuk membuka log AWS Toolkit Anda di jendela output terminal VS Code.
3. Dari jendela output terminal VS Code, perluas menu ikon Gear dan pilih Debug.
4. Perluas menu ikon Gear lagi dan pilih Set As Default.
5. Buka kembali Command Palette dengan menekan **Cmd + Shift + P** atau **Ctrl + Shift + P** (Windows) dan cari **Reload Window**, lalu pilih Developer: Reload Window.
6. VS Code reload dan jendela output terminal VS Code menampilkan log Debug AWS Toolkit Anda yang diperbarui.

## Profil... tidak dapat ditemukan di file konfigurasi

### Masalah

#### Note

Masalah ini hanya berlaku untuk `~/.aws/config` file dan bukan `~/.aws/credentials` file. Untuk informasi mendetail tentang file AWS konfigurasi dan AWS kredensial, lihat topik [File konfigurasi dan kredensial bersama di panduan referensi SDK](#) dan Alat.AWS

Saat memilih kredensi Log AWS Toolkit menampilkan pesan dengan struktur ini: `Profile name could not be found in shared credentials file`

Berikut ini adalah contoh dari apa kesalahan ini terlihat seperti di log AWS Toolkit Anda:

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

### Solusi

Jika profil Anda sudah ada di `~/.aws/config`, periksa apakah itu dimulai dengan `[profile ]`. Berikut ini adalah contoh profil pengguna yang terstruktur dengan benar:

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

Berikut ini adalah contoh profil pengguna yang tidak terstruktur dengan benar:

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

## Skema SAM json: tidak dapat mengubah skema dalam file `template.yaml`

### Masalah

Anda tidak dapat memilih skema json yang berbeda secara manual di SAM `template.yaml`

### Solusi

Setelah memperbarui ke `vscode-yaml` versi 1.11+, Anda dapat menambahkan **`yaml-language-server`** modeline ke bagian atas file YAMAL untuk memaksa penggunaan skema oleh URI. Untuk informasi tambahan tentang [Menggunakan bagian skema sebaris](#) dalam topik server bahasa yaml dari repositori pengembang Redhat. GitHub Berikut ini adalah contoh **`yaml-language-server`** modeline.

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

# 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. Lihat informasi yang lebih lengkap tentang privasi data dalam [Pertanyaan Umum Privasi Data](#). Lihat informasi tentang perlindungan data di Eropa di pos blog [Model Tanggung Jawab Bersama dan GDPR AWS](#) di Blog Keamanan AWS .

Untuk tujuan perlindungan data, kami menyarankan Anda melindungi Akun AWS kredensial 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 autentikasi multi-faktor (MFA) pada setiap akun.
- Gunakan SSL/TLS untuk berkomunikasi dengan sumber daya. AWS Kami mensyaratkan TLS 1.2 dan menganjurkan TLS 1.3.
- Siapkan API dan pencatatan aktivitas pengguna dengan AWS CloudTrail. Untuk informasi tentang penggunaan CloudTrail jejak untuk menangkap AWS aktivitas, lihat [Bekerja dengan CloudTrail jejak](#) di AWS CloudTrail Panduan Pengguna.
- Gunakan solusi AWS enkripsi, bersama dengan semua kontrol keamanan default di dalamnya Layanan AWS.
- Gunakan layanan keamanan terkelola tingkat lanjut seperti Amazon Macie, yang membantu menemukan dan mengamankan data sensitif yang disimpan di Amazon S3.
- Jika Anda memerlukan modul kriptografi tervalidasi FIPS 140-3 saat mengakses AWS melalui antarmuka baris perintah atau API, gunakan titik akhir FIPS. Lihat informasi selengkapnya tentang titik akhir FIPS yang tersedia di [Standar Pemrosesan Informasi Federal \(FIPS\) 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 konsol, API 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. Saat Anda memberikan URL ke server eksternal, kami sangat menganjurkan supaya Anda tidak menyertakan informasi kredensial di dalam URL untuk memvalidasi permintaan Anda ke server itu.

# 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 notifikasi tentang pembaruan dokumentasi ini, Anda dapat berlangganan ke [umpan RSS](#).

Perubahan	Deskripsi	Tanggal
<a href="#">Studio SageMaker Terpadu Amazon</a>	Integrasi dengan layanan Amazon SageMaker Unified Studio.	September 18, 2025
<a href="#">LocalStack</a>	Topik Panduan Pengguna Baru ditambahkan untuk mendukung peluncuran LocalStack.	September 11, 2025
<a href="#">Bekerja dengan AWS Lambda fungsi</a>	Memperbarui topik panduan pengguna untuk menyertakan fitur Lambda yang diperbarui dalam toolkit.	Juli 17, 2025
<a href="#">AWS Lambda debugging jarak jauh</a>	Menambahkan topik debugging AWS Lambda jarak jauh baru ke Panduan AWS Toolkit for Visual Studio Code Pengguna.	Juli 17, 2025
<a href="#">AWS Lambda Konsol ke IDE</a>	Ditambahkan baru AWS Lambda console ke topik IDE ke Panduan AWS Toolkit for Visual Studio Code Pengguna.	Juli 17, 2025
<a href="#">Pembaruan AWS Step Functions konten dan menambahkan dukungan untuk Workflow Studio</a>	Menambahkan pembaruan ke konten yang ada untuk AWS Step Functions dan topik Panduan Pengguna untuk	Maret 6, 2025

---

	AWS Step Functions Workflow Studio, untuk mendukung peluncuran fitur.	
<a href="#">AWS Tanah Tanpa Server</a>	Menambahkan topik AWS Serverless Land baru ke TOC AWS Application Builder.	Maret 6, 2025
<a href="#">Memperbarui firewall dan gateway untuk memungkinkan akses</a>	Daftar titik akhir dan sumber daya yang harus diizinkan terdaftar untuk mengakses semua layanan dan fitur di AWS Toolkit for Visual Studio Code dan Amazon Q untuk ekstensi Kode VS.	Februari 28, 2025
<a href="#">Dukungan untuk Amazon ECR App Runner</a>	Menambahkan dukungan dokumentasi untuk meluncurkan AWS App Runner layanan dari node Amazon Elastic Container Registry, di AWS Toolkit.	Februari 6, 2025
<a href="#">Amazon DocumentDB</a>	Menambahkan topik Amazon DocumentDB baru ke AWS Toolkit for Visual Studio Code Panduan Pengguna.	Februari 6, 2025
<a href="#">EC2 dukungan</a>	Support untuk layanan Amazon Elastic Compute Cloud telah ditambahkan ke toolkit.	Januari 31, 2025
<a href="#">AWS Dokumen</a>	Menambahkan topik panduan pengguna baru untuk AWS Dokumen.	Januari 20, 2025

<a href="#">Amazon CloudWatch Log Ekor Langsung</a>	Menambahkan subtopik baru untuk mendukung fitur Amazon CloudWatch Logs Live Tail di. AWS Toolkit for Visual Studio Code	Desember 15, 2024
<a href="#">AWS Pembuat Aplikasi</a>	Menambahkan topik Pembuat AWS Aplikasi baru ke Panduan AWS Toolkit for Visual Studio Code Pengguna.	Oktober 30, 2024
<a href="#">Komposer Infrastruktur</a>	AWS Application Composer sekarang adalah AWS Infrastructure Composer.	Oktober 3, 2024
<a href="#">AWS Pembaruan Access Analyzer Identity and Access Management (IAM)</a>	Konten IAM Access Analyzer yang diperbarui untuk menyertakan referensi API baru.	Juli 10, 2024
<a href="#">AWS Access Analyzer Identity and Access Management (IAM)</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 mencakup perubahan UI untuk mendukung VPC di Lingkungan Dev.	Januari 21, 2024
<a href="#">Komposer Infrastruktur</a>	Menambahkan topik Komposer Infrastruktur baru ke Panduan AWS Toolkit for Visual Studio Code Pengguna.	28 November 2023
<a href="#">Dukungan SSO untuk CodeCatalyst</a>	Konten yang diperbarui untuk mencakup dukungan Pusat Identitas IAM 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 layanan Amazon ECR untuk AWS Toolkit for Visual Studio Code</a>	Menambahkan ikhtisar dan panduan untuk fitur dan fungsi layanan Amazon ECR 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="#">Dukungan YALL untuk AWS Step Functions</a>	Menambahkan dukungan YAMAL 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="#">Amazon API Gateway</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 tugas Amazon ECS.	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
<a href="#">Konfigurasi rantai alat Anda</a>	Menambahkan informasi tentang mengkonfigurasi berbagai toolchain.	12 Juni 2019
<a href="#">Rilis Awal</a>	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.