

Panduan Pengguna

# AWS Toolkit for Visual Studio



# AWS Toolkit for Visual Studio: Panduan Pengguna

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

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

---

# Table of Contents

AWS Toolkit for Visual Studio .....	1
Apa itu Toolkit for Visual Studio .....	1
AWS Penjelajah .....	1
Kredensi dan Manajemen Wilayah .....	2
Amazon EC2 .....	2
AWS Lambda .....	2
AWS CodeCommit .....	2
Amazon DynamoDB .....	2
Amazon S3 .....	2
Amazon RDS .....	3
AWS Elastic Beanstalk .....	3
AWS CloudFormation .....	3
AWS Identity and Access Management (IAM) .....	3
Informasi Terkait .....	3
Amazon Q dan Amazon CodeWhisperer .....	4
Apa itu Amazon Q .....	4
Unduh Toolkit .....	5
Mengunduh Toolkit dari Marketplace Visual Studio .....	5
Toolkit IDE tambahan dari AWS .....	5
Memulai .....	6
Instalasi dan pengaturan .....	6
Prasyarat .....	6
Menginstal AWS Toolkit .....	7
Menghapus Instalasi Toolkit AWS .....	8
Menghubungkan ke AWS .....	10
Prasyarat .....	10
Menghubungkan ke AWS dari Toolkit .....	10
Otentikasi untuk Pengembang Amazon Q .....	12
Otentikasi untuk Explorer AWS .....	1
Memecahkan masalah instalasi .....	15
Izin administrator untuk Visual Studio .....	15
Mendapatkan log instalasi .....	16
Menginstal ekstensi Visual Studio yang berbeda .....	17
Menghubungi dukungan .....	17

Profil dan Window Binding .....	17
Profil dan Window Binding for Toolkit for Visual Studio .....	17
Otentikasi dan akses .....	19
Pusat Identitas IAM .....	19
Mengautentikasi dengan IAM Identity Center dari AWS Toolkit for Visual Studio .....	20
Kredensial IAM .....	21
Membuat pengguna IAM .....	22
Membuat file kredensial .....	22
Mengedit kredensial pengguna IAM dari toolkit .....	23
Mengedit kredensial pengguna IAM dari editor teks .....	24
Membuat pengguna IAM dari AWS Command Line Interface (AWS CLI) .....	24
AWS ID Pembangun .....	25
Autentikasi multi-faktor (MFA) .....	25
Langkah 1: Membuat peran IAM untuk mendelegasikan akses ke pengguna IAM .....	25
Langkah 2: Membuat pengguna IAM yang mengasumsikan izin peran .....	26
Langkah 3: Menambahkan kebijakan untuk memungkinkan pengguna IAM mengambil peran .....	27
Langkah 4: Mengelola perangkat MFA virtual untuk pengguna IAM .....	28
Langkah 5: Membuat profil untuk memungkinkan MFA .....	28
Kredensial eksternal .....	29
Bekerja dengan AWS Layanan .....	31
Amazon CodeCatalyst .....	31
Apa yang dimaksud dengan Amazon CodeCatalyst? .....	31
Memulai dengan CodeCatalyst .....	32
Bekerja dengan CodeCatalyst .....	33
Pemecahan Masalah .....	35
CloudWatch Integrasi log .....	36
Menyiapkan CloudWatch Beberapa catatan .....	36
Bekerja dengan CloudWatch Beberapa catatan .....	36
Mengelola Instans Amazon EC2 .....	43
Gambar Mesin Amazon dan Tampilan Instans Amazon EC2 .....	43
Meluncurkan Instans Amazon EC2 .....	45
Menghubungkan ke Instans Amazon EC2 .....	48
Mengakhiri Instans Amazon EC2 .....	51
Mengelola instans Amazon ECS .....	54
Memodifikasi properti layanan .....	54

Menghentikan tugas .....	55
Menghapus layanan .....	55
Menghapus klaster .....	56
Membuat repositori .....	56
Menghapus repositori .....	56
Mengelola Grup Keamanan dariAWSPenjelajah .....	56
Membuat Grup Keamanan .....	57
Menambahkan Izin ke Grup Keamanan .....	57
Membuat AMI dari Instans Amazon EC2 .....	59
Menyiapkan Izin Luncurkan pada Amazon Machine Image .....	61
Amazon Virtual Private Cloud (VPC) .....	62
Membuat VPC Publik-Swasta untuk Deployment denganAWS Elastic Beanstalk .....	63
Menggunakan Editor AWS CloudFormation Template untuk Visual Studio .....	67
MembuatAWS CloudFormationProyek Template di Visual Studio .....	68
Deploy aAWS CloudFormationTemplate dalam Visual Studio .....	71
FormatAWS CloudFormationTemplat dalam Visual Studio .....	74
Menggunakan Amazon S3 dariAWSPenjelajah .....	75
Membuat sebuah Bucket Amazon S3 .....	75
Mengelola Bucket Amazon S3AWSPenjelajah .....	75
Mengunggah File dan Folder ke Amazon S3 .....	77
Operasi File Amazon S3AWSToolkit for Visual Studio .....	79
Menggunakan DynamoDB dariAWSPenjelajah .....	83
Membuat Tabel DynamoDB .....	84
Melihat Tabel DynamoDB sebagai Grid .....	85
Mengedit dan Menambahkan Atribut dan Nilai .....	86
Memindai Tabel DynamoDB .....	88
MenggunakanAWS CodeCommitdengan Visual Studio Team Explorer .....	89
Tipe kredensialnya untukAWS CodeCommit .....	90
Terhubung keAWS CodeCommit .....	90
Membuat Repositori .....	92
Menyiapkan Kredensialnya Git .....	93
Kloning Repositori .....	95
Bekerja dengan Repositori .....	96
Menggunakan CodeArtifact di Visual Studio .....	97
Tambahkan repositori CodeArtifact Anda sebagai sumber paket NuGet .....	97
Amazon RDS dariAWSPenjelajah .....	98

Luncurkan Instans Basis Data Amazon RDS .....	99
Membuat Microsoft SQL Server Database dalam Instans RDS .....	106
Grup keamanan Amazon RDS .....	107
Menggunakan Amazon SimpleDB dariAWSPenjelajah .....	111
Menggunakan Amazon SQSAWSPenjelajah .....	113
Membuat Antrean .....	113
Menghapus Antrean .....	114
Mengelola Properti Antrean .....	114
Mengirim Pesan ke Antrian .....	115
Manajemen Identitas dan Akses .....	116
Membuat dan Mengonfigurasi Pengguna IAM .....	117
Buat Grup IAM .....	118
Menambahkan Pengguna IAM ke Grup IAM .....	119
Menghasilkan Kredensial untuk Pengguna IAM .....	121
Buat IAM Role .....	123
Membuat Kebijakan IAM .....	124
AWS Lambda .....	127
AWS Lambda Proyek Dasar .....	127
AWS Lambda Proyek Dasar Membuat Gambar Docker .....	134
Tutorial: Membangun dan Menguji Aplikasi Tanpa Server dengan AWS Lambda .....	142
Tutorial: Membuat Aplikasi Amazon Rekognition Lambda .....	148
Tutorial: Menggunakan Amazon Logging Frameworks dengan AWS Lambda untuk Membuat Log Aplikasi .....	156
Menyebarkan keAWS .....	159
PublikasikanAWS .....	159
Prasyarat .....	160
Tipe aplikasi yang didukung .....	161
Publikasikan aplikasiAWSsasaran .....	161
AWS Lambda .....	163
Prasyarat .....	163
Topik terkait .....	164
Daftar Perintah Lambda Tersedia melalui .NET Core CLI .....	164
Menerapkan .NET Core Lambda Project dari .NET Core CLI .....	165
Menerapkan ke Elastic Beanstalk .....	167
Menyebarkan Aplikasi ASP.NET (Tradisional) .....	167
Menyebarkan Aplikasi ASP.NET (.NET Core) (Legacy) .....	180

TentukanAWSKredensial .....	182
Dipublikasikan ulang ke Elastic Beanstalk (Legacy) .....	183
Deployment (Tradisional) .....	185
Deployment khusus (.NET Core) .....	187
Beberapa Support Aplikasi .....	191
Menyebarkan ke Amazon EC2 Container Service .....	194
MenentukanAWSKredensial .....	195
Menyebarkan Aplikasi ASP.NET Core 2.0 (Fargate) (Legacy) .....	197
Menyebarkan Aplikasi ASP.NET Core 2.0 (EC2) .....	204
Pemecahan Masalah .....	209
Memecahkan masalah praktik terbaik .....	209
Amazon CodeWhisperer Masuk dan Keluar dinonaktifkan .....	210
Keamanan .....	211
Perindungan Data .....	211
Identity and Access Management .....	213
Audiens .....	213
Mengautentikasi dengan identitas .....	214
Mengelola akses menggunakan kebijakan .....	217
Bagaimana Layanan AWS bekerja dengan IAM .....	220
Memecahkan masalah AWS identitas dan akses .....	220
Validasi Kepatuhan .....	222
Ketangguhan .....	224
Keamanan Infrastruktur .....	224
Analisis Konfigurasi dan Kelemahan .....	225
Riwayat dokumen .....	226
Riwayat dokumen .....	226
.....	ccxxxiii

# AWS Toolkit for Visual Studio

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

## Apa itu Toolkit for Visual Studio

AWS Toolkit for Visual Studio Ini adalah plugin untuk Visual Studio IDE yang memudahkan Anda untuk mengembangkan, men-debug, dan menyebarkan aplikasi.NET yang menggunakan Amazon Web Services. Toolkit for Visual Studio didukung untuk Visual Studio versi 2019 dan yang lebih baru. Untuk detail tentang cara mengunduh dan menginstal kit, lihat topik [Instalasi dan penyiapan](#) di Panduan Pengguna ini.

### Note

Toolkit for Visual Studio juga dirilis untuk versi Visual Studio 2008, 2010, 2012, 2013, 2015, dan 2017. Namun, versi tersebut tidak lagi didukung. Untuk informasi selengkapnya, lihat topik [Instalasi dan penyiapan](#) di Panduan Pengguna ini.

Toolkit for Visual Studio berisi fitur-fitur berikut untuk meningkatkan pengalaman pengembangan Anda.

## AWS Penjelajah

Jendela alat AWS Explorer, tersedia dari menu Tampilan IDE, memungkinkan Anda berinteraksi dengan banyak AWS layanan dari dalam Visual Studio IDE. Layanan data yang didukung termasuk Amazon Simple Storage Service (Amazon S3), Amazon SimpleDB, Amazon SimpleDB, Amazon Simple Notification Service (Amazon SNS), Amazon Simple Queue Service (Amazon SQS), dan Amazon. CloudFront AWS Explorer juga menyediakan akses ke manajemen Amazon Elastic Compute Cloud (Amazon EC2) AWS Identity and Access Management , manajemen pengguna dan kebijakan (IAM), penerapan aplikasi dan fungsi tanpa server ke dan penyebaran aplikasi web ke dan. AWS Lambda AWS Elastic Beanstalk AWS CloudFormation



## Kredensi dan Manajemen Wilayah

AWS Explorer mendukung beberapa AWS akun (termasuk akun pengguna IAM) dan wilayah, dan memungkinkan Anda untuk dengan mudah mengubah tampilan yang ditampilkan dari satu akun ke akun lain atau melihat dan mengelola sumber daya dan layanan di berbagai wilayah.

## Amazon EC2

Dari AWS Explorer, Anda dapat melihat Amazon Machine Images (AMI) yang tersedia, membuat instans Amazon EC2 dari AMI tersebut, dan kemudian menyambung ke instans tersebut dengan menggunakan Windows Remote Desktop. AWS Explorer juga memungkinkan fungsionalitas pendukung, seperti kemampuan untuk membuat dan mengelola pasangan kunci dan grup keamanan.

## AWS Lambda

Anda dapat menggunakan Lambda untuk meng-host fungsi .NET Core C# tanpa server dan aplikasi tanpa server Anda. Gunakan cetak biru untuk membuat proyek tanpa server baru dengan cepat dan mulailah mengembangkan aplikasi tanpa server Anda.

## AWS CodeCommit

CodeCommit terintegrasi dengan Visual Studio Team Explorer. Ini memudahkan untuk mengkloning dan membuat repositori yang disimpan CodeCommit, dan bekerja dengan perubahan kode sumber dari dalam IDE.

## Amazon DynamoDB

DynamoDB adalah layanan database nonrelasional yang cepat, sangat terukur, sangat tersedia, hemat biaya, dan nonrelasional. Toolkit for Visual Studio menyediakan fungsionalitas untuk bekerja dengan Amazon DynamoDB dalam konteks pengembangan. Dengan Toolkit for Visual Studio, Anda dapat membuat dan mengedit atribut dalam tabel DynamoDB dan menjalankan operasi pemindaian pada tabel.

## Amazon S3

Anda dapat dengan cepat dan mudah mengunggah konten ke bucket Amazon S3 dengan menyeret dan menjatuhkan, atau mengunduh konten dari Amazon S3. Anda juga dapat mengatur izin, metadata, dan tag dengan nyaman pada objek dalam ember.

## Amazon RDS

AWS Explorer dapat membantu Anda membuat dan mengelola aset Amazon RDS di Visual Studio. Instans Amazon RDS yang menggunakan Microsoft SQL Server juga dapat ditambahkan ke Server Explorer Visual Studio.

## AWS Elastic Beanstalk

Anda dapat menggunakan Elastic Beanstalk untuk menyebarkan proyek aplikasi web.NET Anda. AWS Anda dapat menerapkan aplikasi ke lingkungan instans tunggal atau ke lingkungan yang diskalakan secara otomatis dan seimbang dengan beban penuh dari dalam IDE. Anda juga dapat menerapkan versi baru aplikasi Anda dengan cepat dan nyaman tanpa meninggalkan Visual Studio. Jika aplikasi Anda menggunakan SQL Server di Amazon RDS, wizard penerapan juga dapat mengatur konektivitas antara lingkungan aplikasi Anda di Elastic Beanstalk dan instance database di Amazon RDS. Toolkit for Visual Studio juga menyertakan alat penyebaran baris perintah mandiri. Gunakan alat penyebaran untuk menjadikan penerapan sebagai bagian otomatis dari proses build Anda, atau untuk menyertakan penerapan dalam skenario skrip lain di luar Visual Studio.

## AWS CloudFormation

Anda dapat menggunakan Toolkit for Visual Studio untuk AWS CloudFormation mengedit template format JSON dengan dukungan penyorotan editor dan sintaks. IntelliSense Dengan AWS CloudFormation template Anda menjelaskan sumber daya yang ingin Anda buat instance untuk meng-host aplikasi Anda. Dari dalam IDE Anda kemudian menyebarkan template ke AWS CloudFormation. Sumber daya yang dijelaskan dalam template disediakan untuk Anda, membebaskan Anda untuk fokus mengembangkan fungsionalitas aplikasi.

## AWS Identity and Access Management (IAM)

Dari AWS Explorer, Anda dapat membuat pengguna, peran, dan kebijakan IAM, serta melampirkan kebijakan ke pengguna.

## Informasi Terkait

Untuk membuka masalah atau melihat masalah yang saat ini terbuka, kunjungi <https://github.com/aws/aws-toolkit-visual-studio/issues>.

Untuk mempelajari lebih lanjut tentang Visual Studio, kunjungi <https://visualstudio.microsoft.com/vs/>.

# Amazon Q dan Amazon CodeWhisperer

## Apa itu Amazon Q

Mulai 30 April 2024, Amazon sekarang menjadi bagian dari CodeWhisperer Pengembang Amazon Q, ini termasuk saran kode sebaris dan pemindaian keamanan.

Untuk mempelajari selengkapnya tentang bekerja dengan Pengembang Amazon Q di bagian AWS Toolkit for Visual Studio, lihat topik [Pengembang Amazon Q di IDE](#) di Panduan Pengguna Pengembang Amazon Q. Untuk informasi terperinci tentang paket dan harga Amazon Q, lihat panduan [harga Amazon Q](#).

# Mengunduh Toolkit for Visual Studio

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

## Mengunduh Toolkit dari Marketplace Visual Studio

Unduh file instalasi Toolkit for Visual Studio dengan menavigasi ke situs downloads [Visual Studio AWS di](#) browser web Anda.

## Toolkit IDE tambahan dari AWS

Selain Toolkit for Visual StudioAWS, juga menawarkan IDE Toolkit untuk VS Code dan. JetBrains

AWS Toolkit for Visual Studio Codelink

- Ikuti tautan ini untuk [Mengunduh AWS Toolkit for Visual Studio Code](#) dari VS Code Marketplace.
- Untuk mempelajari selengkapnyaAWS Toolkit for Visual Studio Code, lihat Panduan [AWS Toolkit for Visual Studio Code](#)Pengguna.

AWS Toolkit for JetBrainslink

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

# Memulai

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

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

Topik

- [Menginstal dan mengatur AWS Toolkit for Visual Studio](#)
- [Menghubungkan ke AWS](#)
- [Memecahkan masalah instalasi AWS Toolkit for Visual Studio](#)
- [Profil dan Window Binding](#)

## Menginstal dan mengatur AWS Toolkit for Visual Studio

Topik berikut menjelaskan cara mengunduh, menginstal, mengatur, dan menghapus instalasi. AWS Toolkit for Visual Studio

Topik

- [Prasyarat](#)
- [Memasang AWS Toolkit for Visual Studio](#)
- [Menghapus instalasi AWS Toolkit for Visual Studio](#)

## Prasyarat

Berikut ini adalah prasyarat untuk menyiapkan versi yang didukung dari AWS Toolkit for Visual Studio

- Visual Studio 19 atau rilis yang lebih baru
- Windows 10 atau rilis Windows yang lebih baru
- Akses administrator ke Windows dan Visual Studio
- Kredensial AWS IAM Aktif

**Note**

Versi yang tidak didukung AWS Toolkit for Visual Studio tersedia untuk Visual Studio 2008, 2010, 2012, 2013, 2015, dan 2017. Untuk mengunduh versi yang tidak didukung, navigasikan ke halaman [AWS Toolkit for Visual Studio](#) arahan dan pilih versi yang Anda inginkan dari daftar tautan unduhan.

[Untuk mempelajari lebih lanjut tentang kredensial IAM atau mendaftar akun, kunjungi gateway Konsol.AWS](#)

## Memasang AWS Toolkit for Visual Studio

Untuk menginstal AWS Toolkit for Visual Studio, temukan versi Visual Studio Anda dari prosedur berikut dan selesaikan langkah-langkah yang diperlukan. Tautan unduhan untuk semua versi AWS Toolkit for Visual Studio dapat ditemukan di halaman [AWS Toolkit for Visual Studio](#) arahan.

**Note**

Jika Anda mengalami masalah saat menginstal AWS Toolkit for Visual Studio, lihat topik Masalah [penginstalan pemecahan masalah](#) di panduan ini.

## Menginstal AWS Toolkit for Visual Studio untuk Visual Studio 2022

Untuk menginstal AWS Toolkit for Visual Studio 2022 dari Visual Studio, selesaikan langkah-langkah berikut:

1. Dari menu Utama, navigasikan ke Ekstensi dan pilih Kelola Ekstensi.
2. Dari kotak pencarian, cari AWS.
3. Pilih tombol Unduh untuk versi Visual Studio 2022 yang relevan dan ikuti petunjuk penginstalan.

**Note**

Anda mungkin perlu menutup dan memulai ulang Visual Studio secara manual untuk menyelesaikan proses instalasi.

4. Ketika download dan instalasi selesai, Anda dapat membuka AWS Toolkit for Visual Studio dengan memilih AWS Explorer dari menu View.

## Menginstal AWS Toolkit for Visual Studio untuk Visual Studio 2019

Untuk menginstal AWS Toolkit for Visual Studio 2019 dari Visual Studio, selesaikan langkah-langkah berikut:

1. Dari menu Utama, navigasikan ke Ekstensi dan pilih Kelola Ekstensi.
2. Dari kotak pencarian, cari AWS.
3. Pilih tombol Unduh untuk Visual Studio 2017 dan 2019 dan ikuti petunjuknya.

### Note

Anda mungkin perlu menutup dan memulai ulang Visual Studio secara manual untuk menyelesaikan proses instalasi.

4. Ketika download dan instalasi selesai, Anda dapat membuka AWS Toolkit for Visual Studio dengan memilih AWS Explorer dari menu View.

## Menghapus instalasi AWS Toolkit for Visual Studio

Untuk menghapus instalasi AWS Toolkit for Visual Studio, temukan versi Visual Studio Anda dari prosedur berikut dan selesaikan langkah-langkah yang diperlukan.

## Menghapus instalasi AWS Toolkit for Visual Studio untuk Visual Studio 2022

Untuk Menghapus AWS Toolkit for Visual Studio 2022 dari Visual Studio, selesaikan langkah-langkah berikut:

1. Dari menu Utama, navigasikan ke Ekstensi dan pilih Kelola Ekstensi.
2. Dari menu navigasi Kelola Ekstensi, perluas judul Terpasang.
3. Temukan ekstensi AWS Toolkit for Visual Studio 2022 dan pilih tombol Copot pemasangan.

### Note

Jika AWS Toolkit for Visual Studio tidak terlihat dari bagian Terinstal pada menu navigasi, Anda mungkin perlu memulai ulang Visual Studio.

4. 4. Ikuti petunjuk di layar untuk menyelesaikan proses pencopotan instalasi.

## Menghapus instalasi AWS Toolkit for Visual Studio untuk Visual Studio 2019

Untuk menghapus AWS Toolkit for Visual Studio 2019 dari Visual Studio, selesaikan langkah-langkah berikut:

1. Dari menu Utama, navigasikan ke Alat dan pilih Kelola Ekstensi.
2. Dari menu navigasi Kelola Ekstensi, perluas judul Terpasang.
3. Temukan ekstensi AWS Toolkit for Visual Studio 2019 dan pilih tombol Uninstall.
4. 4. Ikuti petunjuk di layar untuk menyelesaikan proses pencopotan instalasi.

## Menghapus instalasi AWS Toolkit for Visual Studio untuk Visual Studio 2017

Untuk menghapus AWS Toolkit for Visual Studio 2017 di Visual Studio, selesaikan langkah-langkah berikut:

1. Dari menu Utama, navigasikan ke Alat dan pilih Ekstensi dan Pembaruan.
2. Dari menu navigasi Ekstensi dan Pembaruan, perluas judul Terinstal.
3. Temukan ekstensi AWS Toolkit for Visual Studio 2017 dan pilih tombol Uninstall.
4. 4. Ikuti petunjuk di layar untuk menyelesaikan proses pencopotan instalasi.

## Menghapus instalasi AWS Toolkit for Visual Studio untuk Visual Studio 2013 atau 2015

Untuk menghapus instalasi AWS Toolkit for Visual Studio 2013 atau 2015, selesaikan langkah-langkah berikut:

1. Dari Panel Kontrol Windows Anda, buka Program dan Fitur.

### Note

Anda dapat membuka Program dan Fitur segera dengan menjalankan `appwiz.cpl` dari prompt perintah Windows atau dialog Windows Run.

2. Dari daftar program yang diinstal, buka menu konteks untuk (klik kanan) AWS Alat untuk Windows.
3. Pilih Uninstall dan ikuti petunjuk untuk menyelesaikan proses uninstall.



**Note**

Direktori Sampel Anda tidak dihapus selama proses uninstall. Direktori ini dipertahankan jika Anda telah memodifikasi sampel. Direktori ini harus dihapus secara manual.

## Menghubungkan ke AWS

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

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

## Prasyarat

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

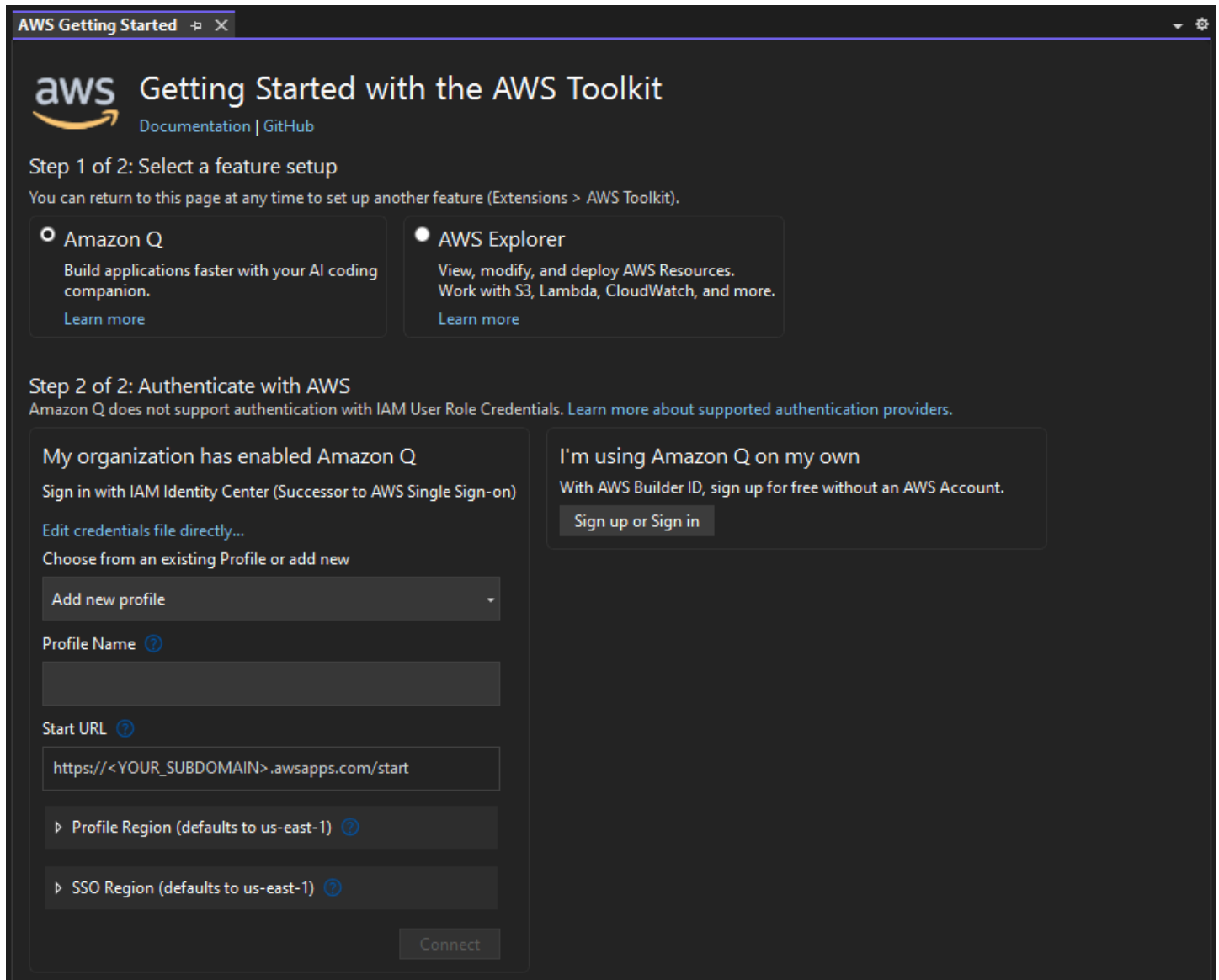
1. Mendaftar untuk AWS akun: Anda dapat mendaftar untuk AWS akun dari [portal AWS pendaftaran](#). Untuk informasi lebih lanjut tentang cara menyiapkan AWS akun baru, lihat topik [Ringkasan](#) di Panduan Pengguna AWS Pengaturan.
2. Menyiapkan otentikasi: Ada 3 metode utama untuk mengautentikasi dengan AWS akun Anda dari Toolkit for Visual Studio. 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.

## Menghubungkan ke AWS dari Toolkit

Untuk menyambung ke AWS akun Anda dari Toolkit for Visual Studio, buka Memulai dengan AWS Antarmuka Pengguna Toolkit (UI koneksi) dengan menyelesaikan prosedur berikut.

1. Dari menu utama Visual Studio, perluas Extensions lalu perluas AWS Toolkit.
2. Dari opsi menu AWS Toolkit pilih Memulai.

### 3. Memulai dengan UI koneksi AWS Toolkit terbuka di Visual Studio.



Tabel berikut menjelaskan metode otentikasi mana yang kompatibel dengan setiap fitur. Untuk mempelajari lebih lanjut tentang masing-masing dari 3 metode autentikasi AWS IAM Identity Center, AWS Identity and Access Management kredensial, dan ID AWS Pembuat, lihat daftar isi [Otentikasi dan akses](#) dalam Panduan Pengguna ini.

#### **Note**

Saat ini saat bekerja dengan CodeCatalyst Toolkit for Visual Studio, Anda hanya perlu mengotorisasi dengan Builder ID AWS Anda saat mengkloning repositori pihak ke-3.

Pengembang Amazon Q	AWS Penjelajah	Amazon CodeCatalyst
<input checked="" type="checkbox"/> ID AWS Pembangun	<input checked="" type="checkbox"/> ID AWS Pembangun	<input checked="" type="checkbox"/> ID AWS Pembangun
<input checked="" type="checkbox"/> Pusat Identitas IAM	<input checked="" type="checkbox"/> Pusat Identitas IAM	<input checked="" type="checkbox"/> Pusat Identitas IAM
<input checked="" type="checkbox"/> AWS Kredensi IAM	<input checked="" type="checkbox"/> AWS Kredensi IAM	<input checked="" type="checkbox"/> AWS Kredensi IAM

## Otentikasi untuk Pengembang Amazon Q

Untuk memulai dengan Amazon Q Developer, autentikasi dan sambungkan dengan kredensi AWS IAM Identity Center atau AWS Builder ID Anda.

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

### Mengautentikasi dan terhubung dengan IAM Identity Center

1. Dari Memulai dengan UI koneksi AWS Toolkit, pilih radial Pengembang Amazon Q untuk memperluas opsi otentikasi Amazon Q Developer.

#### Note

Jika tidak ada kredensial tersimpan, lanjutkan ke Langkah 3 untuk menambah atau memperbarui kredensial Pusat Identitas IAM Anda.

2. Dari bagian My organization has enabled Amazon Q Developer, perluas menu drop-down Pilih dari Profil yang ada atau tambahkan baru untuk memilih dari daftar kredensial yang disimpan.
3. Dari menu tarik-turun Jenis Profil, pilih AWS IAM Identity Center
4. Di bidang teks Nama Profil, masukkan profil Pusat **Profile Name** Identitas IAM yang ingin Anda autentikasi.
5. Di bidang teks URL Mulai, masukkan **Start URL** yang dilampirkan ke kredensial Pusat Identitas IAM Anda.
6. Dari menu tarik-turun Wilayah Profil (default ke us-east-1), pilih Wilayah Profil yang ditentukan oleh profil pengguna Pusat Identitas IAM yang Anda autentikasi.

7. Dari menu drop-down Wilayah SSO (default ke us-east-1), pilih Wilayah SSO yang ditentukan oleh kredensial Pusat Identitas IAM Anda, lalu pilih tombol Connect untuk membuka dialog Login with IAM Identity Center. AWS
8. Dari dialog Masuk dengan Pusat AWS Identitas IAM, pilih tombol Lanjutkan ke Browser untuk membuka situs permintaan AWS Otorisasi di browser web default Anda.
9. Konfirmasikan kode keamanan di IDE Anda cocok dengan kode konfirmasi permintaan AWS Otorisasi yang ditampilkan di browser web Anda dan pilih tombol Kirim dan lanjutkan untuk melanjutkan.
10. Ikuti petunjuk di browser web default Anda, Anda diberi tahu ketika proses otorisasi selesai, aman untuk menutup browser Anda, dan kembali ke Visual Studio.

### Mengautentikasi dan terhubung dengan AWS Builder ID

1. Dari Memulai dengan UI koneksi AWS Toolkit, pilih radial Pengembang Amazon Q untuk memperluas opsi otentikasi Amazon Q Developer.
2. Dari Saya menggunakan Pengembang Amazon Q di bagian saya sendiri, pilih tombol Daftar atau Masuk untuk membuka dialog Masuk dengan ID AWS Pembangun.
3. Pilih tombol Lanjutkan ke Browser untuk membuka situs permintaan AWS Otorisasi di browser web default Anda.
4. Konfirmasikan kode keamanan di IDE Anda cocok dengan kode konfirmasi permintaan AWS Otorisasi yang ditampilkan di browser web Anda dan pilih tombol Kirim dan lanjutkan untuk melanjutkan.
5. Ikuti petunjuk di browser web default Anda, Anda diberi tahu ketika proses otorisasi selesai, aman untuk menutup browser Anda, dan kembali ke Visual Studio.

## Otentikasi untuk Explorer AWS

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

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

## Mengautentikasi dan terhubung dengan IAM Identity Center

1. Dari Memulai dengan UI koneksi AWS Toolkit, pilih radial AWS Explorer untuk memperluas opsi otentikasi Amazon Q Developer.
2. Dari menu **Profile Type** tarik-turun, pilih AWS IAM Identity Center.
3. Di bidang teks Nama Profil, masukkan profil Pusat Identitas IAM yang ingin Anda gunakan.  
**Profile Name**
4. Di bidang teks URL Mulai, masukkan **Start URL** yang dilampirkan ke kredensial Pusat Identitas IAM Anda.
5. Dari menu tarik-turun Wilayah Profil (default ke us-east-1), pilih Wilayah Profil yang ditentukan oleh profil pengguna Pusat Identitas IAM yang Anda autentikasi.
6. Dari menu tarik-turun Wilayah SSO (default ke us-east-1), pilih Wilayah SSO yang ditentukan oleh kredensial Pusat Identitas IAM Anda.
7. Pilih tombol Lanjutkan ke browser untuk membuka situs permintaan AWS Otorisasi di browser web default Anda.
8. Konfirmasikan kode keamanan di IDE Anda cocok dengan kode konfirmasi permintaan AWS Otorisasi yang ditampilkan di browser web Anda dan pilih tombol Kirim dan lanjutkan untuk melanjutkan.
9. Ikuti petunjuk di browser web default Anda, Anda diberi tahu ketika proses otorisasi selesai, aman untuk menutup browser Anda, dan kembali ke Visual Studio.

## Mengautentikasi dan terhubung dengan Kredensial IAM

1. Dari Memulai dengan UI koneksi AWS Toolkit, pilih radial AWS Explorer untuk memperluas opsi otentikasi Amazon Q Developer.
2. Dari menu **Profile Type** tarik-turun, pilih Peran Pengguna IAM.
3. Di bidang teks Nama Profil, masukkan profil **Profile Name** yang ingin Anda autentikasi.
4. Di bidang teks ID Kunci Akses, masukkan profil **Access Key ID** yang ingin Anda autentikasi.
5. Di bidang teks Kunci Rahasia, masukkan **Secret Key** untuk profil yang ingin Anda autentikasi.
6. Dari menu drop-down Storage Location (default ke Shared Credentials File), tentukan apakah Anda ingin menyimpan kredensialnya dengan file Shared Credentials atau dengan.NET Encrypted Stored.
7. Dari menu tarik-turun Wilayah Profil (default ke us-east-1), pilih Wilayah Profil yang dilampirkan ke profil yang ingin Anda autentikasi.

# Memecahkan masalah instalasi AWS Toolkit for Visual Studio

Informasi berikut diketahui untuk menyelesaikan masalah instalasi umum saat menginstal AWS Toolkit for Visual Studio.

Jika Anda mengalami kesalahan saat menginstal AWS Toolkit for Visual Studio atau tidak jelas apakah instalasi selesai atau tidak, tinjau informasi di masing-masing bagian berikut.

## Izin administrator untuk Visual Studio

AWS Toolkit for Visual Studio Ekstensi memerlukan izin administrator untuk memastikan bahwa semua AWS layanan dan fitur dapat diakses.

Jika Anda memiliki izin administrator lokal, izin administrator Anda tidak meluas langsung ke instance Visual Studio Anda.

Untuk meluncurkan Visual Studio dengan izin administrator secara lokal:

1. Dari Windows, cari peluncur aplikasi Visual Studio (ikon).
2. Buka menu konteks untuk (klik kanan) ikon Visual Studio untuk membuka menu konteks.
3. Pilih Jalankan sebagai administrator dari menu konteks.

Untuk meluncurkan Visual Studio dengan izin administrator dari jarak jauh:

1. Dari Windows, cari peluncur aplikasi untuk aplikasi yang Anda gunakan untuk menyambung ke instance Visual Studio jarak jauh Anda.
2. Buka menu konteks untuk (klik kanan) aplikasi untuk membuka menu konteks.
3. Pilih Jalankan sebagai administrator dari menu konteks.

### Note

Apakah Anda meluncurkan program secara lokal atau terhubung dari jarak jauh, Windows mungkin meminta Anda untuk mengonfirmasi kredensi administratif Anda.

## Mendapatkan log instalasi

Jika Anda telah menyelesaikan langkah-langkah di bagian izin Administrator sebelumnya yang terletak di atas dan dikonfirmasi bahwa Anda menjalankan atau menyambung ke Visual Studio dengan izin administrator, maka mendapatkan file log instalasi dapat membantu mendiagnosis masalah lain.

Untuk menginstal file AWS Toolkit for Visual Studio dari `.vsix` file secara manual dan menghasilkan file log instalasi, selesaikan langkah-langkah berikut.

1. Dari halaman [AWS Toolkit for Visual Studio](#) arahan, ikuti tautan Unduh dan simpan `.vsix` file AWS Toolkit for Visual Studio versi yang ingin Anda instal.
2. Dari menu utama Visual Studio, perluas header Tools, perluas sub menu Command Line, lalu pilih Visual Studio Developer Command Prompt.
3. Dari Visual Studio Developer Command Prompt masukkan `vsixinstaller` perintah dengan format berikut:

```
vsixinstaller /logfile:[file path to log file] [file path to Toolkit installation file]
```

4. Ganti `[file path to log file]` dengan nama file dan path file lengkap dari direktori tempat Anda ingin log instalasi dibuat. Contoh `vsixinstaller` perintah dengan path file yang Anda tentukan dan nama file menyerupai berikut ini:

```
vsixinstaller /logfile:C:\Users\Documents\install-log.txt [file path to AWSToolkitPackage.vsix]
```

5. Ganti `[file path to Toolkit installation file]` dengan path file lengkap dari direktori tempat `AWSToolkitPackage.vsix` berada.

Contoh `vsixinstaller` perintah dengan path file lengkap ke file instalasi Toolkit harus menyerupai berikut ini:

```
vsixinstaller /logfile:[file path to log file] C:\Users\Downloads\AWSToolkitPackage.vsix
```

6. Periksa untuk memastikan nama file dan path Anda benar, lalu jalankan `vsixinstaller` perintah.

Contoh `vsixinstaller` perintah lengkap menyerupai berikut ini:

```
vsixinstaller /logfile:C:\Users\Documents\install-log.txt C:\Users  
\Downloads\AWSToolkitPackage.vsix
```

## Menginstal ekstensi Visual Studio yang berbeda

Jika Anda telah memperoleh file log instalasi dan Anda masih tidak dapat menentukan mengapa proses instalasi gagal, periksa untuk melihat apakah Anda dapat menginstal ekstensi Visual Studio lainnya. Menginstal ekstensi Visual Studio yang berbeda dapat memberikan wawasan tambahan untuk masalah instalasi Anda. Jika Anda tidak dapat menginstal ekstensi Visual Studio, mungkin perlu untuk memecahkan masalah dengan Visual Studio, bukan. AWS Toolkit for Visual Studio

## Menghubungi dukungan

Jika Anda telah meninjau semua bagian yang terdapat dalam panduan ini dan memerlukan sumber daya atau dukungan tambahan, Anda dapat melihat masalah sebelumnya atau membuka masalah baru dari situs Masalah [AWS Toolkit for Visual StudioGithub](#).

Untuk membantu mempercepat solusi untuk masalah Anda:

- Periksa masalah masa lalu dan saat ini untuk melihat apakah orang lain mengalami situasi serupa.
- Simpan catatan terperinci tentang setiap langkah yang telah Anda ambil untuk mengatasi masalah ini.
- Simpan file log apa pun yang Anda peroleh dari menginstal AWS Toolkit for Visual Studio atau ekstensi lainnya.
- Lampirkan file log AWS Toolkit for Visual Studio instalasi Anda ke masalah baru.

## Profil dan Window Binding

### Profil dan Window Binding for Toolkit for Visual Studio

Saat bekerja dengan alat penerbitan, wizard, dan fitur lain dari Toolkit for Visual Studio, perhatikan hal berikut:

- JendelaAWS Explorer terikat ke satu profil dan wilayah pada satu waktu. Windows dibuka dari defaultAWS Explorer ke profil dan wilayah terikat itu.



- Setelah jendela baru dibuka, Anda dapat menggunakan instance AWS Explorer untuk beralih ke profil atau wilayah yang berbeda.
- Toolkit untuk alat penerbitan Visual Studio dan fitur secara otomatis default ke profil dan wilayah diatur dalam AWS Explorer.
- Jika profil atau wilayah baru ditentukan dalam alat penerbitan, wizard, atau fitur: semua sumber daya yang dibuat setelahnya akan terus menggunakan pengaturan profil dan wilayah baru.
- Jika Anda memiliki beberapa contoh Visual Studio terbuka, setiap contoh dapat terikat ke profil yang berbeda dan wilayah.
- AWS Explorer menyimpan profil terakhir dan wilayah yang ditentukan dan contoh Visual Studio terakhir ditutup akan memiliki nilai-nilai tetap.

# Otentikasi dan akses

Anda tidak perlu mengautentikasi AWS untuk mulai bekerja dengan AWS Toolkit for Visual Studio. Namun, sebagian besar AWS sumber daya dikelola melalui AWS akun. Untuk mengakses semua layanan dan fitur AWS Toolkit for Visual Studio, Anda memerlukan setidaknya 2 jenis otentikasi akun:

1. Baik AWS Identity and Access Management (IAM) atau AWS IAM Identity Center otentikasi untuk akun Anda AWS . Sebagian besar AWS layanan dan sumber daya dikelola melalui IAM dan IAM Identity Center.
2. AWS Builder ID adalah opsional untuk AWS layanan tertentu lainnya.

Topik berikut berisi rincian tambahan dan mengatur instruksi untuk setiap jenis kredensi dan metode otentikasi.

Topik

- [AWS Identitas Pusat Identitas IAM di AWS Toolkit for Visual Studio](#)
- [AWS Kredensyal IAM](#)
- [AWS ID Pembangun](#)
- [Otentikasi multi-faktor \(MFA\) di Toolkit for Visual Studio](#)
- [Menyiapkan kredensyal eksternal](#)

## AWS Identitas Pusat Identitas IAM di AWS Toolkit for Visual Studio

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

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

# Mengautentikasi dengan IAM Identity Center dari AWS Toolkit for Visual Studio

Untuk mengautentikasi dengan IAM Identity Center dari AWS Toolkit for Visual Studio dengan menambahkan profil IAM Identity Center ke config file `credentials` atau Anda, selesaikan langkah-langkah berikut.

1. Dari editor teks pilihan Anda, buka informasi AWS kredensial yang disimpan dalam file.  
`<home-directory>\.aws\credentials`
2. Dari bagian `credentials` file bawah[default], tambahkan template untuk profil Pusat Identitas IAM bernama. Berikut ini adalah contoh template:

## Important

Jangan gunakan profil kata saat membuat entri dalam `credential` file karena membuat konflik dengan konvensi penamaan `credential` file.

Sertakan kata awalan `profile_` hanya saat mengonfigurasi profil bernama dalam file. `config`

```
[sso-user-1]
sso_start_url = https://example.com/start
sso_region = us-east-2
sso_account_id = 123456789011
sso_role_name = readOnly
region = us-west-2
```

- **sso\_start\_url**: URL yang mengarah ke portal pengguna Pusat Identitas IAM organisasi Anda.
- **sso\_region**: AWS Wilayah yang berisi host portal Pusat Identitas IAM Anda. Ini bisa berbeda dari AWS Wilayah yang ditentukan nanti dalam `region` parameter default.
- **sso\_account\_id**: ID AWS akun yang berisi peran IAM dengan izin yang ingin Anda berikan kepada pengguna Pusat Identitas IAM ini.
- **sso\_role\_name**: Nama peran IAM yang menentukan izin pengguna saat menggunakan profil ini untuk mendapatkan kredensial melalui IAM Identity Center.
- **region**: AWS Wilayah default tempat pengguna IAM Identity Center ini masuk.

**Note**

Anda juga dapat menambahkan profil yang diaktifkan Pusat Identitas IAM ke profil Anda AWS CLI dengan menjalankan `aws configure sso` perintah. Setelah menjalankan perintah ini, Anda memberikan nilai untuk URL awal Pusat Identitas IAM (`sso_start_url`) dan AWS Region (`region`) yang menghosting direktori IAM Identity Center.

Untuk informasi selengkapnya, lihat [Mengonfigurasi AWS CLI untuk AWS menggunakan Single Sign-On](#) di Panduan Pengguna AWS Command Line Interface

## Masuk dengan IAM Identity Center

Saat masuk dengan profil Pusat Identitas IAM, browser default diluncurkan ke yang `sso_start_url` ditentukan di profil `Andacredential` file. Anda harus memverifikasi login IAM Identity Center Anda sebelum Anda dapat mengakses AWS sumber daya Anda di AWS Toolkit for Visual Studio. Jika kredensial Anda kedaluwarsa, Anda harus mengulangi proses koneksi untuk mendapatkan kredensial sementara yang baru.

## AWS Kredensial IAM

AWS Kredensial IAM mengautentikasi dengan AWS akun Anda melalui kunci akses yang disimpan secara lokal.

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

**Important**

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

- Jika Anda telah menyetel kredensial IAM melalui AWS layanan lain (seperti AWS CLI), maka AWS Toolkit for Visual Studio secara otomatis mendeteksi kredensial tersebut.
- AWS merekomendasikan menggunakan AWS IAM Identity Center otentikasi. 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.

Sebaliknya, gunakan federasi dengan penyedia identitas seperti AWS IAM Identity Center. Untuk informasi lebih lanjut lihat [Apa itu Pusat Identitas IAM?](#) dalam AWS IAM Identity Center User Guide.

## Membuat pengguna IAM

Sebelum Anda dapat mengatur AWS Toolkit for Visual Studio 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 AWS SDK and Tools Reference Guide.

### Note

Langkah 3: Perbarui kredensial bersama adalah opsional.

Jika Anda menyelesaikan Langkah 3, secara AWS Toolkit for Visual Studio otomatis mendeteksi kredensi Anda dari `credentials file`

Jika Anda belum menyelesaikan Langkah 3, AWS Toolkit for Visual Studio memandu Anda melalui proses membuat `credentials file` seperti yang dijelaskan dalam [Membuat file kredensial dari AWS Toolkit for Visual Studio bagian, yang terletak di bawah ini](#).

## Membuat file kredensial

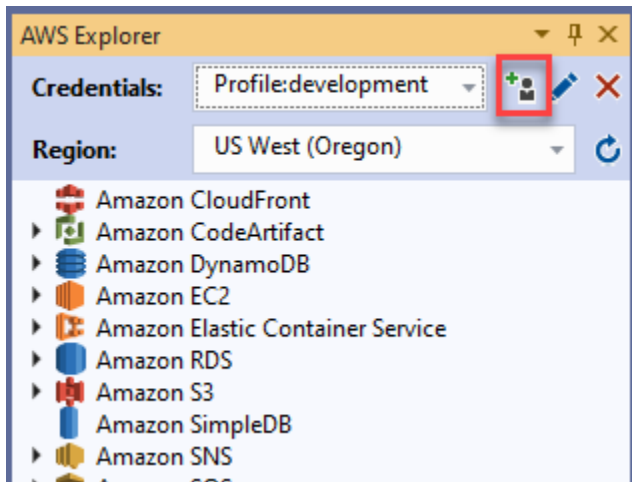
Untuk menambahkan pengguna ke atau membuat `credentials file` dari AWS Toolkit for Visual Studio:

### Note

Ketika profil pengguna baru ditambahkan dari toolkit:

- Jika `credentials file` sudah ada, informasi pengguna baru ditambahkan ke file yang ada.
- Jika `credentials file` tidak ada file baru dibuat.

1. Dari AWS Explorer pilih ikon Profil Akun Baru untuk membuka dialog Profil Akun Baru.



2. Lengkapi kolom wajib di dialog Profil Akun Baru dan pilih tombol OK untuk membuat pengguna IAM.

## Mengedit kredensial pengguna IAM dari toolkit

Untuk mengedit kredensial pengguna IAM dari toolkit, selesaikan langkah-langkah berikut:

1. Dari drop-down Credentials di AWS Explorer, pilih kredensi pengguna IAM yang ingin Anda edit.
2. Pilih ikon Edit Profil untuk membuka dialog Edit Profil.
3. Dari dialog Edit Profil, selesaikan pembaruan Anda dan pilih OK tombol untuk menyimpan perubahan Anda.

Untuk menghapus kredensial pengguna IAM dari toolkit, selesaikan langkah-langkah berikut:

1. Dari tarik-turun Kredensial di AWS Explorer, pilih kredensi pengguna IAM yang ingin Anda hapus.
2. Pilih ikon Hapus Profil untuk membuka prompt Hapus Profil.
3. Konfirmasikan bahwa Anda ingin menghapus profil untuk menghapusnya dari profil AndaCredentials file.

### Important

Profil yang mendukung fitur akses lanjutan, seperti Pusat Identitas IAM atau otentikasi Multi-faktor (MFA) dalam dialog Edit Profil, tidak dapat diedit dari file. AWS Toolkit for Visual Studio

Untuk membuat perubahan pada jenis profil ini, Anda harus mengedit `credentials` file menggunakan editor teks.

## Mengedit kredensial pengguna IAM dari editor teks

Selain mengelola pengguna IAM dengan AWS Toolkit for Visual Studio, Anda dapat mengedit `credential` files dari editor teks pilihan Anda. Lokasi default `credential` file di Windows adalah `C:\Users\USERNAME\.aws\credentials`.

Untuk detail selengkapnya tentang lokasi dan struktur `credential` files, lihat bagian [File konfigurasi dan kredensial bersama](#) pada panduan Referensi AWS SDK dan Alat.

## Membuat pengguna IAM dari AWS Command Line Interface (AWS CLI)

AWS CLI ini adalah alat lain yang dapat Anda gunakan untuk membuat pengguna IAM di `credentials` file, menggunakan perintah `aws configure`.

Untuk informasi rinci tentang membuat pengguna IAM dari AWS CLI lihat [Mengkonfigurasi AWS CLI topik dalam AWS CLI](#) Panduan Pengguna.

Toolkit for Visual Studio mendukung properti konfigurasi berikut:

```
aws_access_key_id
aws_secret_access_key
aws_session_token
credential_process
credential_source
external_id
mfa_serial
role_arn
role_session_name
source_profile
sso_account_id
sso_region
sso_role_name
sso_start_url
```

## AWS ID Pembangun

AWS Builder ID adalah metode AWS otentikasi tambahan yang mungkin diperlukan untuk menggunakan layanan atau fitur tertentu, seperti mengkloning repositori pihak ketiga dengan Amazon. CodeCatalyst

Untuk informasi lebih lanjut tentang metode autentikasi AWS Builder ID, lihat topik [Masuk dengan AWS Builder ID](#) di Panduan Pengguna AWS Masuk.

Untuk informasi tambahan tentang mengkloning repositori CodeCatalyst dari AWS Toolkit for Visual Studio, lihat CodeCatalyst topik Bekerja [dengan Amazon](#) di Panduan Pengguna ini.

## Otentikasi multi-faktor (MFA) di Toolkit for Visual Studio

Otentikasi multi-faktor (MFA) adalah keamanan tambahan untuk akun Anda. AWS MFA mengharuskan pengguna untuk memberikan kredensi masuk dan otentikasi unik dari mekanisme MFA yang AWS didukung saat mengakses situs web atau layanan. AWS

AWS mendukung berbagai perangkat virtual dan perangkat keras untuk otentikasi MFA. Berikut ini adalah contoh perangkat MFA virtual yang diaktifkan melalui aplikasi smartphone. Untuk informasi selengkapnya tentang opsi perangkat MFA, lihat [Menggunakan otentikasi multi-faktor \(MFA\) AWS](#) di Panduan Pengguna IAM.

### Langkah 1: Membuat peran IAM untuk mendelegasikan akses ke pengguna IAM

Prosedur berikut menjelaskan cara mengatur delegasi peran untuk menetapkan izin ke pengguna IAM. Untuk informasi rinci tentang delegasi peran, lihat [Membuat peran untuk mendelegasikan izin ke topik pengguna IAM di Panduan Pengguna](#).AWS Identity and Access Management

1. Buka konsol IAM di <https://console.aws.amazon.com/iam>.
2. Pilih Peran di bilah navigasi, lalu pilih Buat Peran.
3. Di halaman Buat peran, pilih AWS Akun lain.
4. Masukkan ID Akun yang Anda butuhkan dan tandai kotak centang Memerlukan MFA.



**Note**

Untuk menemukan 12 digit nomor akun (ID) Anda, buka bilah navigasi di konsol, lalu pilih Support, Support Center.

5. Pilih Berikutnya: Izin.
6. Lampirkan kebijakan yang ada ke peran Anda atau buat kebijakan baru untuknya. Kebijakan yang Anda pilih di halaman ini menentukan AWS layanan mana yang dapat diakses pengguna IAM dengan Toolkit.
7. Setelah melampirkan kebijakan, pilih Berikutnya: Tag untuk opsi menambahkan tag IAM ke peran Anda. Kemudian pilih Berikutnya: Tinjau untuk melanjutkan.
8. Di halaman Tinjauan, masukkan nama Peran yang diperlukan (toolkit-role, misalnya). Anda juga dapat menambahkan deskripsi Peran opsional.
9. Pilih Buat peran.
10. Ketika pesan konfirmasi ditampilkan (“Peran toolkit-peran telah dibuat”, misalnya), pilih nama peran dalam pesan.
11. Di halaman Ringkasan, pilih ikon salin untuk menyalin ARN Peran dan tempelkan ke dalam file. (Anda memerlukan ARN ini saat mengonfigurasi pengguna IAM untuk mengambil peran.).

## Langkah 2: Membuat pengguna IAM yang mengasumsikan izin peran

Langkah ini membuat pengguna IAM tanpa izin sehingga kebijakan in-line dapat ditambahkan.

1. Buka konsol IAM di <https://console.aws.amazon.com/iam>.
2. Pilih Pengguna di bilah navigasi dan kemudian pilih Tambah pengguna.
3. Di halaman Tambah pengguna, masukkan nama pengguna yang diperlukan (toolkit-user, misalnya) dan tandai kotak centang Akses program.
4. Pilih Berikutnya: Izin, Berikutnya: Tag, dan Berikutnya: Tinjau untuk bergerak melalui halaman berikutnya. Anda tidak menambahkan izin pada tahap ini karena pengguna akan mengambil izin peran.
5. Di halaman Tinjauan, Anda diberi tahu bahwa Pengguna ini tidak memiliki izin. Pilih Create user (Buat pengguna).

6. Di halaman Sukses, pilih Unduh.csv untuk mengunduh file yang berisi ID kunci akses dan kunci akses rahasia. (Anda memerlukan keduanya saat menentukan profil pengguna di file kredensial.)
7. Pilih Tutup.

### Langkah 3: Menambahkan kebijakan untuk memungkinkan pengguna IAM mengambil peran

Prosedur berikut membuat kebijakan in-line yang memungkinkan pengguna untuk mengambil peran (dan izin peran tersebut).

1. Di halaman Pengguna konsol IAM, pilih pengguna IAM yang baru saja Anda buat (toolkit-user, misalnya).
2. Di tab Izin pada halaman Ringkasan, pilih Tambahkan kebijakan sebaris.
3. Di halaman Buat kebijakan, pilih Pilih layanan, masukkan STS di Temukan layanan, lalu pilih STS dari hasilnya.
4. Untuk Tindakan, mulailah memasukkan istilah AssumeRole. Tandai kotak AssumeRolecentang saat muncul.
5. Di bagian Sumber Daya, pastikan Spesifik dipilih, dan klik Tambahkan ARN untuk membatasi akses.
6. Dalam Tambahkan ARN kotak dialog, untuk Tentukan ARN untuk peran tambahkan ARN dari peran yang Anda buat di Langkah 1.

Setelah Anda menambahkan ARN peran, akun tepercaya dan nama peran yang terkait dengan peran tersebut akan ditampilkan di Akun dan nama Peran dengan jalur.

7. Pilih Tambahkan.
8. Kembali ke halaman Buat kebijakan, pilih Tentukan kondisi permintaan (opsional), tandai kotak centang MFA wajib, lalu pilih dekat untuk mengonfirmasi..
9. Pilih Tinjau kebijakan
10. Di halaman Kebijakan tinjauan, masukkan Nama untuk kebijakan, lalu pilih Buat kebijakan.

Tab Izin menampilkan kebijakan inline baru yang dilampirkan langsung ke pengguna IAM.

## Langkah 4: Mengelola perangkat MFA virtual untuk pengguna IAM

1. Unduh dan instal aplikasi MFA virtual ke ponsel cerdas Anda.

Untuk daftar aplikasi yang didukung, lihat halaman sumber daya [Otentikasi Multi-faktor](#).

2. Di konsol IAM, pilih Pengguna dari bilah navigasi dan kemudian pilih pengguna yang mengambil peran (toolkit-user, dalam hal ini).
3. Di halaman Ringkasan, pilih tab Security credentials, dan untuk perangkat MFA yang Ditugaskan pilih Kelola.
4. Di panel Kelola perangkat MFA, pilih Perangkat MFA virtual, lalu pilih Lanjutkan.
5. Di panel Siapkan perangkat MFA virtual, pilih Tampilkan kode QR dan kemudian pindai kode menggunakan aplikasi MFA virtual yang Anda instal di ponsel cerdas Anda.
6. Setelah Anda memindai kode QR, aplikasi MFA virtual menghasilkan kode MFA satu kali. Masukkan dua kode MFA berturut-turut dalam kode MFA 1 dan kode MFA 2.
7. Pilih Tugaskan MFA.
8. Kembali ke tab Security credentials untuk pengguna, salin ARN dari perangkat MFA yang Ditugaskan baru.

ARN menyertakan ID akun 12 digit Anda dan formatnya mirip dengan yang berikut:

`arn:aws:iam::123456789012:mfa/toolkit-user` Anda memerlukan ARN ini saat mendefinisikan profil MFA di langkah berikutnya.

## Langkah 5: Membuat profil untuk memungkinkan MFA

Prosedur berikut membuat profil yang memungkinkan MFA saat mengakses AWS layanan dari Toolkit for Visual Studio.

Profil yang Anda buat mencakup tiga bagian informasi yang telah Anda salin dan simpan selama langkah-langkah sebelumnya:

- Kunci akses (ID kunci akses dan kunci akses rahasia) untuk pengguna IAM
- ARN dari peran yang mendelegasikan izin ke pengguna IAM
- ARN perangkat MFA virtual yang ditetapkan untuk pengguna IAM

Di file kredensial AWS bersama atau SDK Store yang berisi AWS kredensial Anda, tambahkan entri berikut:

```
[toolkit-user]
aws_access_key_id = AKIAIOSFODNN7EXAMPLE
aws_secret_access_key = wJalrXUtnFEMI/K7MDENG/bPxrFiCYEXAMPLEKEY

[mfa]
source_profile = toolkit-user
role_arn = arn:aws:iam::111111111111:role/toolkit-role
mfa_serial = arn:aws:iam::111111111111:mfa/toolkit-user
```

Ada dua profil yang didefinisikan dalam contoh yang diberikan:

- [toolkit-user] profil termasuk kunci akses dan kunci akses rahasia yang dihasilkan dan disimpan saat Anda membuat pengguna IAM di Langkah 2.
- [mfa] profil mendefinisikan bagaimana otentikasi multi-faktor didukung. Ada tiga entri:
  - `source_profile`: Menentukan profil yang kredensialnya digunakan untuk mengambil peran yang ditentukan oleh `role_arn` pengaturan ini dalam profil ini. Dalam hal ini, itu adalah `toolkit-user` profil.
  - `role_arn`: Menentukan Nama Sumber Daya Amazon (ARN) dari peran IAM yang ingin Anda gunakan untuk melakukan operasi yang diminta menggunakan profil ini. Dalam hal ini, ini adalah ARN untuk peran yang Anda buat di Langkah 1.
  - `mfa_serial`: Menentukan identifikasi atau nomor seri perangkat MFA yang harus digunakan pengguna saat mengambil peran. Dalam hal ini, itu adalah ARN dari perangkat virtual yang Anda atur di Langkah 3.

## Menyiapkan kredensial eksternal

Jika Anda memiliki metode untuk menghasilkan atau mencari kredensial yang tidak didukung secara langsung AWS, Anda dapat menambahkan ke file kredensi bersama profil yang berisi setelan `credential_process`. Pengaturan ini menentukan perintah eksternal yang dijalankan untuk menghasilkan atau mengambil kredensial otentikasi untuk digunakan. Misalnya, Anda mungkin menyertakan entri yang mirip dengan yang berikut ini dalam `config` file:

```
[profile developer]
credential_process = /opt/bin/awscreds-custom --username helen
```

Untuk informasi selengkapnya tentang penggunaan kredensial eksternal dan risiko keamanan terkait, lihat [Sumber kredensial dengan proses eksternal](#) di Panduan Pengguna.AWS Command Line Interface

# Bekerja dengan AWS Layanan

Topik berikut menjelaskan cara memulai bekerja dengan AWS layanan dari AWS Toolkit for Visual Studio.

## Topik

- [Amazon CodeCatalyst untuk AWS Toolkit untuk Visual Studio](#)
- [Amazon CloudWatch Integrasi log untuk Visual Studio](#)
- [Mengelola Instans Amazon EC2](#)
- [Mengelola instans Amazon ECS](#)
- [Mengelola Grup Keamanan dariAWSPenjelajah](#)
- [Membuat AMI dari Instans Amazon EC2](#)
- [Menyiapkan Izin Luncurkan pada Amazon Machine Image](#)
- [Amazon Virtual Private Cloud \(VPC\)](#)
- [Menggunakan Editor AWS CloudFormation Template untuk Visual Studio](#)
- [Menggunakan Amazon S3 dariAWSPenjelajah](#)
- [Menggunakan DynamoDB dariAWSPenjelajah](#)
- [MenggunakanAWS CodeCommitdengan Visual Studio Team Explorer](#)
- [Menggunakan CodeArtifact di Visual Studio](#)
- [Amazon RDS dariAWSPenjelajah](#)
- [Menggunakan Amazon SimpleDB dariAWSPenjelajah](#)
- [Menggunakan Amazon SQSAWSPenjelajah](#)
- [Manajemen Identitas dan Akses](#)
- [AWS Lambda](#)

## Amazon CodeCatalyst untuk AWS Toolkit untuk Visual Studio

### Apa yang dimaksud dengan Amazon CodeCatalyst?

Amazon CodeCatalyst adalah ruang kolaborasi berbasis cloud untuk tim pengembangan perangkat lunak. Menggunakan AWS Toolkit untuk Visual Studio, Anda dapat melihat dan mengelola

CodeCatalyst sumber daya langsung dari AWS Toolkit untuk Visual Studio. Untuk informasi selengkapnya CodeCatalyst, lihat Panduan CodeCatalyst Pengguna [Amazon](#).

Topik berikut menjelaskan cara menghubungkan AWS Toolkit untuk Visual Studio dengan CodeCatalyst dan bagaimana bekerja dengan CodeCatalyst melalui AWS Toolkit untuk Visual Studio.

Topik

- [Memulai dengan Amazon CodeCatalyst dan AWS Toolkit untuk Visual Studio](#)
- [Bekerja dengan CodeCatalyst sumber daya Amazon dari AWS Toolkit untuk Visual Studio](#)
- [Pemecahan Masalah](#)

## Memulai dengan Amazon CodeCatalyst dan AWS Toolkit untuk Visual Studio

Untuk mulai bekerja dengan Amazon CodeCatalyst dari AWS Toolkit untuk Visual Studio, selesaikan hal berikut.

Topik

- [Menginstal AWS Toolkit untuk Visual Studio](#)
- [Membuat CodeCatalyst akun dan AWS Builder ID](#)
- [Menghubungkan AWS Toolkit untuk Visual Studio dengan CodeCatalyst](#)

## Menginstal AWS Toolkit untuk Visual Studio

Sebelum Anda mengintegrasikan AWS Toolkit untuk Visual Studio dengan CodeCatalyst account Anda, pastikan bahwa Anda menggunakan versi AWS Toolkit untuk Visual Studio saat ini. Untuk detail tentang cara menginstal dan mengatur versi terbaru AWS Toolkit untuk Visual Studio, lihat bagian [Menyiapkan AWS Toolkit untuk Visual Studio](#) dari Panduan Pengguna ini.

## Membuat CodeCatalyst akun dan AWS Builder ID

Selain menginstal versi terbaru dari AWS Toolkit untuk Visual Studio, Anda harus memiliki AWS Builder ID aktif dan CodeCatalyst account untuk terhubung dengan AWS Toolkit untuk Visual Studio. Jika Anda tidak memiliki ID atau CodeCatalyst akun AWS Builder aktif, lihat CodeCatalyst bagian [Pengaturan dengan](#) di Panduan CodeCatalyst Pengguna.

**Note**

ID AWS Builder berbeda dari AWS Kredensial Anda. Untuk petunjuk tentang cara mendaftar dan mengautentikasi dengan ID AWS Pembuat, lihat topik [Authentication and access: AWS Builder ID](#) di Panduan Pengguna ini.

Untuk informasi selengkapnya tentang ID AWS Pembuat, lihat topik [ID AWS Pembuat](#) di Panduan Pengguna Referensi AWS Umum.

## Menghubungkan AWS Toolkit untuk Visual Studio dengan CodeCatalyst

Untuk menghubungkan AWS Toolkit untuk Visual Studio dengan CodeCatalyst akun Anda, selesaikan langkah-langkah berikut.

1. Dari item menu Git di Visual Studio, pilih Clone Repository... .
2. Dari bagian Jelajahi Repositori, pilih Amazon CodeCatalyst sebagai penyedia.
3. Dari bagian Koneksi, pilih Hubungkan dengan ID AWS Pembuat untuk membuka CodeCatalyst konsol di browser web pilihan Anda.
4. Dari browser Anda, masukkan ID AWS Builder Anda ke kolom yang disediakan dan ikuti petunjuk untuk melanjutkan.
5. Saat diminta, pilih Izinkan untuk mengonfirmasi koneksi antara AWS Toolkit untuk Visual Studio dan akun AndaCodeCatalyst. Ketika proses koneksi selesai, CodeCatalyst menampilkan konfirmasi yang menunjukkan bahwa aman untuk menutup browser Anda.

## Bekerja dengan CodeCatalyst sumber daya Amazon dari AWS Toolkit untuk Visual Studio

Bagian berikut memberikan ikhtisar fitur pengelolaan CodeCatalyst sumber daya Amazon Amazon yang tersedia untuk AWS Toolkit untuk Visual Studio.

### Topik

- [Kloning repositori](#)



## Kloning repositori

CodeCatalyst adalah layanan berbasis cloud yang mengharuskan Anda terhubung ke cloud untuk CodeCatalyst mengerjakan proyek. Untuk mengerjakan proyek secara lokal, Anda dapat mengkloning CodeCatalyst repositori ke mesin lokal Anda dan menyinkronkan dengan CodeCatalyst proyek Anda saat berikutnya Anda terhubung ke cloud.

Untuk mengkloning repositori ke mesin lokal Anda, selesaikan langkah-langkah berikut.

1. Dari item menu Git di Visual Studio, pilih Clone Repository... .
2. Dari bagian Jelajahi Repositori, pilih Amazon CodeCatalyst sebagai penyedia.

### Note

Jika bagian Koneksi menampilkan Not Connected pesan, selesaikan langkah-langkah di bagian [Authentication and access: AWS Builder ID](#) dari Panduan Pengguna ini sebelum melanjutkan.

3. Pilih Space dan Project yang ingin Anda kloning repositori.
4. Dari bagian Repositori, pilih repositori yang ingin Anda kloning.
5. Dari bagian Path, pilih folder yang ingin Anda kloning repositori Anda.

### Note

Folder ini awalnya harus kosong untuk mengkloning berhasil.

6. Pilih Clone untuk memulai kloning repositori.
7. Setelah repositori telah dikloning, Visual Studio akan memuat solusi kloning Anda

### Note

Jika Visual Studio tidak membuka solusi di repositori kloning, opsi Visual Studio Anda dapat disesuaikan dari otomatis memuat solusi ketika membuka pengaturan repositori Git, yang terletak di Pengaturan Global Git, dari menu Source Control.

## Pemecahan Masalah

Berikut ini adalah topik pemecahan masalah untuk mengatasi masalah yang diketahui saat bekerja dengan Amazon CodeCatalyst dari AWS Toolkit untuk Visual Studio.

Topik

- [Kredensial](#)

### Kredensial

Jika Anda menemukan dialog yang meminta kredensial saat mencoba mengkloning repositori berbasis git, pembantu AWS CodeCommit Kredensial Anda dapat dikonfigurasi secara globalCodeCatalyst, yang menyebabkan gangguan. CodeCatalyst Untuk informasi tambahan tentang penolong AWS CodeCommit kredensi, lihat bagian [Menyiapkan langkah-langkah untuk koneksi HTTPS ke AWS CodeCommit repositori di Windows dengan bagian pembantu AWS kredensi CLI](#) di Panduan Pengguna. AWSCodeCommit

Untuk membatasi AWSCodeCommitCredential helper hanya menangani CodeCommit URL, selesaikan langkah-langkah berikut.

1. buka file konfigurasi git global di: %userprofile%\ .gitconfig
2. Temukan bagian berikut di file Anda:

```
[credential]
  helper = !aws codecommit credential-helper $@
  UseHttpPath = true
```

3. Ubah bagian itu menjadi berikut:

```
[credential "https://git-codecommit.*.amazonaws.com"]
  helper = !aws codecommit credential-helper $@
  UseHttpPath = true
```

4. Simpan perubahan Anda, lalu selesaikan langkah-langkah untuk mengkloning repositori Anda.

# Amazon CloudWatch Integrasi log untuk Visual Studio

Amazon CloudWatch Integrasi log dari AWS Toolkit for Visual Studio memberi Anda kemampuan untuk memantau, menyimpan, dan mengakses CloudWatch Log sumber daya, tanpa harus meninggalkan IDE Anda. Untuk mempelajari selengkapnya tentang menyiapkan CloudWatch layanan dan cara bekerja dengan CloudWatch Fitur log, pilih dari topik berikut.

Topik

- [Menyiapkan CloudWatch Integrasi log untuk Visual Studio](#)
- [Bekerja dengan CloudWatch Log di Visual Studio](#)

## Menyiapkan CloudWatch Integrasi log untuk Visual Studio

Sebelum Anda dapat menggunakan Amazon CloudWatch Integrasi log dengan Toolkit for Visual Studio, Anda memerlukan AWS akun. Anda dapat membuat AWS akun dari [AWS masuk ke Situs](#). Sebagian besar CloudWatch Fitur log yang tersedia dari Toolkit for Visual Studio dapat diakses dengan aktif AWS kredensi. Jika fitur tertentu memerlukan konfigurasi tambahan, persyaratan termasuk dalam bagian yang relevan dari [Bekerja dengan CloudWatch Beberapa catatan](#) guide.

Untuk informasi dan opsi tambahan tentang pengaturan CloudWatch Log, lihat [Mempersiapkan](#) bagian dari Amazon CloudWatch Panduan log.

## Bekerja dengan CloudWatch Log di Visual Studio

Amazon CloudWatch Integrasi log memungkinkan Anda memantau, menyimpan, dan mengakses CloudWatch Log dari AWS Toolkit for Visual Studio. Memiliki akses ke CloudWatch Fitur log — tanpa perlu meninggalkan IDE Anda—meningkatkan efisiensi dengan menyederhanakan CloudWatch Log proses pengembangan dan mengurangi gangguan pada alur kerja Anda. Topik berikut menjelaskan cara bekerja dengan fitur dan fungsi dasar CloudWatch Integrasi log.

Topik

- [CloudWatch Grup log](#)
- [CloudWatch Streaming Log](#)
- [CloudWatch Log acara](#)
- [Akses tambahan ke CloudWatch Beberapa catatan](#)

## CloudWatch Grup log

SEBUAH log group adalah sekelompok log streams yang berbagi pengaturan retensi, pemantauan, dan kontrol akses yang sama. Tidak ada batas jumlah pengaliran log yang dapat bergabung dalam satu grup log.

### Melihat Grup log

Parameter View Log Groups Fitur menampilkan daftar log di CloudWatch Log Grup Explorer.

Untuk mengakses fitur Lihat Grup Log dan buka CloudWatch Grup log, selesaikan langkah-langkah berikut.

1. Dari AWS Explorer, memperluas Amazon CloudWatch.
2. Klik dua kali Grup log atau buka menu konteks (klik kanan) dan pilih Lihat, untuk membuka CloudWatch Grup log.

#### Note

Parameter CloudWatch Log Grup Explorer akan terbuka di lokasi jendela yang sama dengan Solutions Explorer.

### Memfilter Grup log

Akun individual Anda dapat berisi ribuan grup log yang berbeda. Untuk menyederhanakan pencarian Anda untuk grup tertentu, gunakan filtering dijelaskan di bawah ini.

1. Dari CloudWatch Grup log, atur kursor Anda ke bilah pencarian yang terletak di bagian atas jendela.
2. Mulai ketikkan awalan yang terkait dengan grup log yang Anda cari.
3. CloudWatch Grup log diperbarui secara otomatis untuk menampilkan hasil yang cocok dengan istilah pencarian yang Anda tentukan pada langkah sebelumnya.

### Grup log

Untuk menghapus grup log tertentu, lihat prosedur berikut.

1. Dari CloudWatch Grup log, klik kanan Grup log yang ingin Anda hapus.

2. Saat diminta, konfirmasikan bahwa Anda ingin menghapus Grup log yang saat ini dipilih.
3. Memilihyatombol menghapus grup log yang dipilih, kemudian menyegarkanCloudWatch Grup log.

## Grup log refresh

Untuk me-refresh daftar grup log saat ini yang ditampilkan diCloudWatch Grup log, pilihikon refreshtombol yang terletak dibatang alat.

## ARN Grup log

Untuk menyalin ARN grup log tertentu, selesaikan langkah-langkah yang dijelaskan di bawah ini.

1. DariCloudWatch Grup log, klik kanan Grup Log tempat Anda ingin menyalin ARN.
2. PilihSalin ARNopsi dari menu.
3. ARN sekarang disalin ke clipboard lokal Anda dan siap ditempel.

## CloudWatch Streaming Log

Pengaliran log adalah urutan log acara yang berbagi sumber yang sama.

### Note

Saat melihat log, perhatikan properti berikut:

- Secara default, aliran log diurutkan berdasarkan stempel waktu peristiwa terbaru.
- Kolom yang terkait dengan aliran log dapat diurutkan dalam urutan menaik atau menurun, dengan mengubahsisipanterletak di header kolom.
- Entri yang difilter hanya dapat diurutkan berdasarkanNama log.

## Melihat Log

1. DariCloudWatch Grup logklik dua kali Grup Log, atau klik kanan grup log dan pilihPengaliran logdari menu konteks.
2. Tab baru akan terbuka didokumenwindow, yang berisi daftar aliran log yang terkait dengan grup log Anda.

## Memfilter Log

1. Dari Streaming Log tab, didokumentasikan jendela, mengatur kursor Anda ke bilah pencarian.
2. Mulai ketikkan awalan yang terkait dengan aliran log yang Anda cari.
3. Saat Anda mengetik, tampilan saat ini secara otomatis diperbarui untuk memfilter Aliran Log Anda berdasarkan input Anda.

## Streaming Log

Untuk me-refresh daftar aliran log saat ini ditampilkan didokumentasikan jendela, pilih ikon refresh tombol, terletak dibatang alat, di samping Bilah pencarian.

## ARN Salin Log

Untuk menyalin ARN aliran log tertentu, selesaikan langkah-langkah yang dijelaskan di bawah ini.

1. Dari Streaming Log tab, didokumentasikan jendela, klik kanan log yang ingin Anda salin ARN dari.
2. Pilih Salin ARN opsi dari menu.
3. ARN sekarang disalin ke clipboard lokal Anda dan siap ditempel.

## Unduh Log

Parameter Pengaliran log mendownload fitur dan menyimpan aliran log yang dipilih secara lokal, di mana ia dapat diakses oleh alat kustom dan perangkat lunak untuk pengolahan tambahan.

1. Dari Streaming Log tab, didokumentasikan jendela, klik kanan log yang ingin Anda unduh.
2. Pilih Pengaliran log membuka Ekspor ke file teks dialog.
3. Pilih lokasi di mana Anda ingin menyimpan file secara lokal dan tentukan nama di bidang teks yang disediakan.
4. Konfirmasikan unduhan dengan memilih OKE. Status unduhan ditampilkan di Pusat Status Tugas Visual Studio

## CloudWatch Log acara

Log acara adalah catatan aktivitas yang direkam oleh aplikasi atau sumber daya yang dipantau oleh CloudWatch.

## Log acara

Peristiwa log ditampilkan sebagai tabel. Secara default, peristiwa diurutkan dari acara tertua ke yang terbaru.

Tindakan berikut dikaitkan dengan peristiwa log di Visual Studio:

- Mode teks terbungkus: Anda dapat beralih wrapped-text dengan mengklik suatu peristiwa.
- Teks-wrap tombol: terletak di document window **toolbar**, tombol ini matikan text-wrap dan off, untuk semua entri.
- Salin pesan ke clipboard Anda: pilih pesan yang ingin Anda salin, lalu klik kanan pilihan dan pilih Salin (Pintasan keyboard `Ctrl + C`).

## Melihat Log Acara

1. Dari dokumen jendela, pilih tab yang berisi daftar aliran log.
2. Klik dua kali aliran log, atau klik kanan aliran log dan pilih Pengaliran log Dari menu.
3. Baru Log acara tab akan terbuka di dokumen window, yang berisi tabel peristiwa log yang terkait dengan aliran log pilihan Anda.

## Memfilter Log Acara

Ada tiga cara bagi Anda untuk memfilter peristiwa log: berdasarkan konten, rentang waktu, atau keduanya. Untuk memfilter peristiwa log berdasarkan konten dan rentang waktu, mulailah dengan memfilter pesan berdasarkan konten atau rentang waktu, lalu filter hasil tersebut dengan metode lain.

Untuk memfilter peristiwa log Anda berdasarkan konten:

1. Dari Log acara tab, di dokumen jendela, mengatur kursor Anda ke bilah pencarian, yang terletak di bagian atas jendela.
2. Mulai ketikkan istilah atau frasa yang terkait dengan peristiwa log yang Anda cari.
3. Saat Anda mengetik, tampilan saat ini secara otomatis mulai memfilter peristiwa log Anda.


### Note

Pola filter peka huruf besar/kecil. Anda dapat meningkatkan hasil pencarian dengan melampirkan istilah, dan frasa yang tepat, dengan karakter non-alfanumerik dalam tanda

kutip ganda (\*""\*). Untuk informasi lebih detail tentang pola filter, lihat [Filter dan Pola Sintaks](#) topik di Amazon CloudWatch guide.

Untuk melihat peristiwa log yang dihasilkan selama rentang waktu tertentu:

1. Dari Log acara tab, didokumentasikan jendela, pilih ikon kalender tombol, terletak dibatang alat.
2. Menggunakan bidang yang disediakan, tentukan rentang waktu yang ingin Anda cari.
3. Hasil yang difilter diperbarui secara otomatis saat Anda menentukan batasan tanggal dan waktu.

 Note

Parameter Filter yang jelas opsi menghapus semua saat Anda date-and-time pilihan filter.

## Memrefresh Log Acara


Untuk me-refresh daftar peristiwa log saat ini ditampilkan di Log acara tab, pilih ikon refresh tombol, terletak dibatang alat.

## Akses tambahan ke CloudWatch Beberapa catatan

Anda dapat mengakses CloudWatch Log yang terkait dengan AWS layanan dan sumber daya langsung dari AWS Toolkit dalam Visual Studio.

## Lambda

Untuk melihat aliran log yang terkait dengan fungsi Lambda:

 Note

Peran eksekusi Lambda Anda harus memiliki izin yang sesuai untuk mengirim log CloudWatch Log. Untuk informasi lebih lanjut tentang izin Lambda yang diperlukan CloudWatch Log, lihat <https://docs.aws.amazon.com/lambda/latest/dg/monitoring-cloudwatchlogs.html#monitoring-cloudwatchlogs-prereqs>

1. Dari AWS Toolkit Explorer, memperluas Lambda.



2. klik kanan fungsi yang ingin Anda lihat, lalu pilih **Melihat log** untuk membuka aliran log terkait di dokumen jendela.

Untuk melihat aliran log menggunakan integrasi **Lambda function view**:

1. Dari **AWS Toolkit Explorer**, memperluas **Lambda**.
2. klik kanan fungsi yang ingin Anda lihat, lalu pilih **Lihat Fungsi** untuk membuka tampilan fungsi di dokumen jendela.
3. Dari **function view**, beralih ke **Beberapa catatan**, aliran log yang terkait dengan fungsi Lambda yang dipilih ditampilkan.

## ECS

Untuk melihat sumber daya log yang terkait dengan ECS Task Container, selesaikan prosedur berikut.

### Note

Agar layanan Amazon ECS dapat mengirim log ke CloudWatch, setiap kontainer untuk Tugas Amazon ECS tertentu harus memenuhi konfigurasi yang diperlukan. Untuk informasi tambahan tentang pengaturan dan konfigurasi yang diperlukan, silakan lihat panduan [Menggunakan AWS Driver Log Log](#).

1. Dari **AWS Toolkit Explorer**, memperluas **Amazon ECS**.
2. Pilih **Cluster Amazon ECS** yang ingin Anda lihat untuk membuka yang baru **Kluster ECS**, di dokumen jendela.
3. Dari menu navigasi, terletak di sisi kiri **Kluster ECS**, pilih **Tugas** untuk mencantumkan semua tugas yang terkait dengan cluster.
4. Dari **Tugas display**, pilih tugas dan pilih **Melihat log link**, terletak di sudut kiri bawah.

### Note

Tampilan ini mencantumkan semua tugas yang terkandung dalam kluster, **View Log link** hanya terlihat untuk setiap tugas yang memenuhi konfigurasi log diperlukan.

- Jika Tugas hanya terkait dengan satu kontainer, Melihat loglink membuka aliran log kontainer itu.
- Jika Tugas dikaitkan dengan beberapa kontainer, Melihat loglink membuka Lihat CloudWatch Log untuk Tugas ECS dialog, gunakan Wadah: menu drop-down untuk memilih wadah yang ingin Anda lihat Log, lalu pilih OKE.

5. Tab baru terbuka di dokumen window menampilkan aliran log yang terkait dengan pilihan kontainer Anda.

## Mengelola Instans Amazon EC2

AWSExplorer menyediakan tampilan rinci tentang instans Amazon Machine Image (AMI) dan instans Amazon Elastic Compute Cloud (Amazon EC2). Dari tampilan ini, Anda dapat meluncurkan instans Amazon EC2 dari AMI, terhubung ke instans tersebut, dan menghentikan atau menghentikan instans, semuanya dari dalam lingkungan pengembangan Visual Studio. Anda dapat menggunakan tampilan instans untuk membuat AMI dari instans Anda. Untuk informasi selengkapnya, lihat [Membuat AMI dari Instans Amazon EC2](#).

## Gambar Mesin Amazon dan Tampilan Instans Amazon EC2

From AWSExplorer, Anda dapat menampilkan tampilan Amazon Machine Image (AMI) dan instans Amazon EC2. Masuk AWSExplorer, memperluas Amazon EC2 simpul

Untuk menampilkan tampilan AMI, pada subnode pertama, AMI, buka menu konteks (klik kanan) lalu pilih Lihat.

Untuk menampilkan tampilan instans Amazon EC2, pada Instans node, buka menu konteks (klik kanan) lalu pilih Lihat.

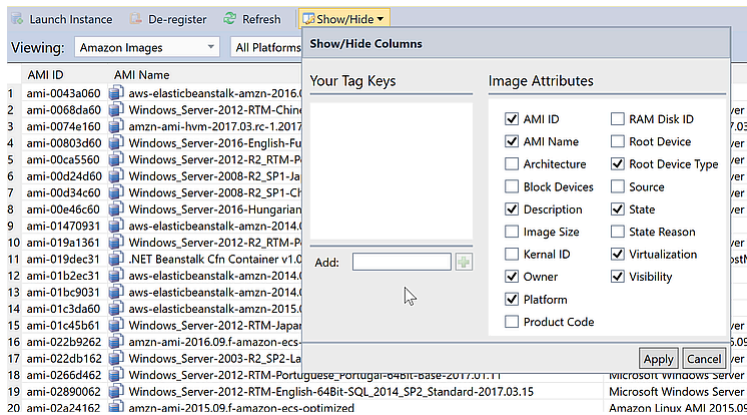
Anda juga dapat menampilkan tampilan baik dengan mengklik dua kali node yang sesuai.

- Pandangan yang scoped ke wilayah yang ditentukan dalam AWSExplorer (misalnya, wilayah Barat AS (N. California)).
- Anda dapat mengatur ulang kolom dengan mengklik dan menyeret. Untuk mengurutkan nilai dalam kolom, klik judul kolom.

- Anda dapat menggunakan daftar drop-down dan kotak filter di bagian atas tampilan untuk mengkonfigurasi tampilan. Tampilan awal menampilkan AMI dari jenis platform apa pun (Windows atau Linux) yang dimiliki oleh akun yang ditentukan dalam AWS Penjelajah.

## Tampilkan/Sembunyikan Kolom

Anda juga dapat memilih Tampilkan/Sembunyikan drop-down di bagian atas tampilan untuk mengkonfigurasi kolom mana yang ditampilkan. Pilihan kolom Anda akan bertahan jika Anda menutup tampilan dan membukanya kembali.



## Tampilkan/Sembunyikan Kolom UI untuk tampilan AMI dan Instans

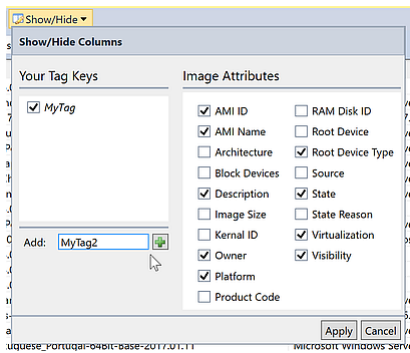
### Menandai AMI, Instans, dan Volume

Anda juga dapat menggunakan Tampilkan/Sembunyikan daftar drop-down untuk menambahkan tag untuk AMI, instans Amazon EC2, atau volume yang Anda miliki. Tag adalah pasangan nilai nama yang memungkinkan Anda melampirkan metadata ke AMI, instans, dan volume Anda. Nama tag dicakup baik ke akun Anda dan juga secara terpisah ke AMI dan instans Anda. Misalnya, tidak akan ada konflik jika Anda menggunakan nama tag yang sama untuk AMI dan instans Anda. Nama tag tidak peka huruf besar/kecil.

Untuk informasi selengkapnya tentang tanda, kunjungi [Menggunakan Tag](#) di dalam Panduan Pengguna Amazon EC2 untuk Instans Linux.

### Untuk menambahkan tanda

1. Di Tambah kotak, ketik nama untuk tanda. Pilih tombol hijau dengan tanda tambah (+), lalu pilih Terapkan.



## Menambahkan tag ke instans AMI atau Amazon EC2

Tag baru ditampilkan dalam huruf miring, yang menunjukkan tidak ada nilai yang belum dikaitkan dengan tag tersebut.

Dalam tampilan daftar, nama tag muncul sebagai kolom baru. Ketika setidaknya satu nilai telah dikaitkan dengan tag, tag akan terlihat di [AWS Management Console](#).

2. Untuk menambahkan nilai untuk tag, klik dua kali sel di kolom untuk tag itu, dan ketik nilai. Untuk menghapus nilai tag, klik dua kali sel dan hapus teks.

Jika Anda menghapus tag di Tampilkan/Sembunyikan daftar drop-down, kolom yang sesuai menghilang dari tampilan. Tag diawetkan, bersama dengan nilai tag yang terkait dengan AMI, instance, atau volume.

### Note

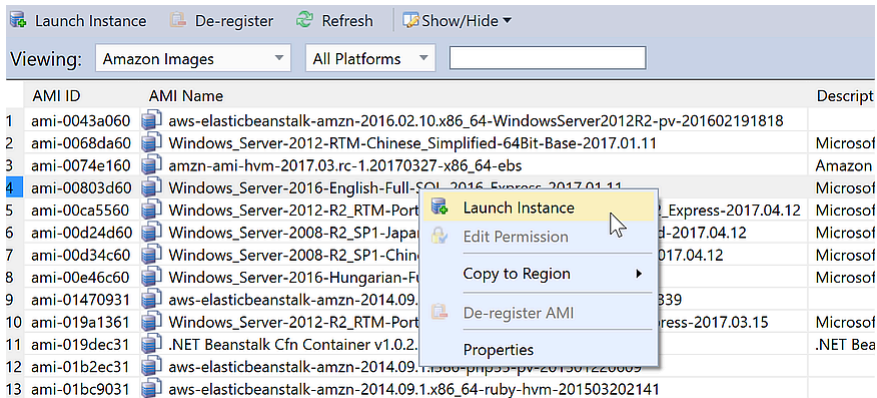
Jika Anda menghapus tag di Tampilkan/Sembunyikan daftar drop-down yang tidak memiliki nilai terkait, AWSToolkit akan menghapus tag seluruhnya. Tidak akan lagi muncul dalam tampilan daftar atau di Tampilkan/Sembunyikan daftar tarik-turun Untuk menggunakan tag itu lagi, gunakan Tampilkan/Sembunyikan kotak dialog untuk membuat ulang itu.

## Meluncurkan Instans Amazon EC2

AWSExplorer menyediakan semua fungsi yang diperlukan untuk meluncurkan instans Amazon EC2. Di bagian ini, kami akan memilih Amazon Machine Image (AMI), mengonfigurasinya, lalu memulainya sebagai instans Amazon EC2.

Untuk meluncurkan instans Amazon EC2 Windows Server

1. Di bagian atas tampilan AMI, dalam daftar drop-down di sebelah kiri, pilih Gambar Amazon. Di daftar tarik-turun di sebelah kanan, pilih Windows. Di kotak filter, ketik es untuk Elastic Block Storage. Mungkin butuh waktu beberapa saat agar tampilan disegarkan.
2. Pilih AMI dalam daftar, buka menu konteks (klik kanan), lalu pilih Instans.



## Daftar AMI

3. Di Luncurkan Instans Amazon EC2 kotak dialog, konfigurasi AMI untuk aplikasi Anda.

## Tipe Instans

Pilih jenis instans EC2 yang akan diluncurkan. Anda dapat menemukan daftar jenis instans dan informasi harga pada [Harga EC2](#) halaman.

## Nama

Ketikkan nama untuk instans Anda. Nama ini tidak boleh lebih dari 256 karakter.

## Pasangan kunci

key pair digunakan untuk mendapatkan kata sandi Windows yang Anda gunakan untuk masuk ke instans EC2 menggunakan Remote Desktop Protocol (RDP). Pilih key pair yang Anda miliki akses ke kunci pribadi, atau pilih opsi untuk membuat key pair. Jika Anda membuat key pair di Toolkit, Toolkit dapat menyimpan kunci pribadi untuk Anda.

Pasangan kunci yang disimpan dalam Toolkit dienkripsi. Anda dapat menemukannya di %LOCALAPPDATA%\AWSToolkit\keypairs (biasanya: C:\Users\\AppData\Local\AWSToolkit\keypairs). Anda dapat mengekspor key pair ke .pem berkas

- a. Di Visual Studio, pilih Lihat dan klik AWSPenjelajah.
- b. Klik Amazon EC2 dan pilih Pasangan kunci.
- c. Pasangan kunci akan terdaftar, dan yang dibuat/dikelola oleh Toolkit ditandai sebagai Disimpan di AWS Toolkit.

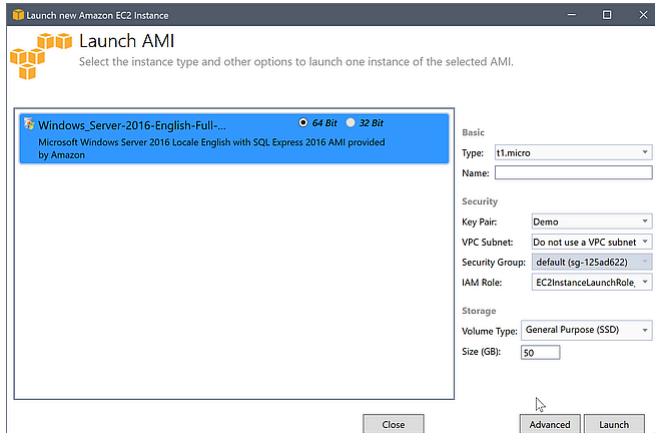
- d. Klik kanan pada key pair yang Anda buat dan pilih Ekspor kunci. Kunci pribadi tidak akan terenkripsi dan disimpan di lokasi yang Anda tentukan.

## Grup keamanan

Grup keamanan mengontrol jenis lalu lintas jaringan yang akan diterima instans EC2. Pilih grup keamanan yang akan memungkinkan lalu lintas masuk pada port 3389, port yang digunakan oleh RDP, sehingga Anda dapat terhubung ke instans EC2. Untuk informasi tentang cara menggunakan Toolkit untuk membuat grup keamanan, lihat [Mengelola Grup Keamanan dari AWS Penjelajah](#).

## Profil instans

Profil instans adalah kontainer logis untuk sebuah peran IAM. Ketika Anda memilih profil instans, Anda mengaitkan peran IAM yang sesuai dengan instans EC2. Peran IAM dikonfigurasi dengan kebijakan yang menentukan akses ke Amazon Web Services dan sumber daya akun. Ketika instans EC2 dikaitkan dengan peran IAM, perangkat lunak aplikasi yang berjalan pada instance berjalan dengan izin yang ditentukan oleh peran IAM. Hal ini memungkinkan perangkat lunak aplikasi berjalan tanpa harus menentukan AWS kredensialnya sendiri, yang membuat perangkat lunak lebih aman. Untuk informasi selengkapnya tentang peran IAM, kunjungi [Panduan Pengguna IAM](#).

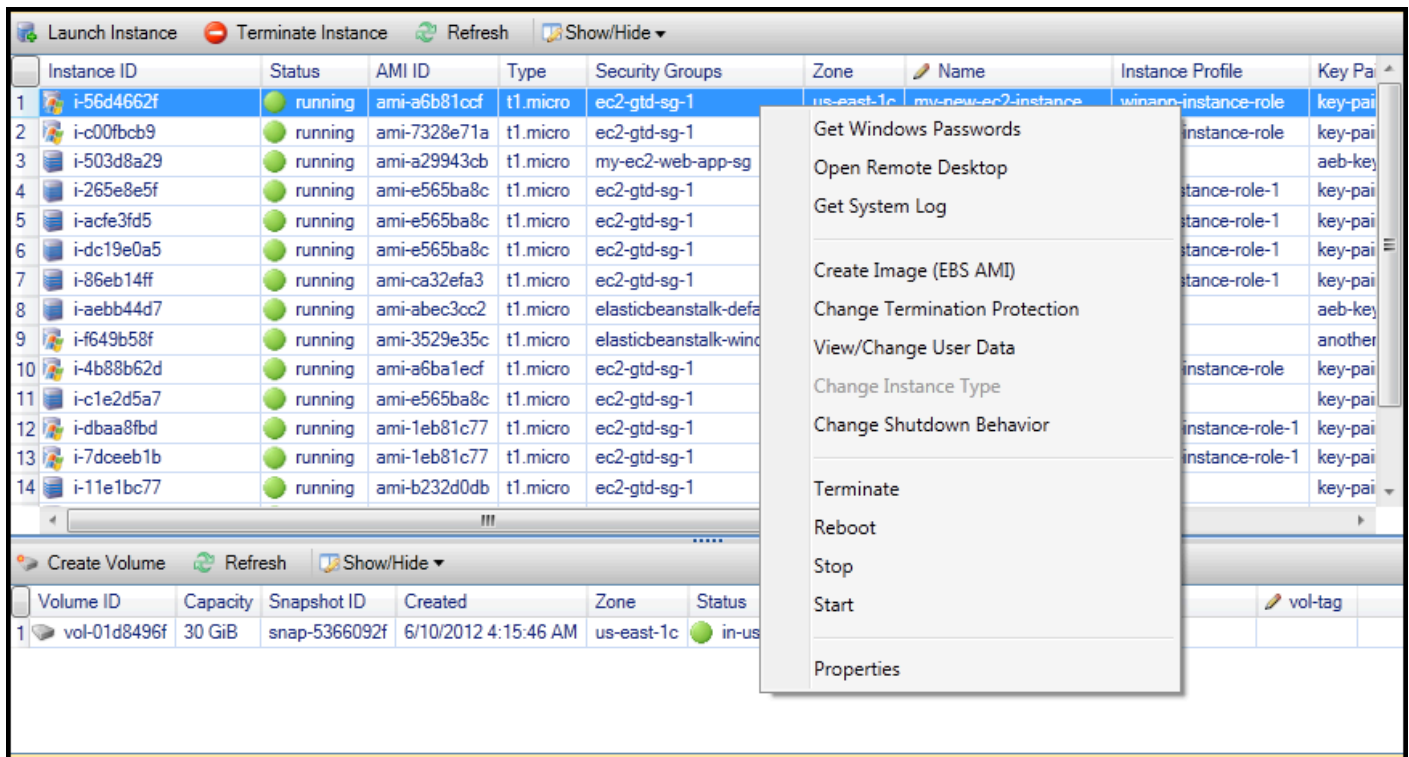


## EC2 Luncurkan AMI kotak dialog

### 4. Pilih Luncurkan.

Masuk AWS Explorer, di Instans subnode dari Amazon EC2, buka menu konteks (klik kanan) lalu pilih Lihat. Parameter AWSToolkit menampilkan daftar instans Amazon EC2 yang terkait dengan akun aktif. Anda mungkin perlu memilih Refresh untuk melihat instans baru. Ketika instance

pertama kali muncul, mungkin dalam keadaan tertunda, tetapi setelah beberapa saat, itu transisi ke keadaan berjalan.



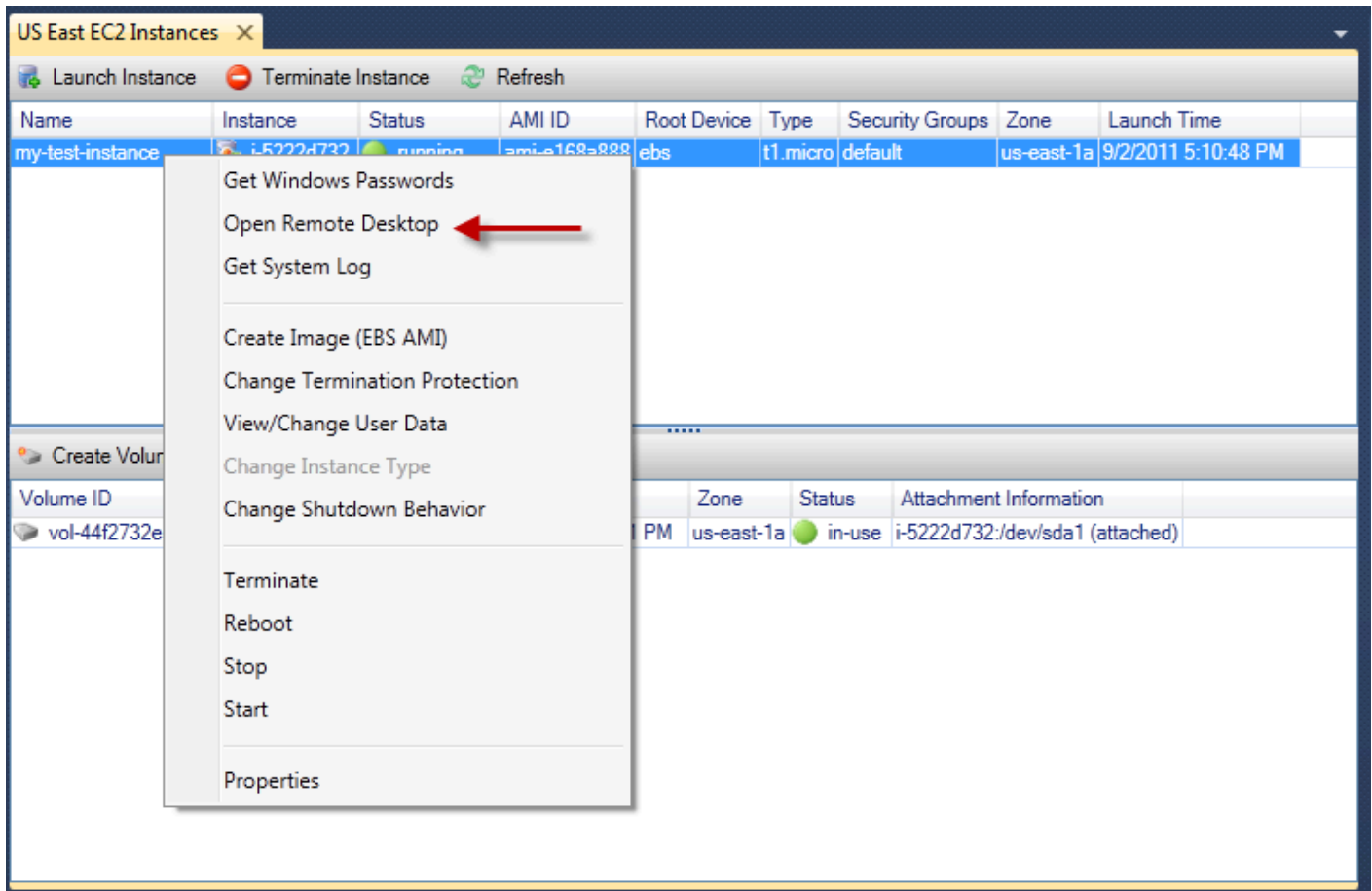
## Menghubungkan ke Instans Amazon EC2

Anda dapat menggunakan Windows Remote Desktop untuk menyambung ke instance Windows Server. Untuk otentikasi, AWSToolkit memungkinkan Anda untuk mengambil password administrator untuk instance, atau Anda hanya dapat menggunakan key pair yang disimpan terkait dengan instance. Dalam prosedur berikut, kita akan menggunakan key pair yang disimpan.

Untuk menyambung ke instans Windows Server menggunakan Windows Remote Desktop

1. Dalam daftar instans EC2, klik kanan instans Windows Server yang ingin Anda sambungkan. Dari menu konteks, pilih Desktop Jarak Jauh.

Jika Anda ingin mengotentikasi menggunakan kata sandi administrator, Anda akan memilih Dapatkan Sandi Windows.

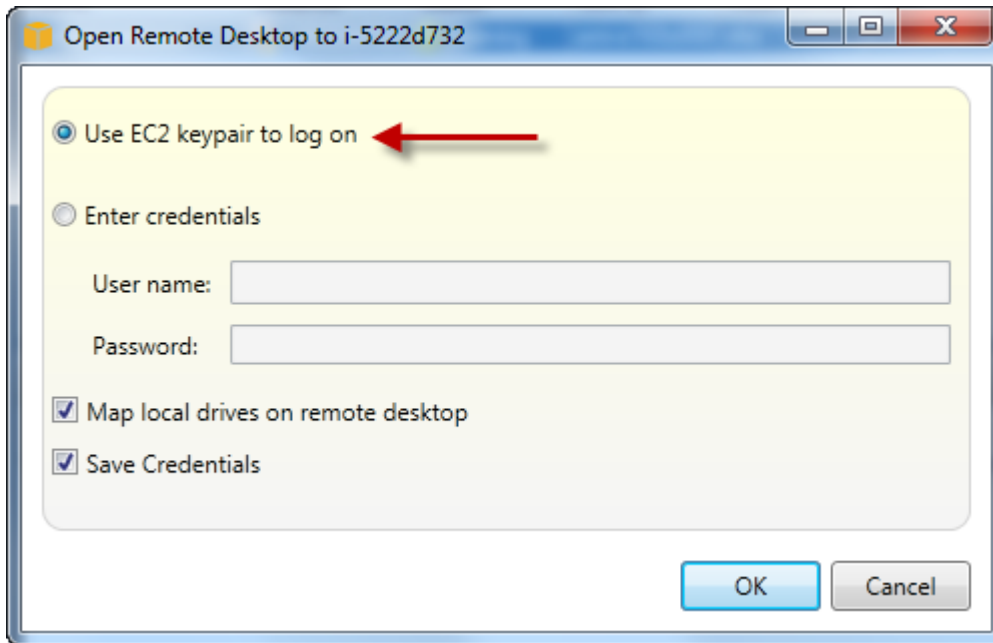


## Menu konteks Instans EC2

2. Di Desktop Jarak Jauh kotak dialog, pilih Gunakan keypair EC2 untuk log on, dan kemudian pilih OKE.

Jika Anda tidak menyimpan key pair dengan AWSToolkit, tentukan file PEM yang berisi kunci privat.



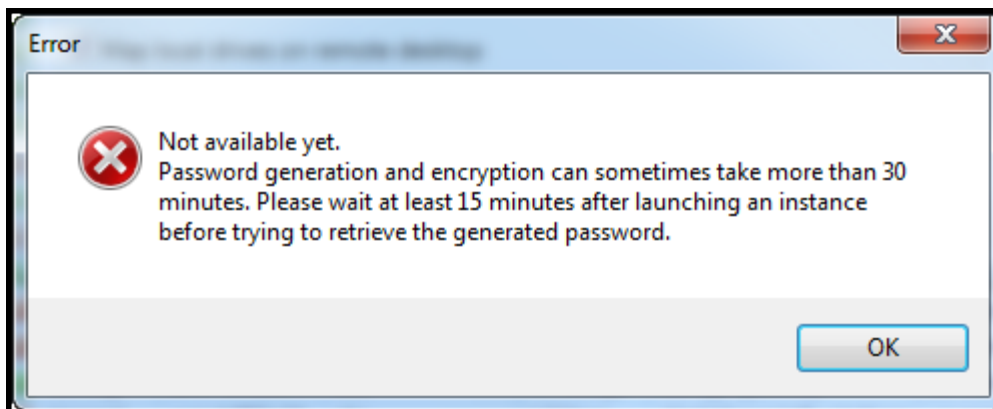


Desktop Jarak Jauh kotak dialog

3. ParameterDesktop Jarak Jauh jendela akan terbuka. Anda tidak perlu masuk karena otentikasi terjadi dengan key pair. Anda akan berjalan sebagai administrator pada instans Amazon EC2.

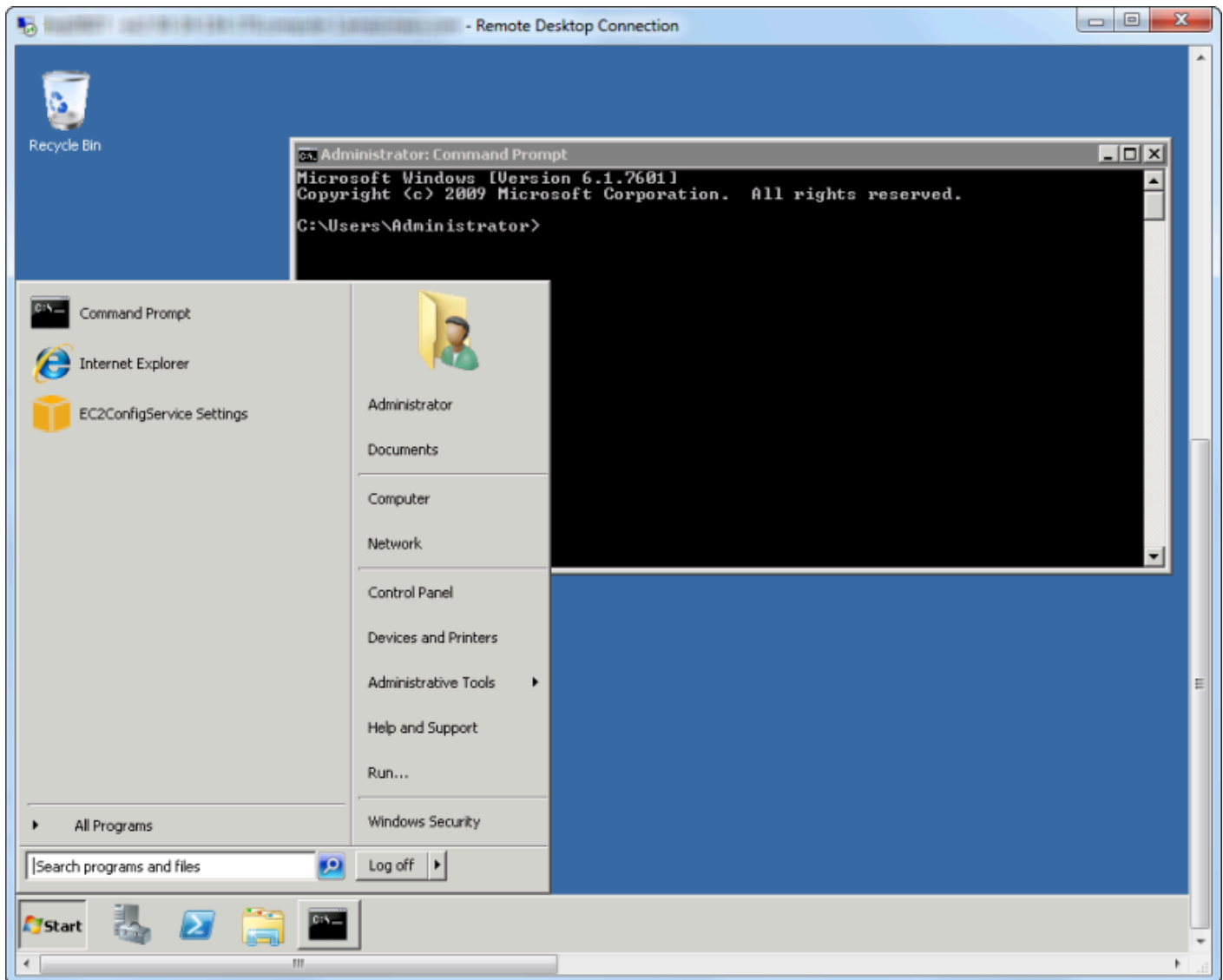
Jika instans EC2 baru saja dimulai, Anda mungkin tidak dapat terhubung karena dua kemungkinan alasan:

- Layanan Remote Desktop mungkin belum aktif dan berjalan. Tunggu beberapa menit dan coba lagi.
- Informasi kata sandi mungkin belum ditransfer ke instance. Dalam hal ini, Anda akan melihat kotak pesan yang serupa dengan yang terlihat berikut.



Kata sandi belum tersedia

Screenshot berikut menunjukkan pengguna yang terhubung sebagai administrator melalui Remote Desktop.



Desktop Jarak Jauh

## Mengakhiri Instans Amazon EC2

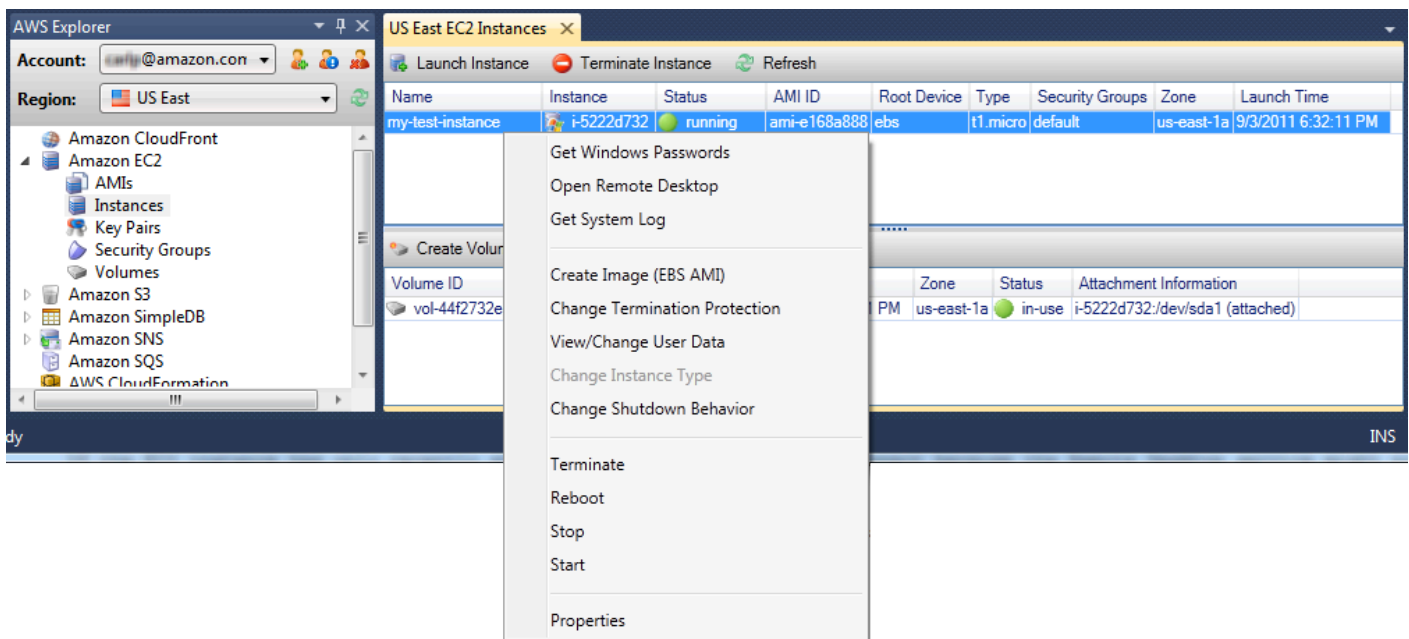
Menggunakan AWSToolkit, Anda dapat menghentikan atau menghentikan instans Amazon EC2 yang berjalan dari Visual Studio. Untuk menghentikan instans, instans EC2 harus menggunakan volume Amazon EBS. Jika instans EC2 tidak menggunakan volume Amazon EBS, maka satu-satunya opsi Anda adalah menghentikan instans.

Jika Anda menghentikan instance, data yang disimpan pada volume EBS dipertahankan. Jika Anda mengakhiri instance, semua data yang disimpan di perangkat penyimpanan lokal instance akan hilang. Dalam kedua kasus, berhenti atau mengakhiri, Anda tidak akan terus dikenakan biaya untuk instans EC2. Namun, jika Anda menghentikan instans, Anda akan terus ditagih untuk penyimpanan EBS yang bertahan setelah instans dihentikan.

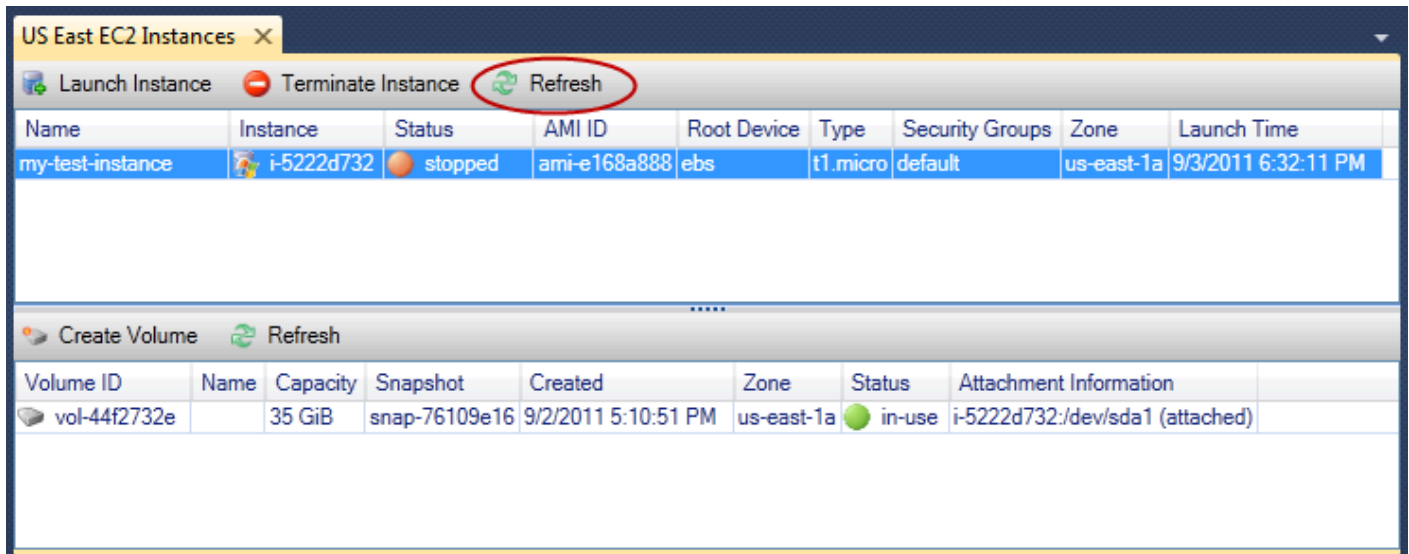
Cara lain yang mungkin untuk mengakhiri instance adalah dengan menggunakan Remote Desktop untuk menyambung ke instance, dan kemudian dari WindowsMulaimenu, gunakanShutdown. Anda dapat mengkonfigurasi instance untuk menghentikan atau mengakhiri dalam skenario ini.

Untuk menghentikan instans Amazon EC2

1. MasukAWSExplorer, memperluasAmazon EC2simpul, buka menu konteks (klik kanan)Instans, dan kemudian pilihLihat. DiInstansdaftar, klik kanan instans yang ingin Anda hentikan dan pilihBerhenti dari menu konteks. PilihYauntuk mengkonfirmasi Anda ingin menghentikan instance.

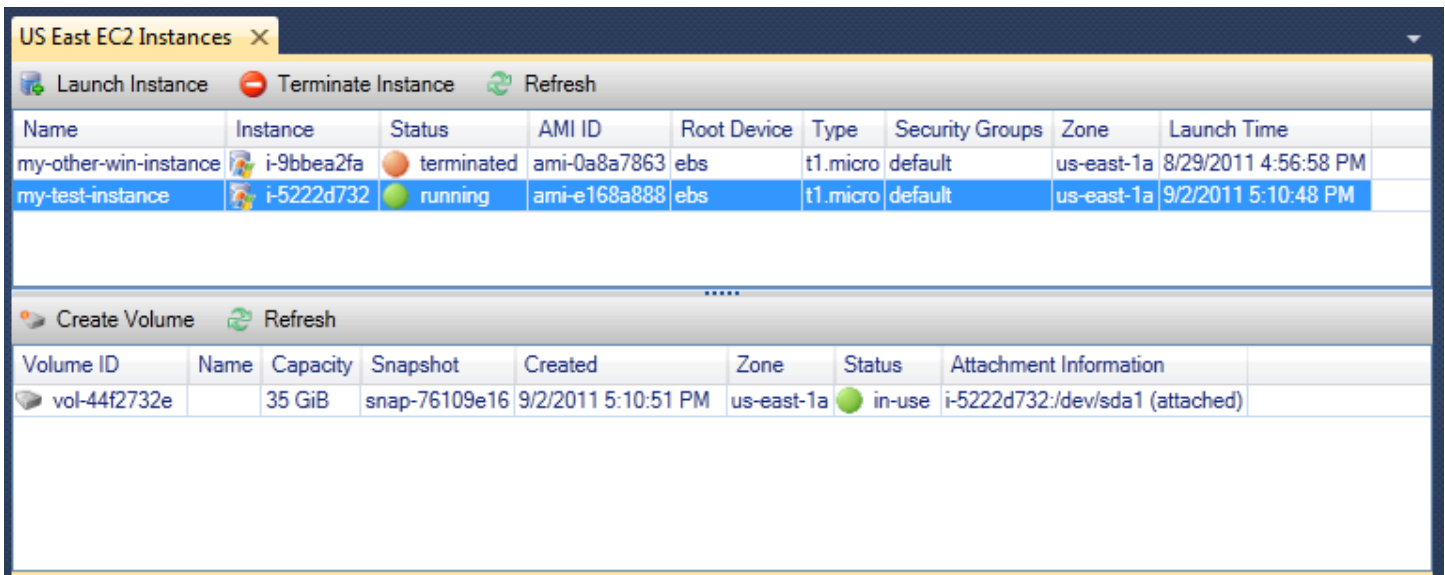


2. Di bagian atasInstansdaftar, pilihRefreshuntuk melihat perubahan status instans Amazon EC2. Karena kita berhenti daripada mengakhiri instance, volume EBS yang terkait dengan instance masih aktif.



## Instans Terhenti Tetap Terlihat

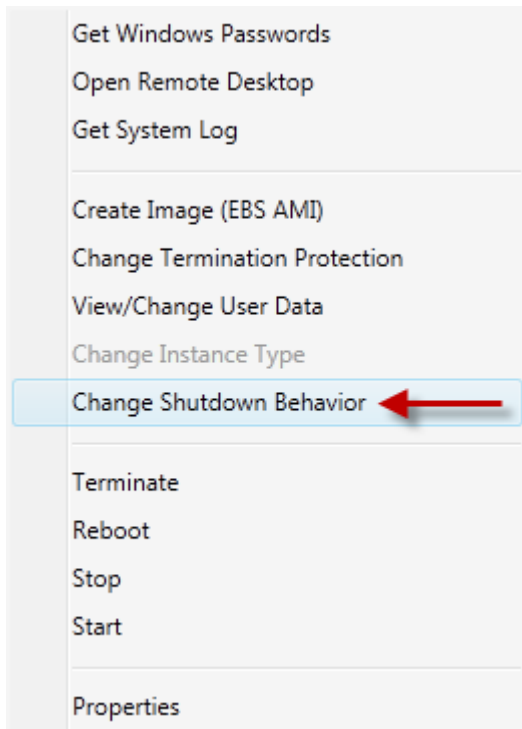
Jika Anda menghentikan sebuah instans, instans akan terus muncul di Instansdaftar bersama menjalankan atau berhenti instance. Pada akhirnya AWS merebut kembali instans ini dan mereka menghilang dari daftar. Anda tidak dikenakan biaya untuk instans dalam keadaan berakhir.



## Untuk menentukan perilaku instans EC2 saat shutdown

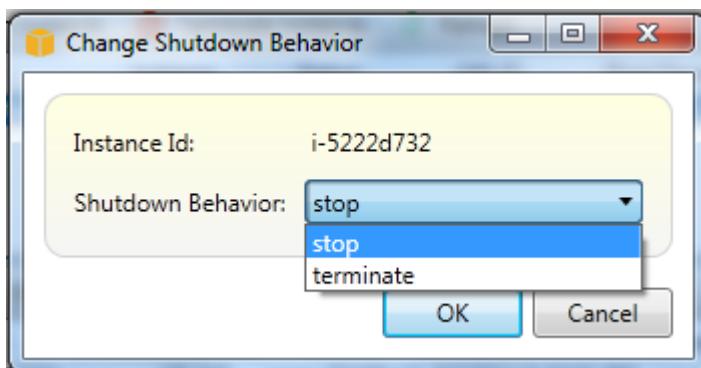
Parameter AWSToolkit memungkinkan Anda menentukan apakah instans Amazon EC2 akan berhenti atau dihentikan jika Shutdown dipilih dari Menu

1. Di Instansdaftar, klik kanan instans Amazon EC2, dan kemudian pilih Perilaku penghentian.



### Perilaku Shutdown item

2. Di Perilaku Shutdown kotak dialog, dari Perilaku Shutdown daftar tarik-turun, pilih Berhenti atau Akhiri.



## Mengelola instans Amazon ECS

AWSExplorer menyediakan tampilan rinci tentang kluster dan repositori kontainer Amazon Elastic Container Service (Amazon ECS). Anda dapat membuat, menghapus, dan mengelola rincian kluster dan kontainer dari dalam lingkungan pengembangan Visual Studio.

### Memodifikasi properti layanan

Anda dapat melihat detail layanan, acara layanan, dan properti layanan dari tampilan kluster.

1. Masuk **AWSExplorer**, buka menu konteks (klik kanan) untuk dikelola klaster, lalu pilih **Lihat**.
2. Di tampilan **Cluster ECS**, klik **Layan** di sebelah kiri, dan kemudian klik **Rinci** tab dalam tampilan detail. Anda dapat mengklik **Peristiwa** untuk melihat pesan acara dan **Deployment status** deployment.
3. Klik **Mengedit**. Anda dapat mengubah jumlah tugas yang diinginkan dan minimum dan maksimum persen sehat.
4. Klik **Simpan** untuk menerima perubahan atau **Membatalkan** untuk kembali ke nilai yang ada.

## Menghentikan tugas

Anda dapat melihat status tugas saat ini dan menghentikan satu atau lebih tugas dalam tampilan klaster.

Untuk menghentikan tugas

1. Masuk **AWSExplorer**, buka menu konteks (klik kanan) untuk klaster dengan tugas yang ingin Anda hentikan, lalu pilih **Lihat**.
2. Di tampilan **Cluster ECS**, klik **Tugas** di sebelah kiri.
3. Pastikan **Status Tugas** yang diinginkan diatur untuk **Running**. Pilih tugas individu untuk dihentikan dan kemudian klik **Berhenti** atau klik **Hentikan Semua** untuk memilih dan menghentikan semua tugas yang berjalan.
4. Di **Tugas Hentikan** kotak dialog, pilih ya.

## Menghapus layanan

Anda dapat menghapus layanan dari klaster dari tampilan klaster.

Untuk menghapus layanan klaster

1. Masuk **AWSExplorer**, buka menu konteks (klik kanan) untuk klaster dengan layanan yang ingin Anda hapus, lalu pilih **Lihat**.
2. Di tampilan **Cluster ECS**, klik **Layan** di sebelah kiri, dan kemudian klik **Hapus**.
3. Di **Menghapus Klaster** kotak dialog, jika ada load balancer dan target group di cluster Anda, Anda dapat memilih untuk menghapusnya dengan cluster. Mereka tidak akan digunakan saat layanan dihapus.
4. Di **Menghapus Klaster** kotak dialog, pilih **OKE**. Ketika cluster dihapus, itu akan dihapus dari **AWSExplorer**.

## Menghapus klaster

Anda dapat menghapus klaster Amazon Elastic Container Service dari AWSExplorer.

Untuk menghapus klaster

1. Masuk AWSExplorer, buka menu konteks (klik kanan) untuk klaster yang ingin Anda hapus di bawah klaster simpul Amazon ECS, dan kemudian pilih Hapus.
2. Di Menghapus Klaster kotak dialog, pilih OKE. Ketika cluster dihapus, itu akan dihapus dari AWSExplorer.

## Membuat repositori

Anda dapat membuat repositori Amazon Elastic Container Registry dari AWSExplorer.

Untuk membuat repositori

1. Masuk AWSExplorer, buka menu konteks (klik kanan) Repositori simpul bawah Amazon ECS, dan kemudian pilih Membuat Repositori.
2. Di Membuat Repositori kotak dialog, memberikan nama repositori dan kemudian memilih OKE.

## Menghapus repositori

Anda dapat menghapus repositori Amazon Elastic Container Registry dari AWSExplorer.

Untuk menghapus repositori

1. Masuk AWSExplorer, buka menu konteks (klik kanan) Repositori simpul bawah Amazon ECS, dan kemudian pilih Menghapus Repositori.
2. Di Menghapus Repositori kotak dialog, Anda dapat memilih untuk menghapus repositori bahkan jika berisi gambar. Jika tidak, itu hanya akan dihapus jika kosong. Klik ya.

## Mengelola Grup Keamanan dari AWS Penjelajah

Toolkit for Visual Studio memungkinkan Anda membuat dan mengonfigurasi grup keamanan yang akan digunakan dengan instans Amazon Elastic Compute Cloud (Amazon EC2) dan AWS CloudFormation. Saat Anda meluncurkan instans Amazon EC2 atau menerapkan aplikasi ke AWS

CloudFormation, Anda menentukan grup keamanan yang terkait dengan instans Amazon EC2. (Deployment keAWS CloudFormationmembuat instans Amazon EC2.)

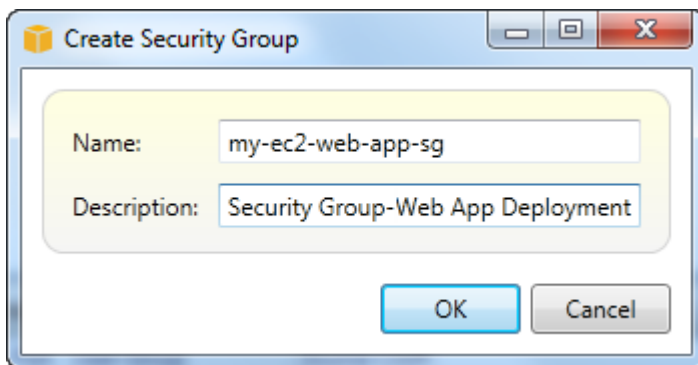
Grup keamanan bertindak seperti firewall di lalu lintas jaringan yang masuk. Grup keamanan menentukan jenis lalu lintas jaringan mana yang diizinkan pada instans Amazon EC2. Hal ini juga dapat menentukan bahwa lalu lintas masuk akan diterima dari alamat IP tertentu saja atau dari pengguna tertentu atau kelompok keamanan lainnya saja.

## Membuat Grup Keamanan

Di bagian ini, kita akan membuat grup keamanan. Setelah dibuat, grup keamanan tidak akan memiliki izin yang dikonfigurasi. Mengkonfigurasi izin ditangani melalui operasi tambahan.

Untuk menciptakan sebuah grup keamanan

1. MasukAWSExplorer, di bawahAmazon EC2node, buka menu konteks (klik kanan) diKelompok Keamanannode, dan kemudian pilihLihat.
2. PadaGrup Keamanan EC2tab, pilihBuat Grup Keamanan.
3. DiBuat Grup Keamanankotak dialog, ketikkan nama dan deskripsi untuk grup keamanan, lalu pilihOKE.



## Menambahkan Izin ke Grup Keamanan

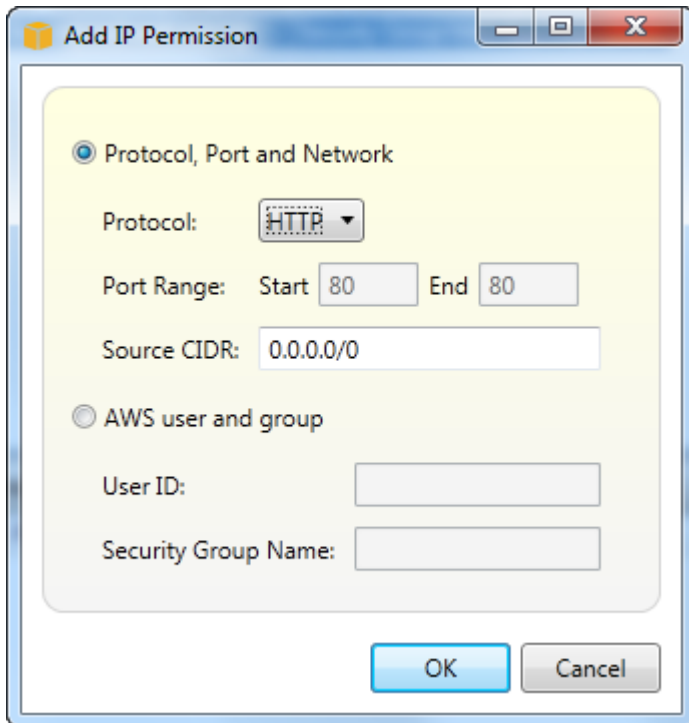
Pada bagian ini, kita akan menambahkan izin ke grup keamanan untuk memungkinkan lalu lintas web melalui protokol HTTP dan HTTPS. Kami juga akan mengizinkan komputer lain untuk terhubung dengan menggunakan Windows Remote Desktop Protocol (RDP).

Untuk menambahkan izin ke sebuah grup keamanan

1. PadaGrup Keamanan EC2tab, pilih grup keamanan dan kemudian pilihTambah Izintombol.

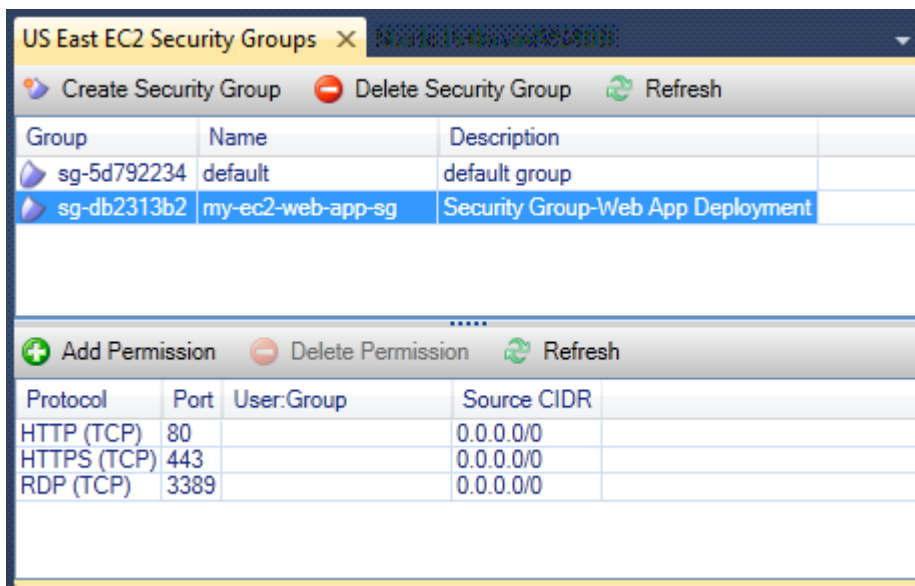


2. DiTambahkan Izin IPkotak dialog, pilihProtokol, Port dan Jaringan tombol radio, dan kemudian dariProtokoldaftar drop-down, pilihHTTP. Rentang port secara otomatis menyesuaikan ke port 80, port default untuk HTTP. ParameterSumber CIDRbidang default ke 0.0.0.0/0, yang menentukan bahwa lalu lintas jaringan HTTP akan diterima dari alamat IP eksternal manapun. Pilih OKE.



Buka port 80 (HTTP) untuk grup keamanan ini

3. Ulangi proses ini untuk HTTPS dan RDP. Izin grup keamanan Anda sekarang akan terlihat seperti berikut ini.



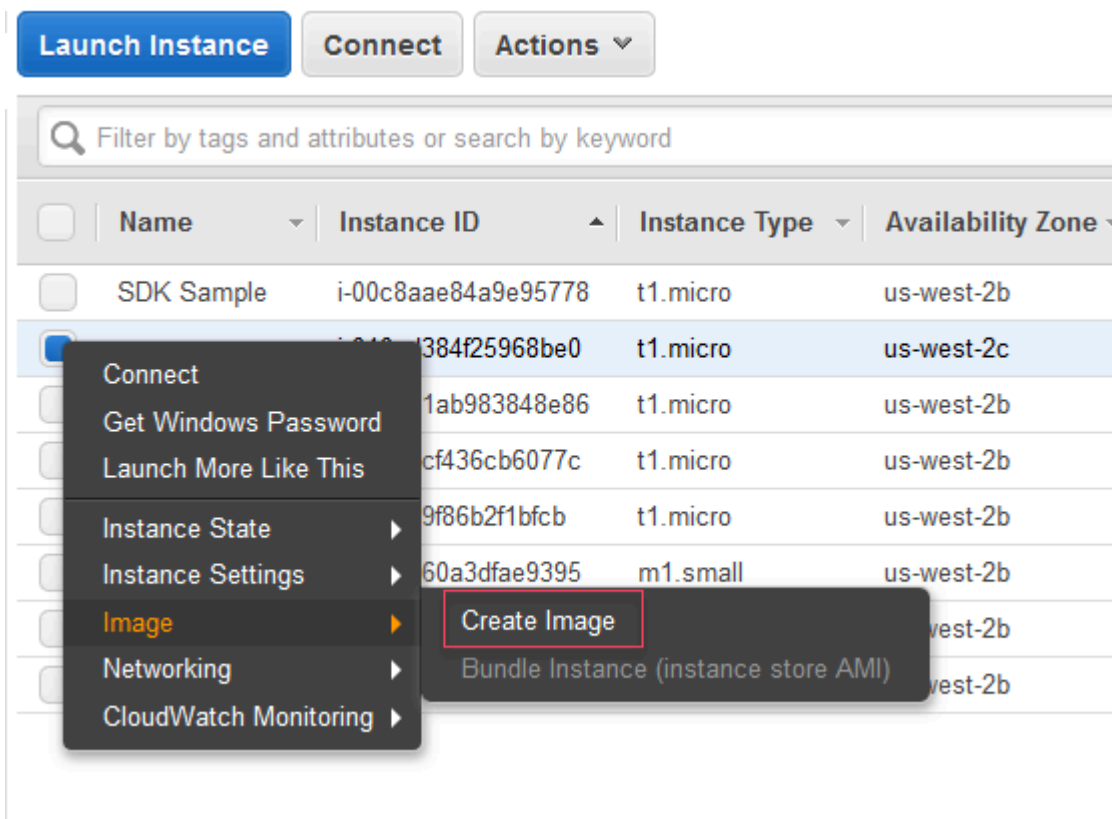
Anda juga dapat mengatur izin dalam grup keamanan dengan menentukan ID pengguna dan nama grup keamanan. Dalam kasus ini, instans Amazon EC2 dalam grup keamanan ini akan menerima semua lalu lintas jaringan masuk dari instans Amazon EC2 dalam grup keamanan yang ditentukan. Anda juga harus menentukan ID pengguna sebagai cara untuk menyingkapkan nama grup keamanan; nama grup keamanan tidak diharuskan unik di semua AWS. Untuk informasi selengkapnya tentang grup keamanan, kunjungi [Dokumentasi EC2](#).

## Membuat AMI dari Instans Amazon EC2

Dari tampilan Instans Amazon EC2, Anda dapat membuat Amazon Machine Images (AMI) baik dari instans yang sedang berjalan atau berhenti. Untuk informasi selengkapnya tentang AMI, lihat topik [Amazon Machine Images \(AMI\)](#) di Panduan Pengguna Amazon Elastic Compute Cloud untuk Instans Windows.

Untuk Membuat AMI dari suatu instans

1. Klik kanan instance yang ingin Anda gunakan sebagai dasar untuk AMI Anda, dan pilih Create Image dari menu konteks.

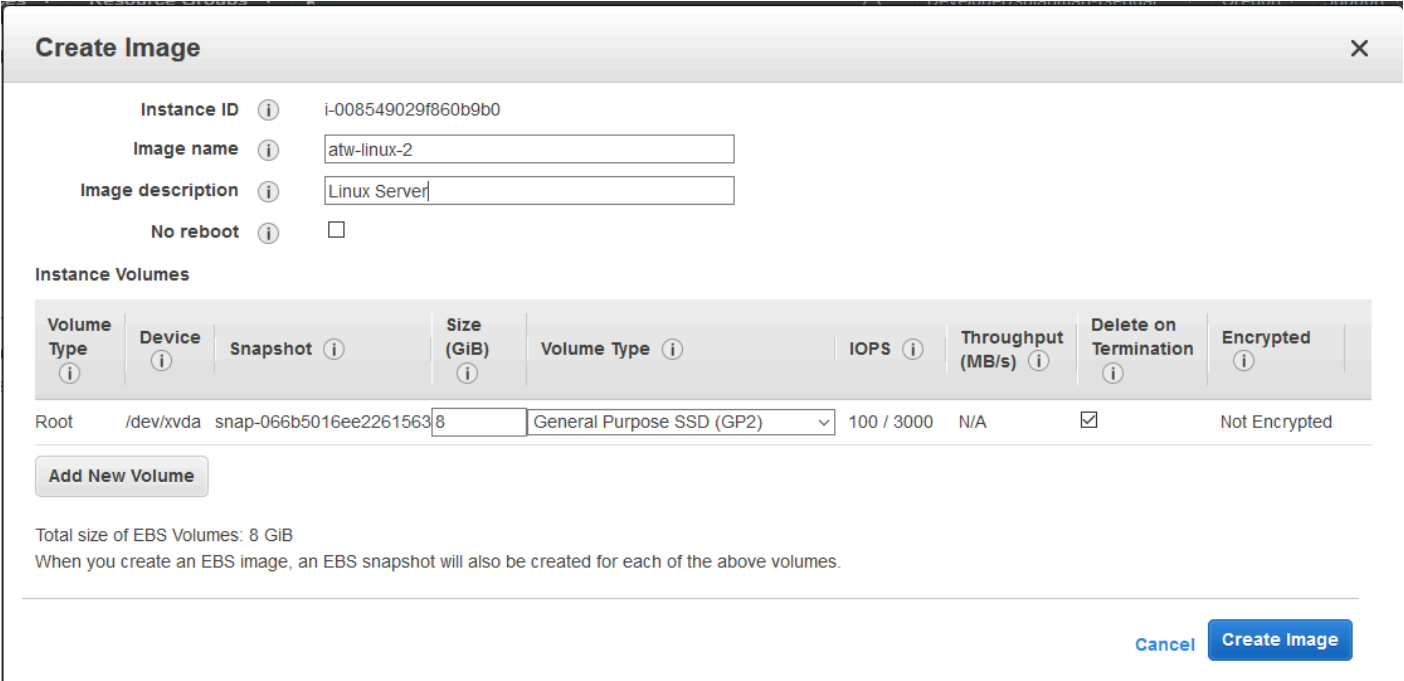


Buat menu konteks Gambar

2. Di kotak dialog Create Image, ketik nama dan deskripsi unik, lalu pilih Create Image. Secara default, Amazon EC2 menutup instans, mengambil snapshot volume yang terpasang, membuat, dan mendaftarkan AMI, kemudian mem-boot ulang instans. Pilih Tidak ada boot ulang jika Anda tidak ingin instans dimatikan.

### Warning

Jika Anda memilih Tidak ada boot ulang, kami tidak dapat menjamin integritas sistem file dari image yang dibuat.



**Create Image** ✕

Instance ID ⓘ i-008549029f860b9b0

Image name ⓘ atw-linux-2

Image description ⓘ Linux Server

No reboot ⓘ

**Instance Volumes**

Volume Type ⓘ	Device ⓘ	Snapshot ⓘ	Size (GiB) ⓘ	Volume Type ⓘ	IOPS ⓘ	Throughput (MB/s) ⓘ	Delete on Termination ⓘ	Encrypted ⓘ
Root	/dev/xvda	snap-066b5016ee22615638	8	General Purpose SSD (GP2) ▾	100 / 3000	N/A	<input checked="" type="checkbox"/>	Not Encrypted

Total size of EBS Volumes: 8 GiB  
When you create an EBS image, an EBS snapshot will also be created for each of the above volumes.

### Buat kotak dialog Image

Mungkin diperlukan waktu beberapa menit hingga AMI selesai dibuat. Setelah dibuat, itu akan muncul di tampilan AMI diAWS Explorer. Untuk menampilkan tampilan ini, klik dua kali simpul Amazon EC2 | AMI diAWS Explorer. Untuk melihat AMI Anda, dari daftar drop-down Melihat, pilih Dimiliki Oleh Saya. Anda mungkin perlu memilih Refresh untuk melihat AMI Anda. Ketika AMI pertama kali muncul, mungkin dalam keadaan tertunda, tetapi setelah beberapa saat, AMI akan beralih ke status yang tersedia.

Name	AMI Name	AMI ID	Source	Owner	Visibility	Status	Creation Date
	atw-linux-2	ami-d18412b1			Private	available	April 4, 2017 at 9:39:06 AM ...

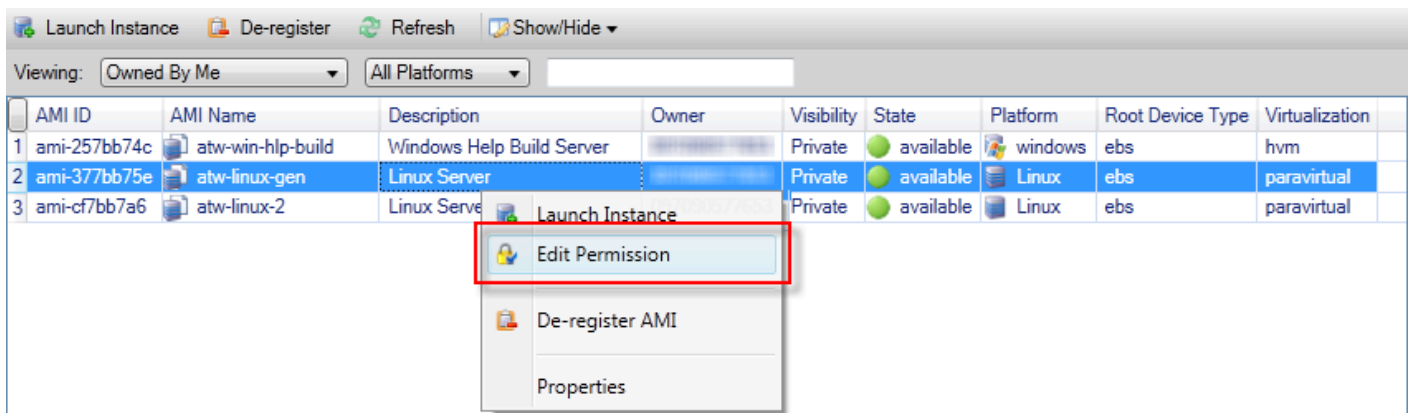
Daftar AMI yang dibuat

## Menyiapkan Izin Luncurkan pada Amazon Machine Image

Anda dapat mengatur izin peluncuran di Amazon Machine Images (AMI) dari AMI lihat AWSE Explorer. Anda dapat menggunakan Menyiapkan Izin AMI kotak dialog untuk menyalin izin dari AMI.

Menyiapkan izin AMI

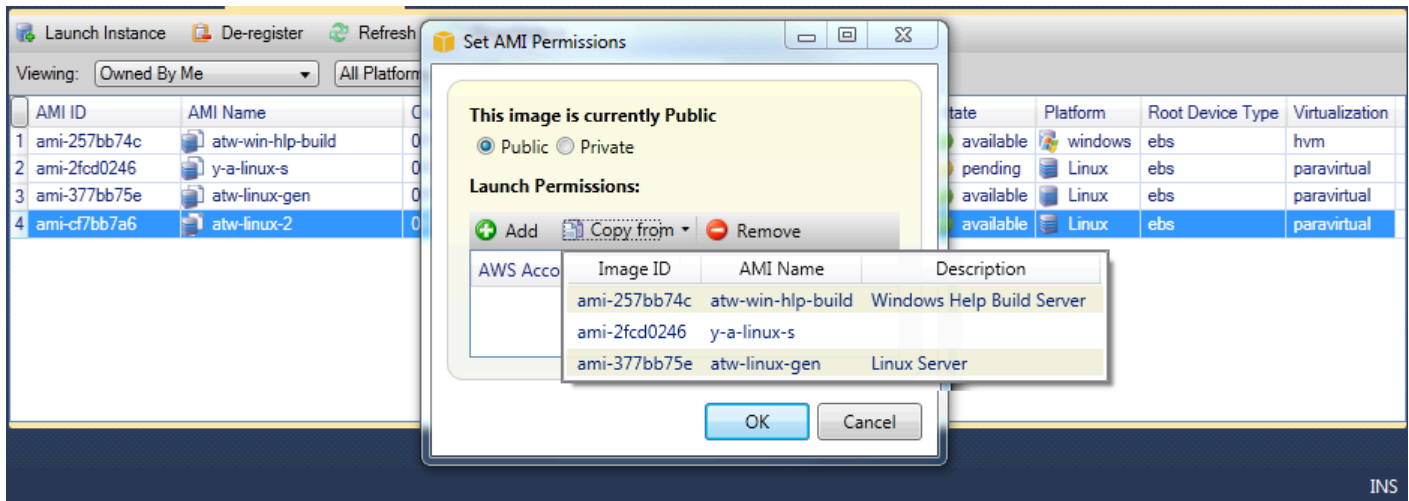
1. Di AMI lihat AWSE Explorer, buka menu konteks (klik kanan) pada AMI, lalu pilih Mengedit Izin.



2. Ada tiga opsi yang tersedia di Menyiapkan Izin AMI kotak dialog:

- Untuk memberikan izin peluncuran, pilih Tambahkan, lalu ketik nomor akun AWS pengguna kepada siapa Anda memberikan izin peluncuran.
- Untuk menghapus izin peluncuran, pilih nomor akun untuk AWS pengguna dari siapa Anda menghapus izin peluncuran, dan memilih Menghapus.
- Untuk menyalin izin dari AMI, pilih AMI dari daftar, lalu pilih Salin dari. Pengguna yang memiliki izin peluncuran pada AMI yang Anda pilih akan diberikan izin peluncuran pada AMI saat ini. Anda dapat mengulangi proses ini dengan AMI lainnya di Salinan-daridaftar untuk menyalin izin dari beberapa AMI ke AMI target.

Parameter Salinan-daridaftar hanya berisi AMI yang dimiliki oleh akun yang aktif saat AMI tampilan ditampilkan dari AWSE Explorer. Akibatnya, Salinan-daridaftar mungkin tidak menampilkan AMI jika tidak ada AMI lain yang dimiliki oleh akun aktif.



Izin salin AMI kotak dialog

## Amazon Virtual Private Cloud (VPC)

Amazon Virtual Private Cloud (Amazon VPC) memungkinkan Anda untuk meluncurkan sumber daya Amazon Web Services ke dalam jaringan virtual yang telah Anda tentukan. Jaringan virtual ini menyerupai jaringan tradisional yang akan Anda operasikan di pusat data Anda sendiri, dengan manfaatnya yaitu menggunakan infrastruktur yang dapat diskalakan AWS. Untuk informasi selengkapnya, kunjungi [Panduan Pengguna Amazon VPC](#).

Toolkit for Visual Studio memungkinkan pengembang untuk mengakses fungsi VPC mirip dengan yang terpapar oleh [AWS Management Console](#) tetapi dari lingkungan pengembangan Visual Studio. Parameter Amazon VPC simpul AWS Explorer termasuk subnode untuk bidang-bidang berikut.

- [VPC](#)
- [Subnet](#)
- [IP elastis](#)
- [Gateway internet](#)
- [ACL Jaringan](#)
- [Tabel Rute](#)
- [Kelompok Keamanan](#)

## Membuat VPC Publik-Swasta untuk Deployment dengan AWS Elastic Beanstalk

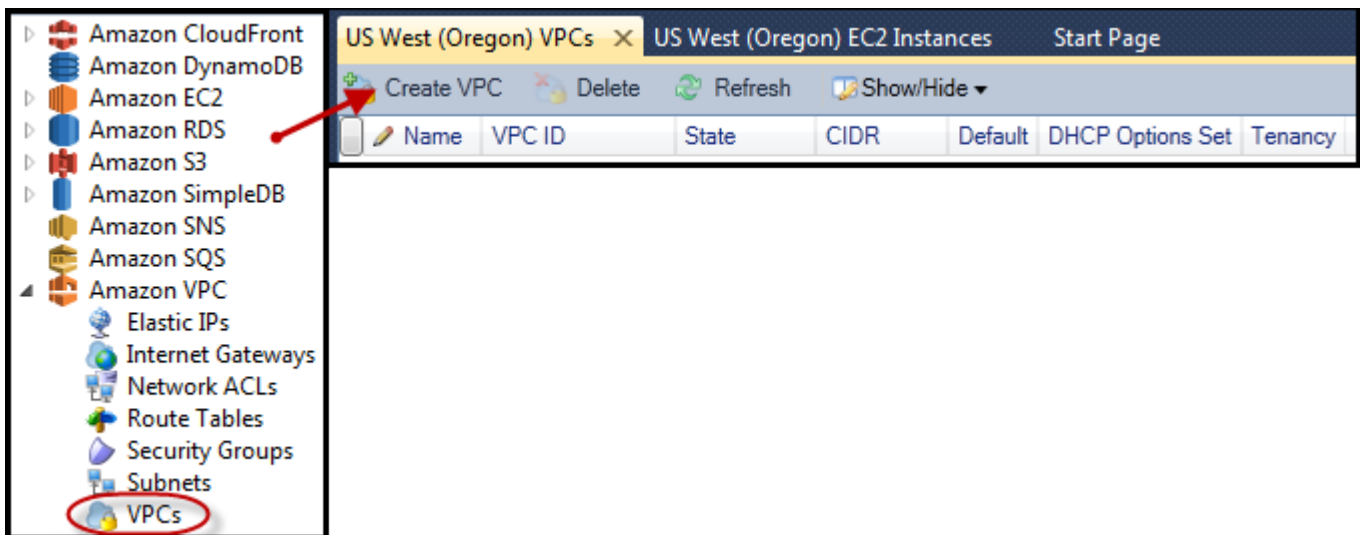
Bagian ini menjelaskan cara membuat Amazon VPC yang berisi subnet publik maupun pribadi. Subnet publik berisi instans Amazon EC2 yang melakukan penerjemahan alamat jaringan (NAT) agar instans di subnet pribadi dapat berkomunikasi dengan internet publik. Kedua subnet harus berada di Availability Zone (AZ) yang sama.

Ini adalah konfigurasi VPC minimal yang diperlukan untuk menyebarkan AWS Elastic Beanstalk lingkungan di VPC. Dalam skenario ini, instans Amazon EC2 yang meng-host aplikasi Anda berada di subnet pribadi; penyeimbang beban Elastic Load Balancing yang merutekan lalu lintas masuk ke aplikasi Anda berada di subnet publik.

Untuk informasi selengkapnya tentang penerjemahan alamat jaringan (NAT), kunjungi [Instans NAT](#) di Panduan Pengguna Amazon Virtual Private Cloud. Untuk contoh cara mengonfigurasi penerapan Anda agar menggunakan VPC, lihat [Menyebarkan ke Elastic Beanstalk](#).

Untuk membuat subnet publik-swasta VPC

1. Di Amazon VPCs simpul di AWS Explorer, buka VPC subnode, lalu pilih **Membuat VPC**.



2. Konfigurasi VPC sebagai berikut:

- Ketikkan nama untuk VPC Anda.
- Pilih **Dengan Subnet publik** dan **Dengan Subnet privat** kotak centang.
- Dari **Availability Zone** kotak daftar drop-down untuk setiap subnet, pilih Availability Zone. Pastikan untuk menggunakan AZ yang sama untuk kedua subnet.

- Untuk subnet pribadi, diNama Pasangan Kunci NAT, berikan key pair. key pair ini digunakan untuk instans Amazon EC2 yang melakukan terjemahan alamat jaringan dari subnet pribadi ke Internet publik.
- PilihKonfigurasi grup keamanan default untuk mengizinkan lalu lintas ke NATkotak centang.

Ketikkan nama untuk VPC Anda. PilihDengan Subnet publikdanDengan Subnet privatkotak centang. DariAvailability ZoneKotak daftar drop-down untuk setiap subnet, pilih Availability Zone. Pastikan untuk menggunakan AZ yang sama untuk kedua subnet. Untuk subnet pribadi, diNama Pasangan Kunci NAT, berikan key pair. key pair ini digunakan untuk instans Amazon EC2 yang melakukan terjemahan alamat jaringan dari subnet pribadi ke Internet publik. PilihKonfigurasi grup keamanan default untuk mengizinkan lalu lintas ke NATkotak centang.

Pilih OKE.

The screenshot shows the 'Create VPC' dialog box with the following configuration:

- Name: myDeploymentVPC
- CIDR Block\*: 10.0.0.0/16
- Tenancy: default
- With Public Subnet
  - Public Subnet: 10.0.0.0/24
  - Availability Zone: us-west-2b
- With Private Subnet
  - Private Subnet: 10.0.1.0/24
  - Availability Zone: us-west-2b
  - NAT Instance Type: Small
  - NAT Key Pair Name: key-pair-vs-1ip
- Configure default security group to allow traffic to NAT

Instances in the private subnet can establish outbound connections to the Internet via the public subnet using Network Address Translation. (Hourly charges for NAT instances apply)

Creation of public or private subnets will be performed in the background. To check the status view the output window.

Anda dapat melihat VPC baru diVPCtab diAWSExplorer.

Name	VPC ID	State	CIDR	Default	DHCP Options Set	Tenancy
1 myDeploymentVPC	vpc-da0013b3	available	10.0.0.0/16	False	dopt-80cddae9	default

Instans NAT mungkin memerlukan waktu beberapa menit untuk diluncurkan. Bila tersedia, Anda dapat melihatnya dengan memperluas Amazon EC2 simpul di AWSExplorer dan kemudian membuka Instans subnode.

Sesi AWS Elastic Beanstalk Volume (Amazon EBS) dibuat untuk instans NAT secara otomatis. Untuk informasi selengkapnya tentang Elastic Beanstalk, kunjungi [AWS Elastic Beanstalk \(EBS\)](#) di Panduan Pengguna Amazon EC2 untuk Instans Linux.

Instance ID	Status	AMI ID	Type	Security Groups	Zone	Name	Instance Profile	Key Pair Name	Launch Time	Public DNS
1 i-709d9342	running	ami-52ff7262	m1.small	default	us-west-2b	NAT		key-pair-vs-1ip	4/5/2013 9:26:57 AM	

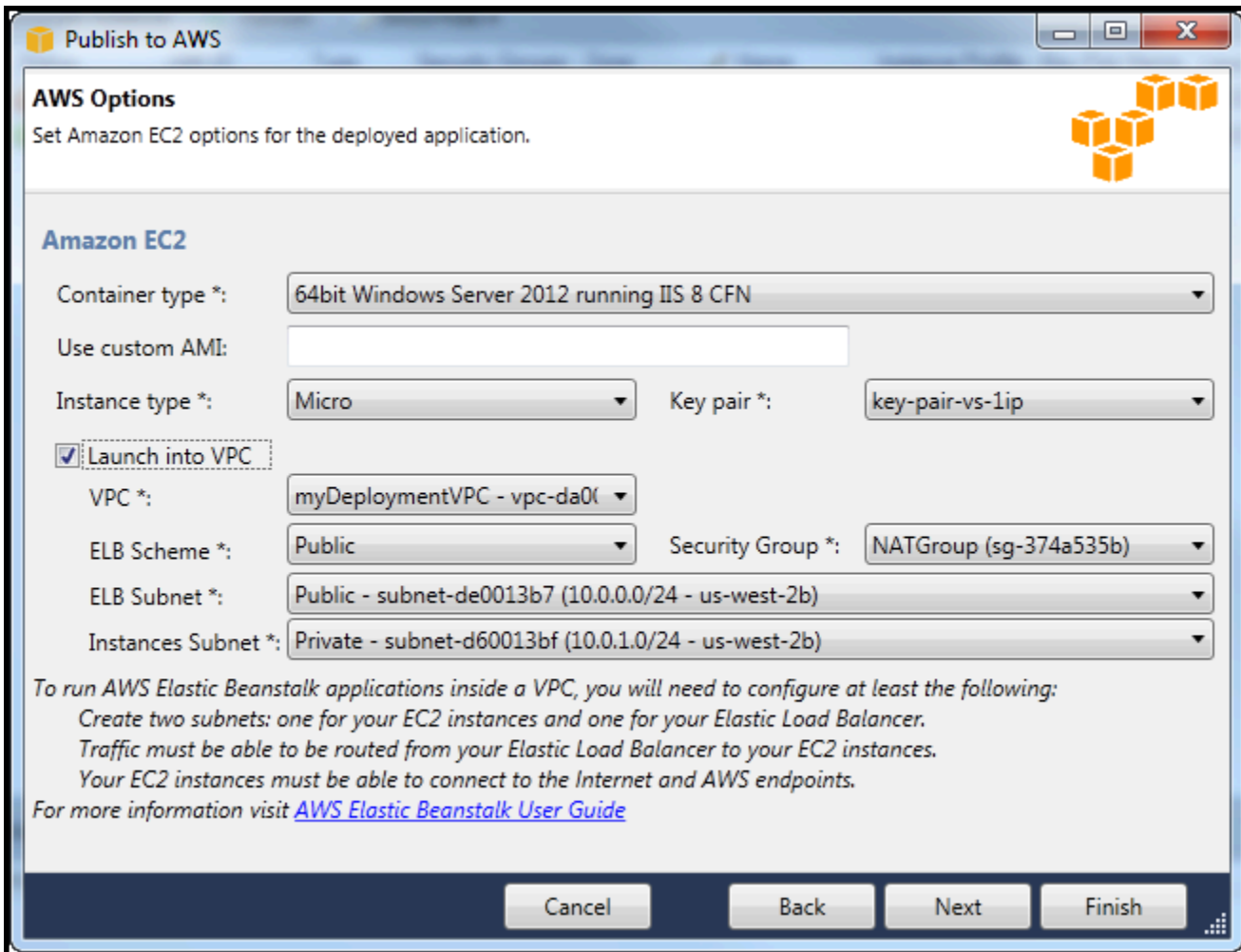
Volume ID	Capacity	Snapshot ID	Created	Zone	Status	Attachment Information	vol-tag
1 vol-da5a91e2	8 GiB	snap-4301d52b	4/5/2013 9:27:00 AM	us-west-2b	in-use	i-709d9342:/dev/sda1 (attached)	

Jika Anda [menyebarkan aplikasi ke AWS Elastic Beanstalk lingkungan](#) dan memilih untuk meluncurkan lingkungan di VPC, Toolkit akan mengisi Publikasikan ke Amazon Web Services kotak dialog dengan informasi konfigurasi untuk VPC Anda.

Toolkit mengisi kotak dialog dengan informasi hanya dari VPC yang dibuat di Toolkit, bukan dari VPC yang dibuat menggunakan AWS Management Console. Hal ini karena ketika Toolkit menciptakan VPC, tag komponen VPC sehingga dapat mengakses informasi mereka.

Screenshot berikut dari Deployment Wizard menunjukkan contoh kotak dialog yang diisi dengan nilai dari VPC yang dibuat di Toolkit.





Untuk menghapus VPC

Untuk menghapus VPC, Anda harus terlebih dahulu menghentikan instans Amazon EC2 di VPC.

1. Jika Anda telah menyebarkan aplikasi keAWS Elastic Beanstalklingkungan di VPC, hapus lingkungan. Tindakan ini akan mengakhiri instans Amazon EC2 yang menjadi hosting aplikasi Anda beserta penyeimbang beban Elastic Load Balancing.

Jika Anda mencoba untuk secara langsung menghentikan instans hosting aplikasi Anda tanpa menghapus lingkungan, layanan Auto Scaling akan secara otomatis membuat instance baru untuk menggantikan yang dihapus. Untuk informasi selengkapnya, kunjungi[Panduan Pengembang Auto Scaling](#).

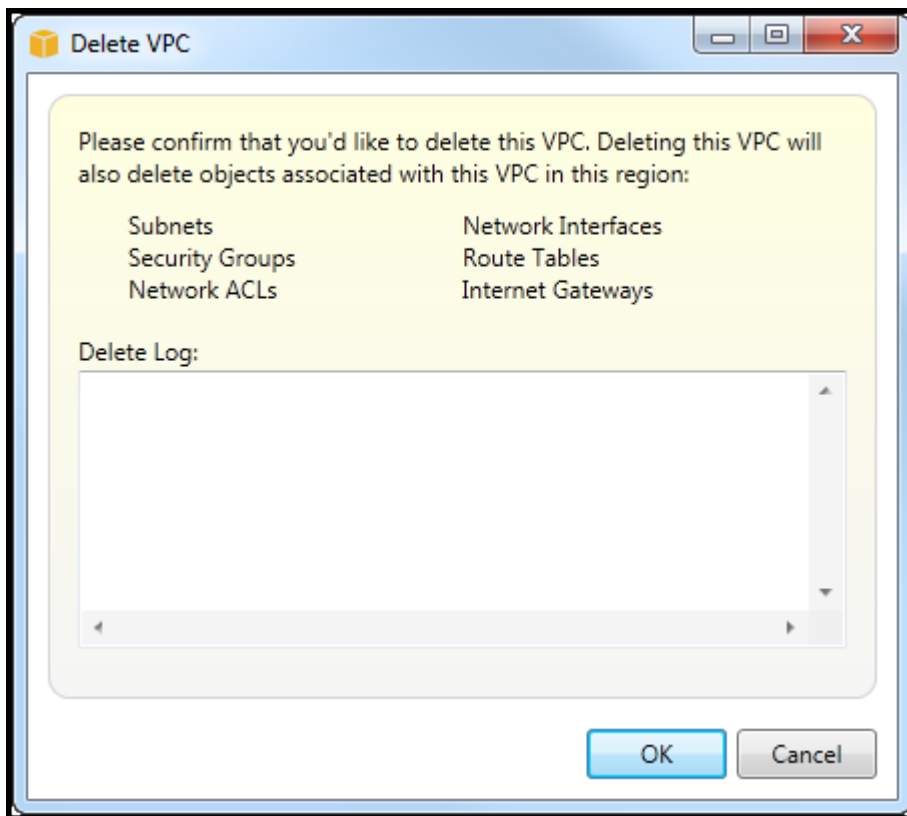
2. Hapus instance NAT untuk VPC.

Anda tidak perlu menghapus volume Amazon EBS yang terkait dengan instans NAT untuk menghapus VPC. Namun, jika Anda tidak menghapus volume, Anda akan terus dikenakan biaya untuk itu bahkan jika Anda menghapus instance NAT dan VPC.

3. Pada VPC tab, pilih Hapus link untuk menghapus VPC.



4. Di Hapus VPC kotak dialog, pilih OKE.



## Menggunakan Editor AWS CloudFormation Template untuk Visual Studio

Toolkit for Visual Studio mencakup AWS CloudFormation editor template AWS CloudFormation dan proyek template untuk Visual Studio. Fitur yang didukung meliputi:

- Membuat template baru (baik kosong atau disalin dari tumpukan atau contoh template yang ada) menggunakan jenis proyek AWS CloudFormation template yang disediakan.
- Mengedit template dengan validasi JSON otomatis, pelengkapan otomatis, pelipatan kode, dan penyorotan sintaks.
- Saran otomatis fungsi intrinsik dan parameter referensi sumber daya untuk nilai bidang di template Anda.
- Item menu untuk melakukan tindakan umum untuk template Anda dari Visual Studio.

## Topik

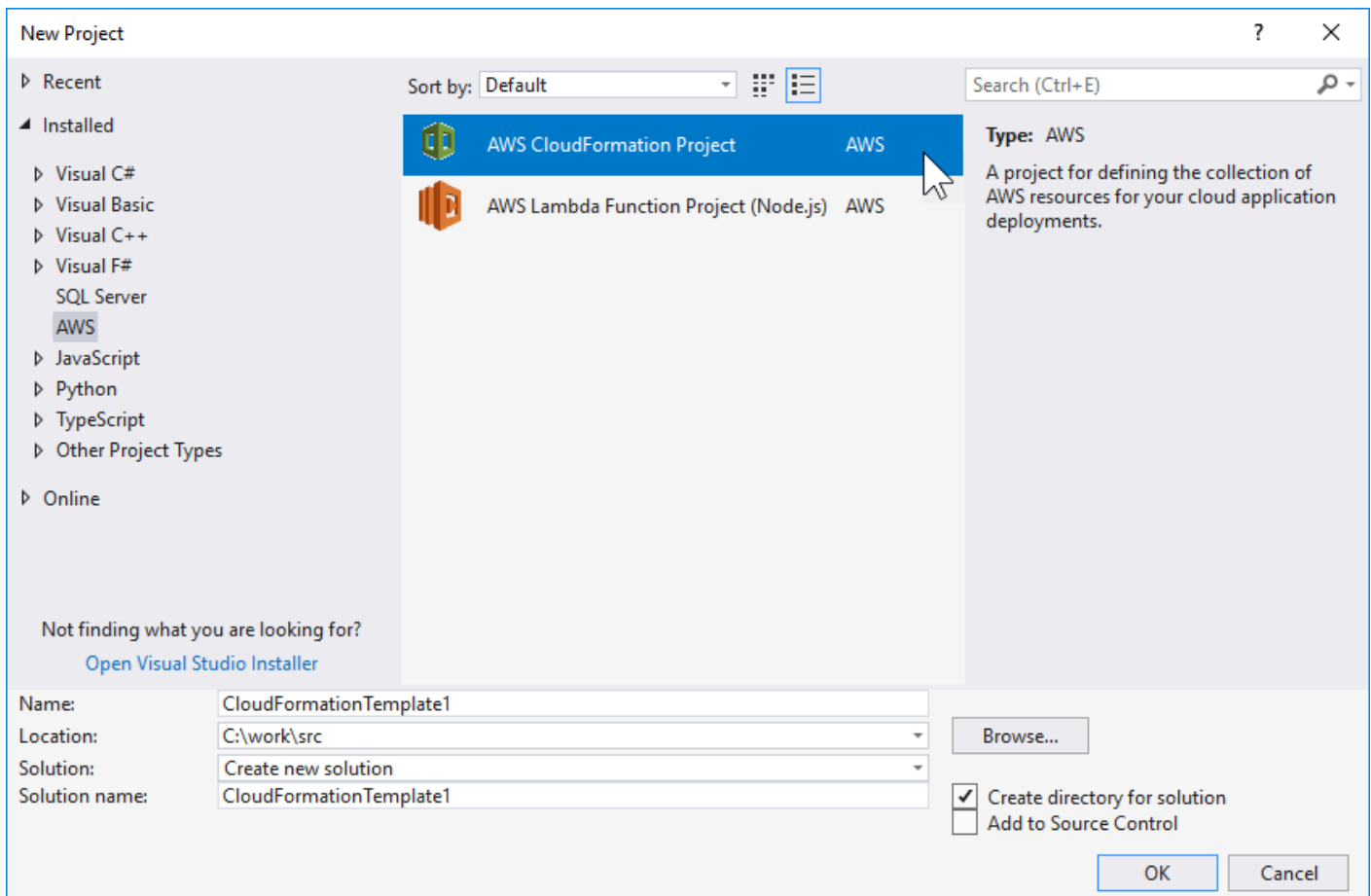
- [MembuatAWS CloudFormationProyek Template di Visual Studio](#)
- [Deploy aAWS CloudFormationTemplate dalam Visual Studio](#)
- [FormatAWS CloudFormationTemplat dalam Visual Studio](#)

## MembuatAWS CloudFormationProyek Template di Visual Studio

Untuk membuat proyek templat

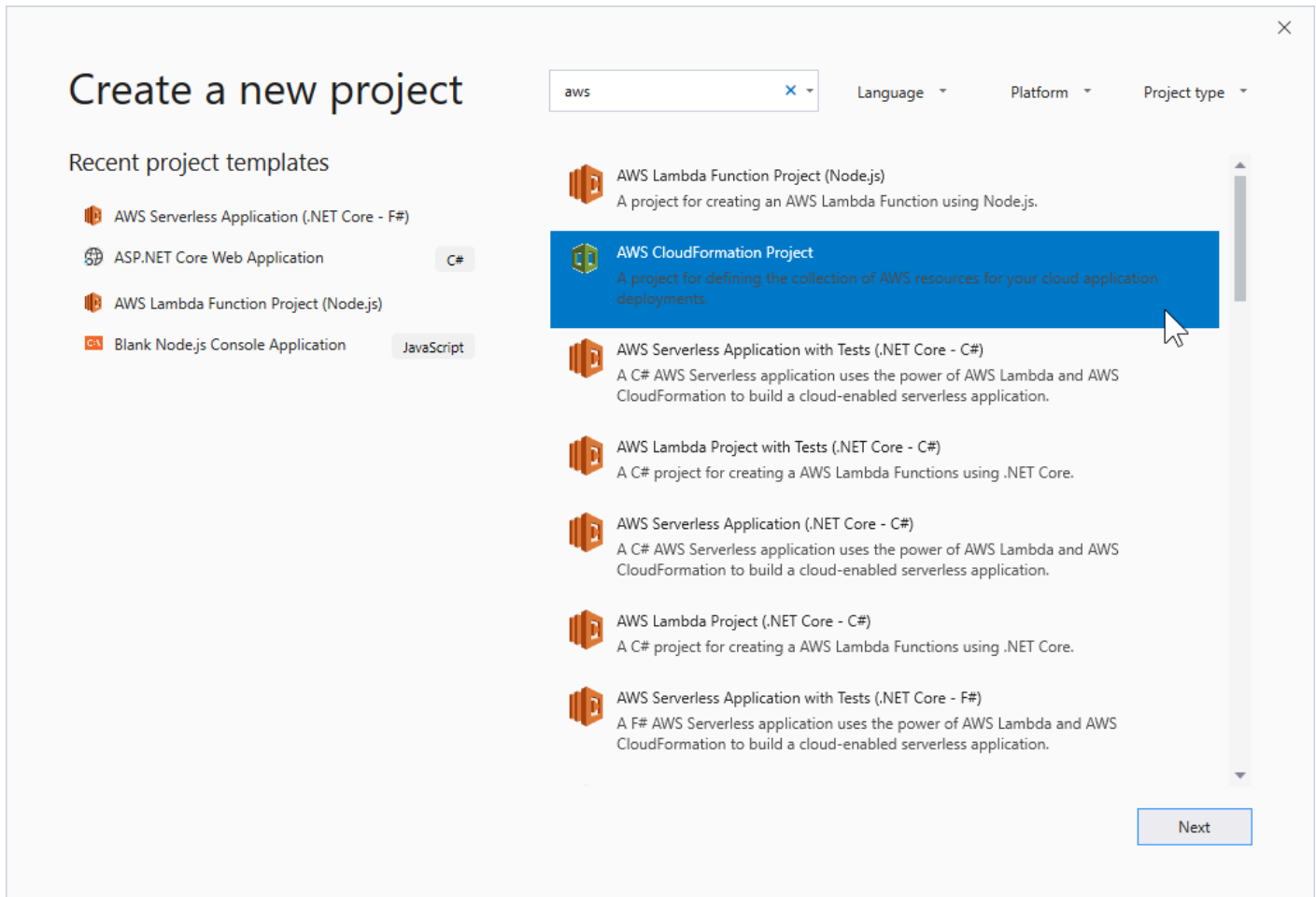
1. Dalam Visual Studio, pilihBerkas, pilihBaru, dan kemudian pilihProyek.
2. Untuk Visual Studio 2017:

DiProyek Barukotak dialog, memperluasTerinstaldan pilihAWS.



Untuk Visual Studio 2019:

Di Proyek Barukotak dialog, memastikan bahwa Bahasa, Platform, dan Tipe proyek kotak drop-down diatur ke "Semua..." dan ketika di Caribidang.



3. Pilih AWS Proyek CloudFormation templat.

4. Untuk Visual Studio 2017:

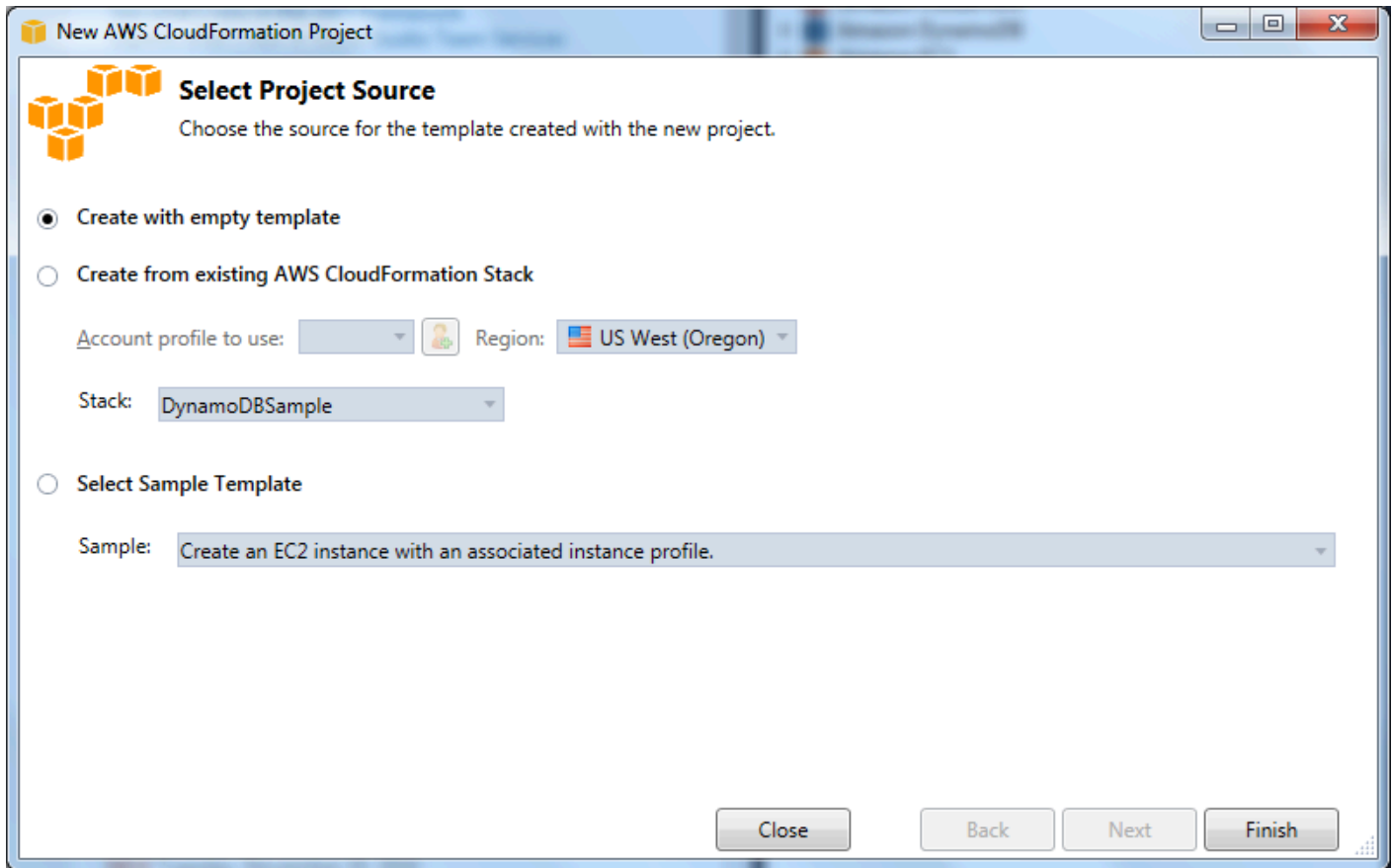
Masukkan yang diinginkan Nama, Lokasi, dll, untuk proyek template Anda, dan kemudian klik OKE.

Untuk Visual Studio 2019:

Klik Selanjutnya. Pada dialog berikutnya, masukkan yang diinginkan Nama, Lokasi, dll, untuk proyek template Anda, dan kemudian klik Buat.

5. Pada Pilih Sumber Proyek halaman, pilih sumber template yang akan Anda buat:

- Buat dengan template kosong menghasilkan yang baru dan kosong AWS CloudFormation templat.
- Buat dari yang ada AWS Tumpukan menghasilkan templat dari tumpukan yang ada di AWS Akun. (Tumpukan tidak perlu memiliki status CREATE\_COMPLETE.)
- Pilih templat contoh menghasilkan template dari salah satu AWS CloudFormation contoh templat.

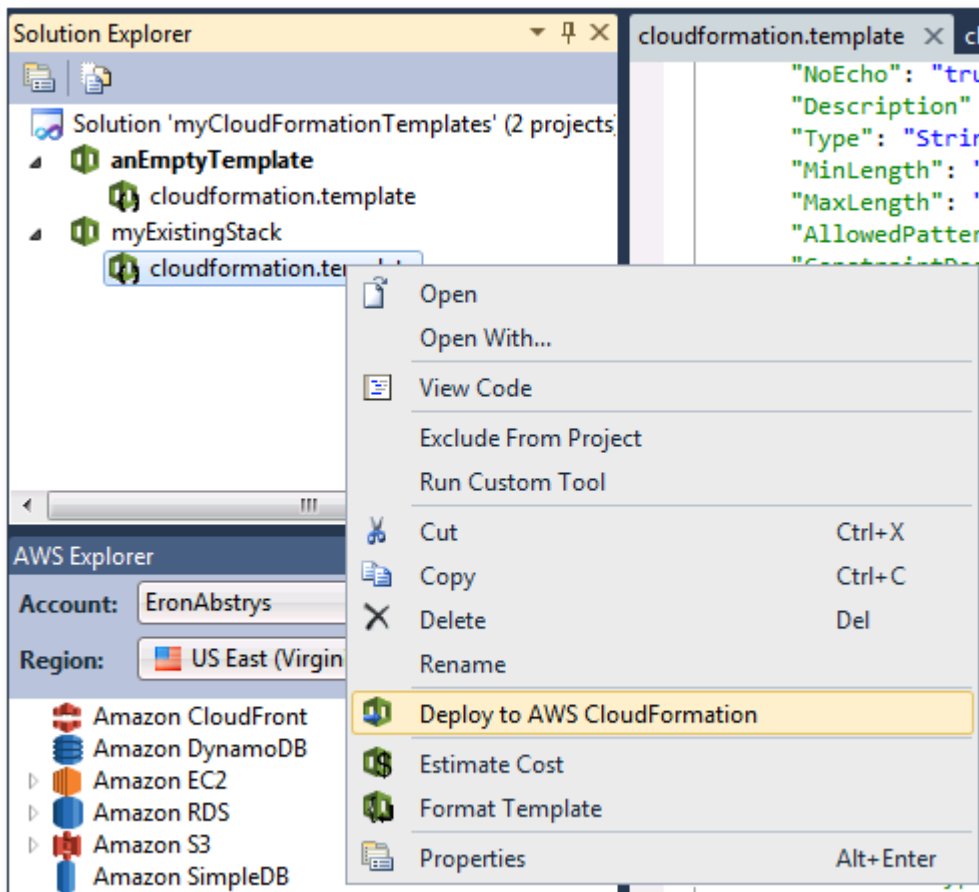


6. Untuk menyelesaikan pembuatan AWS CloudFormation proyek template, pilih Selesai.

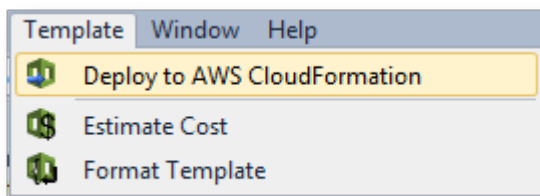
## Deploy a AWS CloudFormation Template dalam Visual Studio

Untuk menyebarkan template CFN

1. Dalam Solution Explorer, buka menu konteks (klik kanan) untuk templat yang ingin Anda gunakan, lalu pilih Deploy AWS CloudFormation.



Atau, untuk menyebarkan template yang sedang Anda edit, dari **Templat** menu, pilih **Deploy AWS CloudFormation**.



2. Pada **Deploy Template** halaman, pilih **Akun AWS** untuk menggunakan untuk meluncurkan tumpukan dan wilayah di mana ia akan diluncurkan.

**Deploy Template**

**Select Template**

To create a stack, fill in the name for your stack and select a template. You may choose one of the sample templates to get started quickly or on your local hard drive.

Account to use: **EronAbstrys** Region: **US East (Virginia)**

**Create New Stack**

SNS Topic (Optional):  **Create New Topic**

Creation Timeout: **None**

Rollback on failure

**Update Existing Stack**

**Cancel** **Back** **Next** **Finish**

3. Pilih Buat Stack Baru dan ketik nama untuk tumpukan Anda.

4. Pilih opsi berikut:

- Untuk menerima notifikasi tentang kemajuan tumpukan, dari Topik SNS daftar drop-down, pilih topik SNS. Anda juga dapat membuat topik SNS dengan memilih Membuat Topik Baru dan mengetik alamat email di dalam kotak.
- Gunakan Waktu Pembuatan untuk menentukan berapa lama AWS CloudFormation harus memungkinkan untuk stack yang akan dibuat sebelum dinyatakan gagal (dan digulung kembali, kecuali Rollback pada kegagalan dibersihkan).
- Gunakan Rollback pada kegagalan jika Anda ingin tumpukan untuk memutar kembali (yaitu, menghapus sendiri) pada kegagalan. Biarkan opsi ini dihapus jika Anda ingin tumpukan tetap aktif untuk tujuan debugging, bahkan jika gagal menyelesaikan peluncuran.

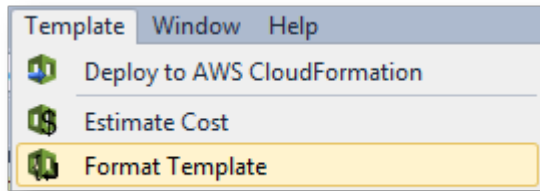
5. Pilih Selesai untuk meluncurkan tumpukan.



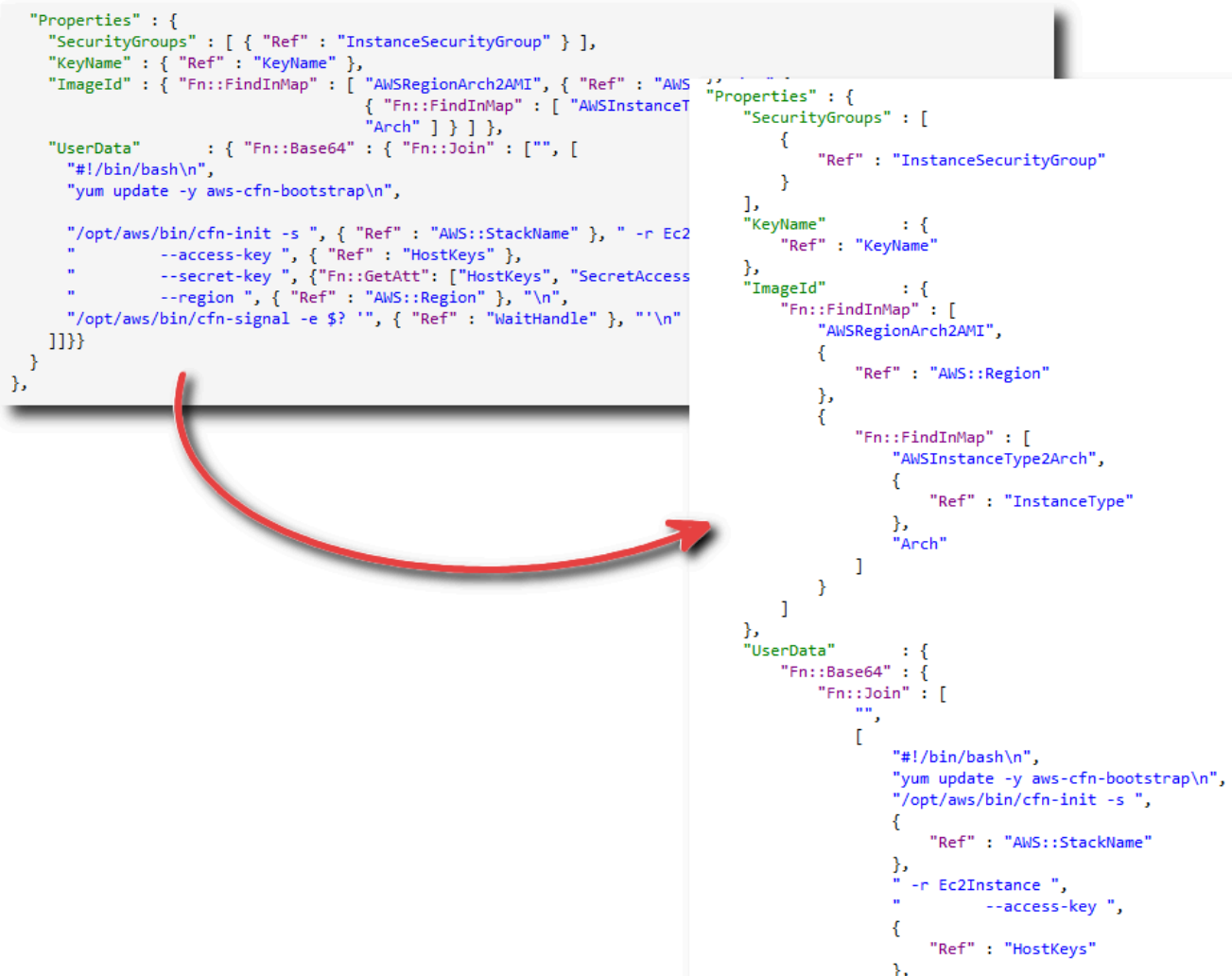
## FormatAWS CloudFormationTemplat dalam Visual Studio

- Dalam Solution Explorer, buka menu konteks (klik kanan) untuk templat dan pilihFormat templat.

Atau, untuk memformat template yang sedang Anda edit, dariTemplatmenu, pilihFormat templat.



Kode JSON Anda akan diformat sehingga strukturnya disajikan dengan jelas.



```
"Properties" : {
  "SecurityGroups" : [ { "Ref" : "InstanceSecurityGroup" } ],
  "KeyName" : { "Ref" : "KeyName" },
  "ImageId" : { "Fn::FindInMap" : [ "AWSRegionArch2AMI", { "Ref" : "AWS
    { "Fn::FindInMap" : [ "AWSInstanceT
      "Arch" ] } ] ] },
  "UserData" : { "Fn::Base64" : { "Fn::Join" : [ "", [
    "#!/bin/bash\n",
    "yum update -y aws-cfn-bootstrap\n",
    "/opt/aws/bin/cfn-init -s ", { "Ref" : "AWS::StackName" }, " -r Ec2
    " --access-key ", { "Ref" : "HostKeys" },
    " --secret-key ", { "Fn::GetAtt" : [ "HostKeys", "SecretAccess
    " --region ", { "Ref" : "AWS::Region" }, "\n",
    "/opt/aws/bin/cfn-signal -e $? ", { "Ref" : "WaitHandle" }, "\n"
  ] ] ] }
}
},
```

```

  "SecurityGroups" : [
    {
      "Ref" : "InstanceSecurityGroup"
    }
  ],
  "KeyName" : {
    "Ref" : "KeyName"
  },
  "ImageId" : {
    "Fn::FindInMap" : [
      "AWSRegionArch2AMI",
      {
        "Ref" : "AWS::Region"
      }
    ],
    {
      "Fn::FindInMap" : [
        "AWSInstanceType2Arch",
        {
          "Ref" : "InstanceType"
        }
      ],
      "Arch"
    }
  ]
},
  "UserData" : {
    "Fn::Base64" : {
      "Fn::Join" : [
        "",
        [
          "#!/bin/bash\n",
          "yum update -y aws-cfn-bootstrap\n",
          "/opt/aws/bin/cfn-init -s ",
          {
            "Ref" : "AWS::StackName"
          },
          " -r Ec2Instance ",
          " --access-key ",
          {
            "Ref" : "HostKeys"
          },

```

## Menggunakan Amazon S3 dariAWSPenjelajah

Amazon Simple Storage Service (Amazon S3) memungkinkan Anda menyimpan dan mengambil data dari koneksi apa pun ke Internet. Semua data yang Anda simpan di Amazon S3 dikaitkan dengan akun Anda dan, secara default, hanya dapat diakses oleh Anda. Toolkit for Visual Studio memungkinkan Anda untuk menyimpan data di Amazon S3 dan untuk melihat, mengelola, mengambil, dan mendistribusikan data tersebut.

Amazon S3 menggunakan konsep ember, yang dapat Anda anggap mirip dengan sistem file atau drive logis. Bucket dapat berisi folder, yang mirip dengan direktori, dan objek, yang mirip dengan file. Pada bagian ini, kita akan menggunakan konsep-konsep ini saat kita berjalan melalui fungsionalitas Amazon S3 yang diekspos oleh Toolkit for Visual Studio.

### Note

Untuk menggunakan alat ini, kebijakan IAM Anda harus memberikan izin untuk `s3:GetBucketAcl`, `s3:GetBucket`, dan `s3:ListBucket` tindakan. Untuk informasi selengkapnya, lihat [IkhtisarAWSKebijakan IAM](#).

## Membuat sebuah Bucket Amazon S3

Bucket adalah unit penyimpanan yang paling mendasar di Amazon S3.

Untuk membuat bucket S3

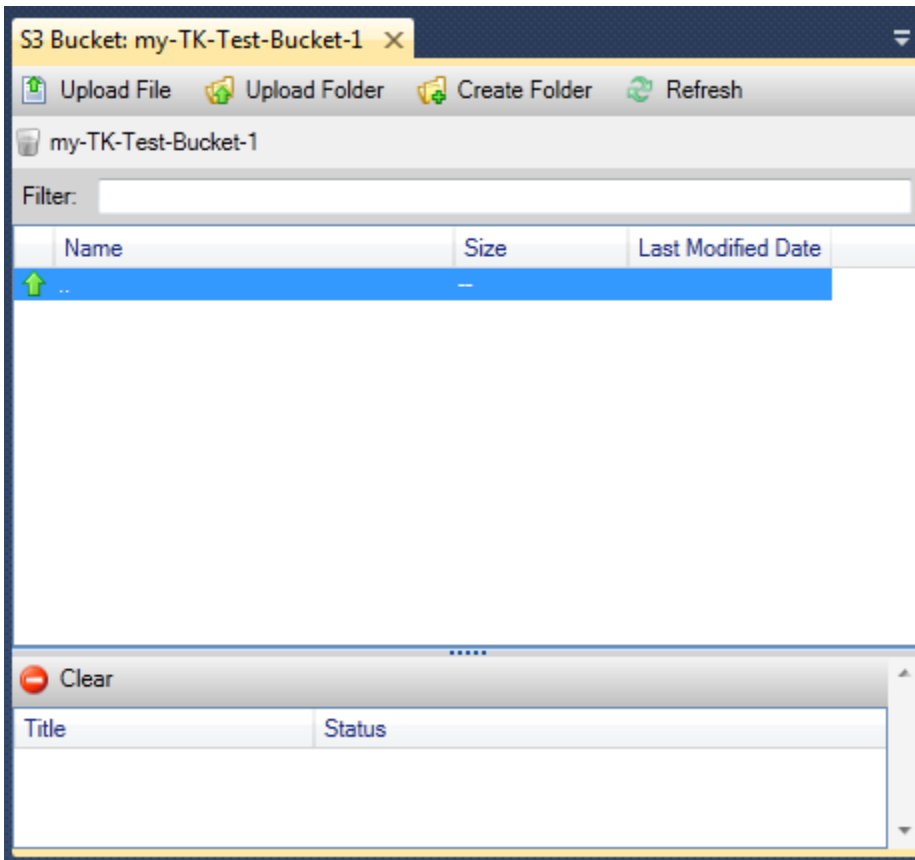
1. MasukAWSExplorer, buka menu konteks (klik kanan) untukAmazon S3node, dan kemudian pilihMembuat Bucket.
2. DiMembuat Bucketkotak dialog, ketik nama untuk bucket. Nama bucket harus unik diAWS. Untuk informasi tentang kendala lainnya, kunjungi[Dokumentasi Amazon S3](#).
3. Pilih OKE.

## Mengelola Bucket Amazon S3AWSPenjelajah

MasukAWSExplorer, operasi berikut tersedia saat Anda membuka menu konteks (klik kanan) untuk bucket Amazon S3.

Jelajahi

Menampilkan tampilan objek yang terkandung dalam ember. Dari sini, Anda dapat membuat folder atau mengunggah file atau seluruh direktori dan folder dari komputer lokal Anda. Panel bawah menampilkan pesan status tentang proses upload. Untuk menghapus pesan ini, pilih **Jelaskan**. Anda juga dapat mengakses tampilan bucket ini dengan mengklik dua kali nama bucket di **AWSExplorer**.



## Properti

Menampilkan kotak dialog di mana Anda dapat melakukan hal berikut:

- Menetapkan izin Amazon S3 yang mencakup:
  - Anda sebagai pemilik ember.
  - semua pengguna yang telah diautentikasi pada AWS.
  - semua orang dengan akses internet.
- Mengaktifkan logging untuk bucket.
- Siapkan notifikasi menggunakan Amazon Simple Notification Service (Amazon SNS) sehingga jika Anda menggunakan Reduced Redundancy Storage (RRS), Anda akan diberi tahu jika kehilangan data terjadi. RRS adalah opsi penyimpanan Amazon S3 yang memberikan daya tahan lebih

sedikit daripada penyimpanan standar, tetapi dengan biaya yang lebih rendah. Untuk informasi selengkapnya, lihat [FAQ S3](#).

- Membuat situs web statis menggunakan data di bucket.

## Kebijakan

Memungkinkan Anda untuk mengatur AWS Identity and Access Management (IAM) kebijakan untuk ember Anda. Untuk informasi selengkapnya, kunjungi [Dokumentasi IAM](#) dan kasus penggunaan untuk [IAM](#) dan [S3](#).

## Buat URL Pra-Ditandatangani

Memungkinkan Anda untuk menghasilkan URL waktu-terbatas Anda dapat mendistribusikan untuk menyediakan akses ke isi bucket. Untuk informasi selengkapnya, lihat [Cara Membuat URL Pra-Ditandatangani](#).

## Lihat Upload Multi-Bagian

Memungkinkan Anda untuk melihat upload multipart. Amazon S3 mendukung pemutusan unggahan objek besar ke beberapa bagian untuk membuat proses upload lebih efisien. Untuk informasi selengkapnya, kunjungi diskusi [upload multipart dalam dokumentasi S3](#).

## Hapus

Memungkinkan Anda untuk menghapus ember. Anda hanya dapat menghapus bucket kosong.

## Mengunggah File dan Folder ke Amazon S3

Anda dapat menggunakan AWSExplorer untuk mentransfer file atau seluruh folder dari komputer lokal Anda ke salah satu ember Anda.

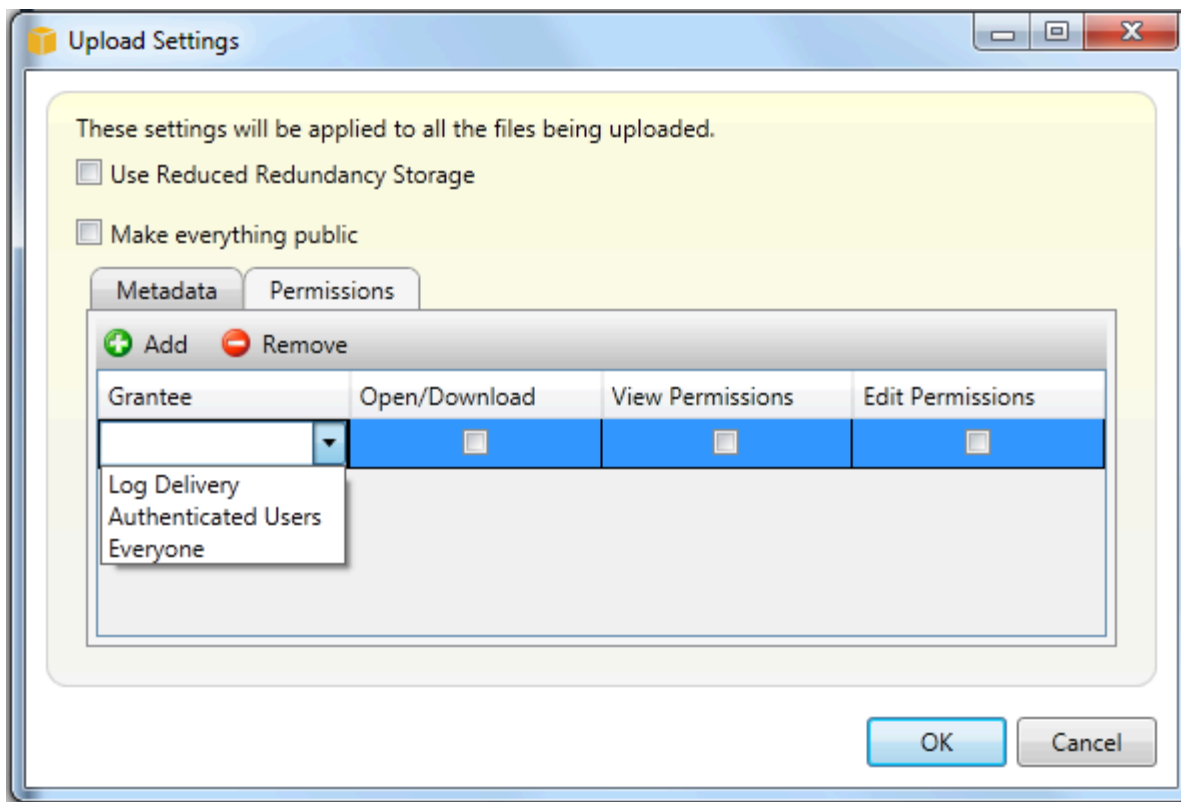
### Note

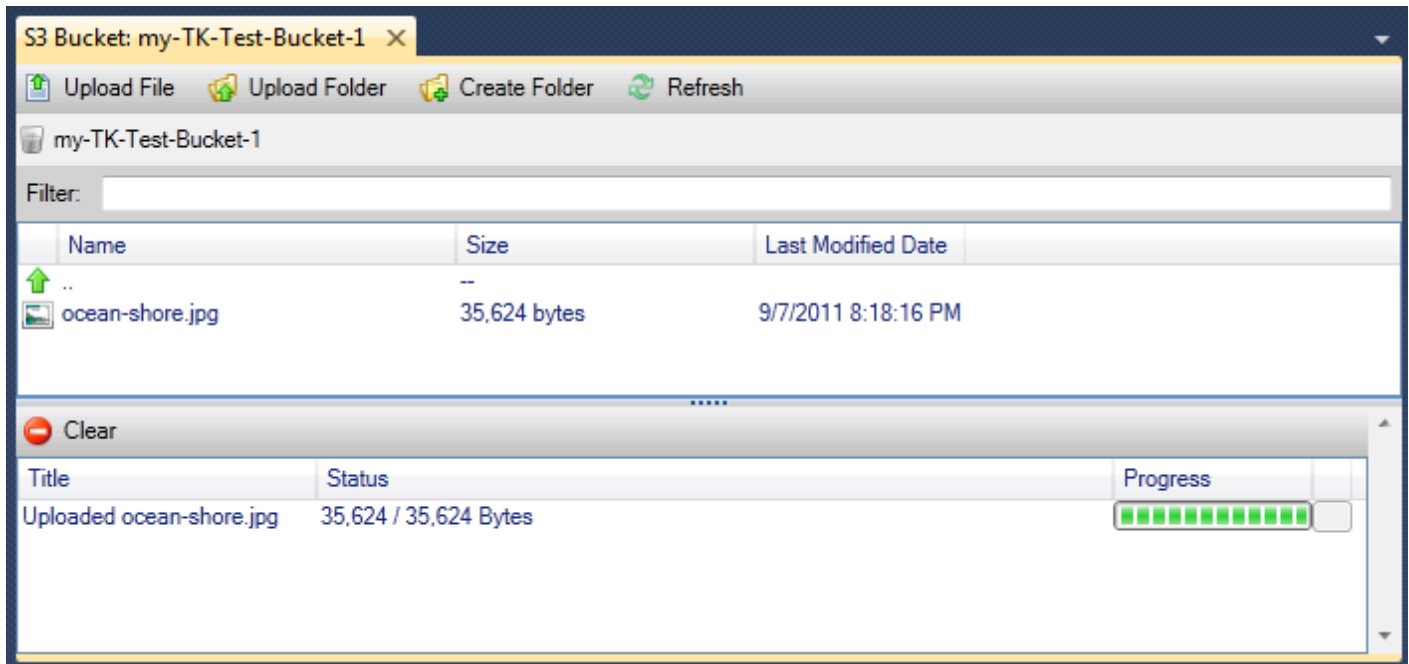
Jika Anda mengunggah file atau folder yang memiliki nama yang sama dengan file atau folder yang sudah ada di bucket Amazon S3, file yang diunggah akan menimpa file yang ada tanpa peringatan.

## Mengunggah File ke S3

1. Masuk **AWSExplorer**, memperluas **Amazon S3 node**, dan klik dua kali bucket atau buka menu konteks (klik kanan) untuk bucket dan pilih **Jelajahi**.
2. Di **Jelajahi** lihat ember Anda, pilih **Mengunggah File** atau **Unggah Folder**.
3. Di **Buka File** kotak dialog, arahkan ke file yang akan diunggah, pilih, lalu pilih **Buka**. Jika Anda mengunggah folder, arahkan ke dan pilih folder itu, lalu pilih **Buka**.

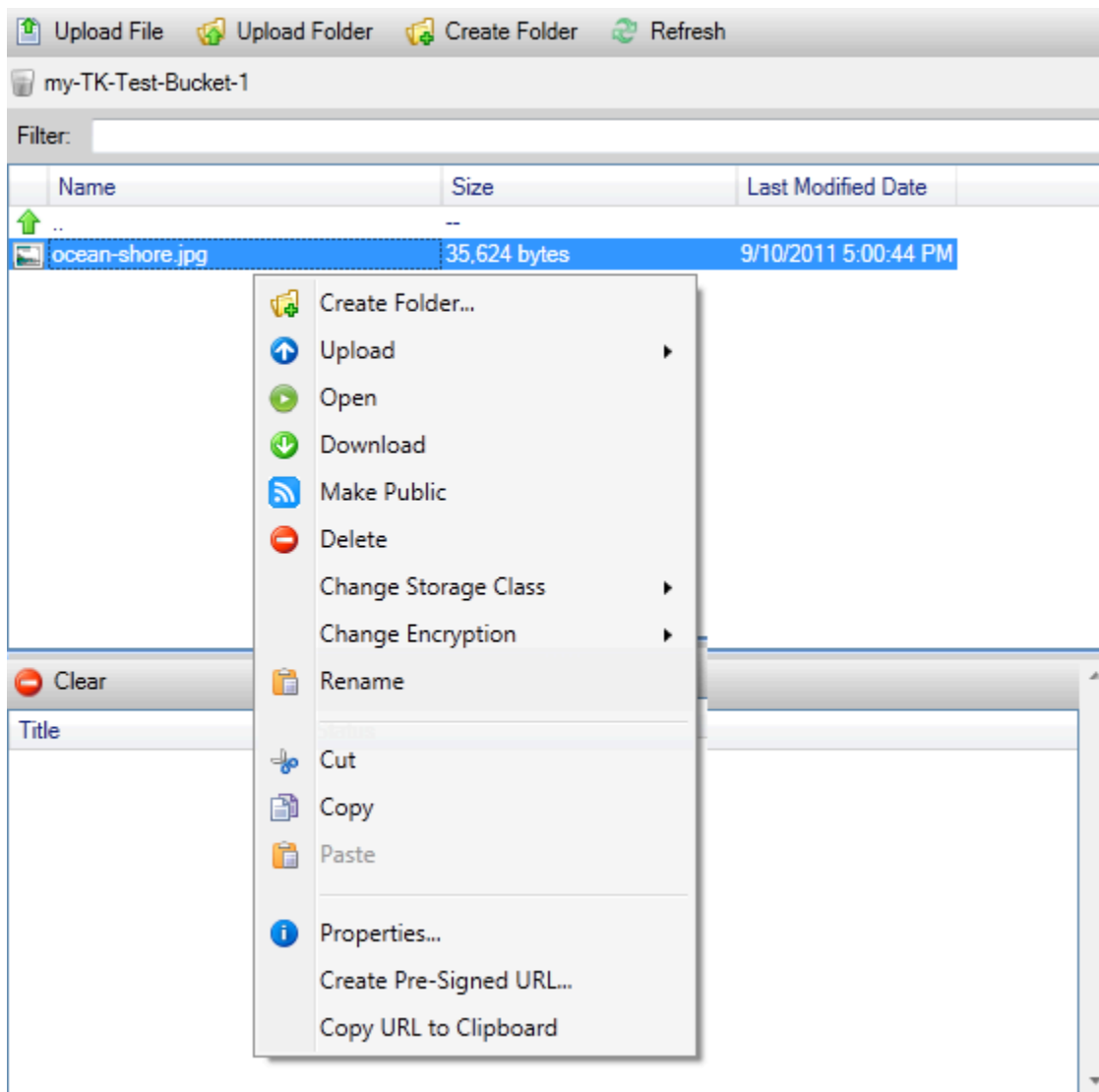
Parameter **Mengunggah Pengaturan** kotak dialog memungkinkan Anda untuk mengatur metadata dan izin pada file atau folder yang Anda upload. Memilih **Jadikan semuanya publik** kotak centang setara dengan pengaturan **Buka/Unduh izin untuk Semua orang**. Anda dapat memilih opsi untuk menggunakan **Penyimpanan Redundansi** untuk file yang diunggah.





## Operasi File Amazon S3AWS Toolkit for Visual Studio

Jika Anda memilih file di tampilan Amazon S3 dan membuka menu konteks (klik kanan), Anda dapat melakukan berbagai operasi pada file.



## Membuat Folder

Memungkinkan Anda untuk membuat folder dalam bucket saat ini. (Setara dengan memilih Membuat Folderlink.)

## Unggah

Memungkinkan Anda untuk mengunggah file atau folder. (Setara dengan memilih Mengunggah FileatauUnggah Folderlink.)

## Buka

Upaya untuk membuka file yang dipilih di browser default Anda. Tergantung pada jenis file dan kemampuan browser default Anda, file mungkin tidak ditampilkan. Ini mungkin hanya didownload oleh browser Anda sebagai gantinya.

## Unduh

Membuka folder kotak dialog untuk memungkinkan Anda mengunduh file yang dipilih.

## Membuat Publik

Menetapkan izin pada file yang dipilih ke Buka/Unduh dan Semua orang. (Setara dengan memilih Jadikan semuanya publik kotak centang di Mengunggah Pengaturan kotak dialog.)

## Hapus

Menghapus file atau folder yang dipilih. Anda juga dapat menghapus file atau folder dengan memilih mereka dan menekan Delete.

## Ganti Kelas Penyimpanan

Menetapkan kelas penyimpanan baik Standard atau Reduced Redundansi Storage (RRS). Untuk melihat pengaturan kelas penyimpanan saat ini, pilih Properti.

## Ubah Enkripsi

Memungkinkan Anda untuk mengatur enkripsi sisi server pada file. Untuk melihat pengaturan enkripsi saat ini, pilih Properti.

## Ubah Nama

Memungkinkan Anda untuk mengubah nama file. Anda tidak dapat mengganti nama folder.

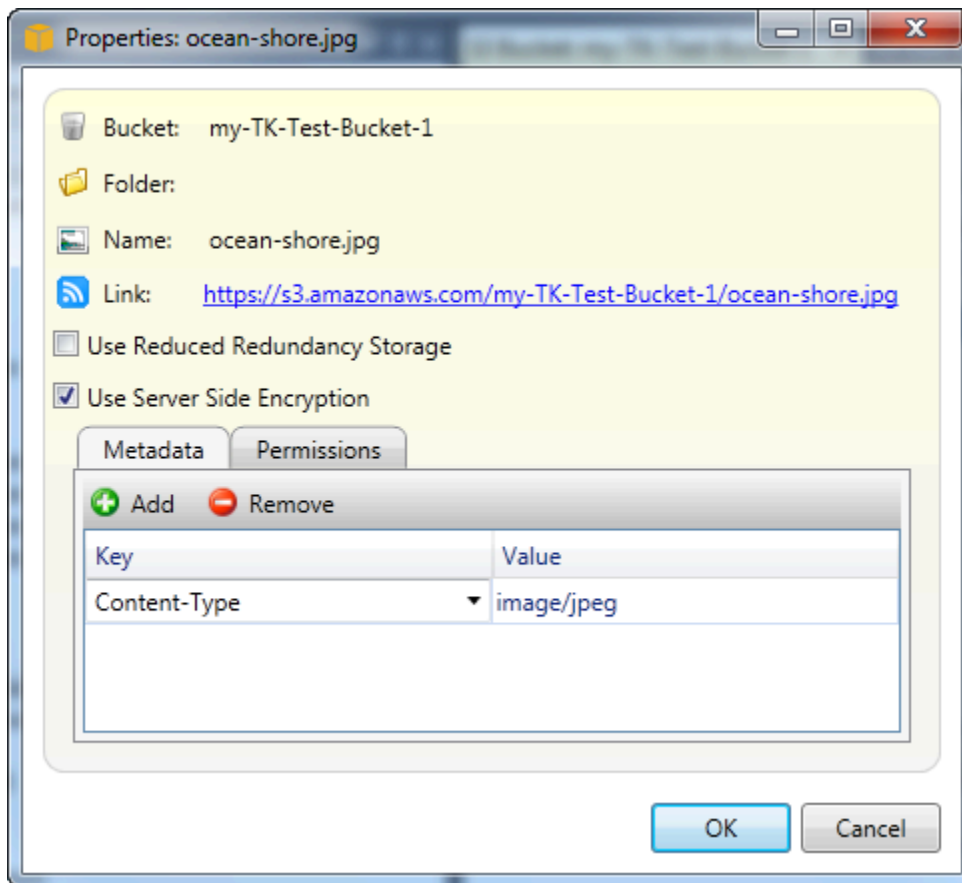
## Potong | Salin | Paste

Memungkinkan Anda untuk memotong, menyalin, dan menempelkan file atau folder antara folder atau antara ember.

## Properti

Menampilkan kotak dialog yang memungkinkan Anda untuk mengatur metadata dan izin untuk file, serta penyimpanan beralih untuk file antara Reduced Redundancy Storage (RRS) dan Standard, dan mengatur enkripsi sisi server untuk file. Kotak dialog ini juga menampilkan tautan https ke file. Jika Anda memilih tautan ini, Toolkit for Visual Studio akan membuka file di browser default Anda. Jika Anda memiliki izin pada file diatur ke Buka/Unduh dan Semua orang, orang lain akan dapat mengakses file melalui link ini. Daripada mendistribusikan tautan ini, kami sarankan Anda membuat dan mendistribusikan URL yang telah ditandatangani sebelumnya.





## Buat URL Pra-Ditandatangani

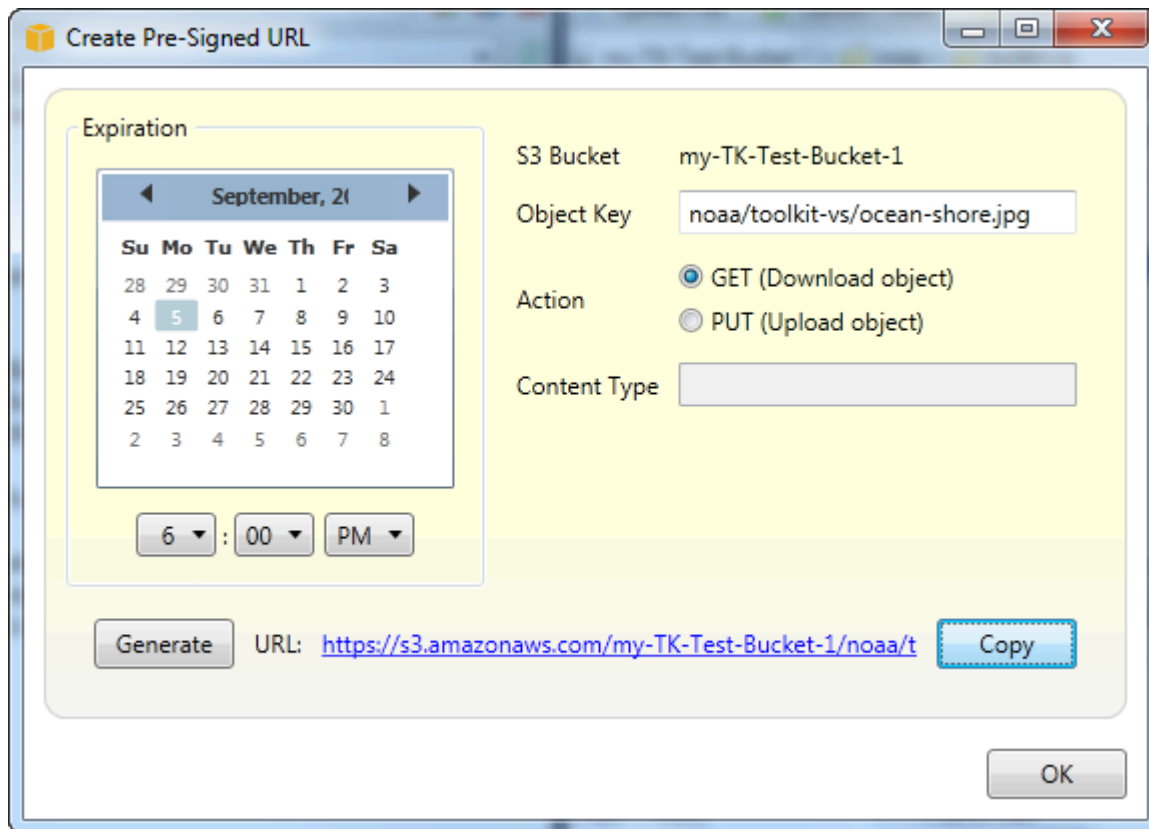
Memungkinkan Anda membuat URL pra-tanda tangan terbatas waktu yang dapat Anda distribusikan untuk memungkinkan orang lain mengakses konten yang telah Anda simpan di Amazon S3.

## Cara Membuat URL Pra-Ditandatangani

Anda dapat membuat URL yang telah ditandatangani sebelumnya untuk bucket atau file dalam bucket. Orang lain kemudian dapat menggunakan URL ini untuk mengakses bucket atau file. URL akan kedaluwarsa setelah jangka waktu yang Anda tentukan saat Anda membuat URL.

Untuk membuat URL yang telah ditandatangani sebelumnya

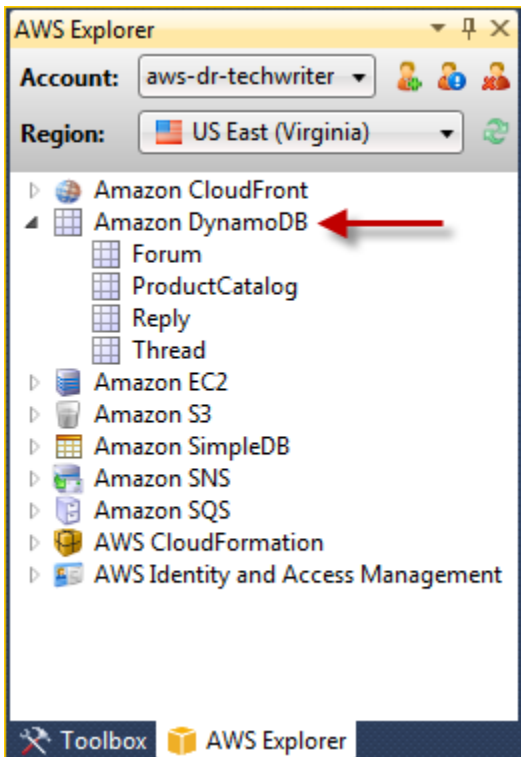
1. DiBuat URL Pra-Ditandatangani kotak dialog, mengatur tanggal kedaluwarsa dan waktu untuk URL. Pengaturan default adalah satu jam dari waktu saat ini.
2. PilihMenghasilkantombol.
3. Untuk menyalin URL ke clipboard, pilihSalin.



## Menggunakan DynamoDB dariAWSPenjelajah

Amazon DynamoDB adalah layanan basis data yang cepat, sangat dapat diskalakan, sangat tersedia, hemat biaya, dan bukan basis data relasional. DynamoDB menghilangkan keterbatasan skalabilitas tradisional pada penyimpanan data sekaligus mempertahankan performa latensi rendah dan dapat diprediksi. Toolkit for Visual Studio menyediakan fungsionalitas untuk bekerja dengan DynamoDB dalam konteks pengembangan. Untuk informasi selengkapnya tentang DynamoDB, lihat [DynamoDB](#) di situs web Amazon Web Services.

Dalam Toolkit for Visual Studio, AWSExplorer menampilkan semua tabel DynamoDB yang terkait dengan akun AWS.



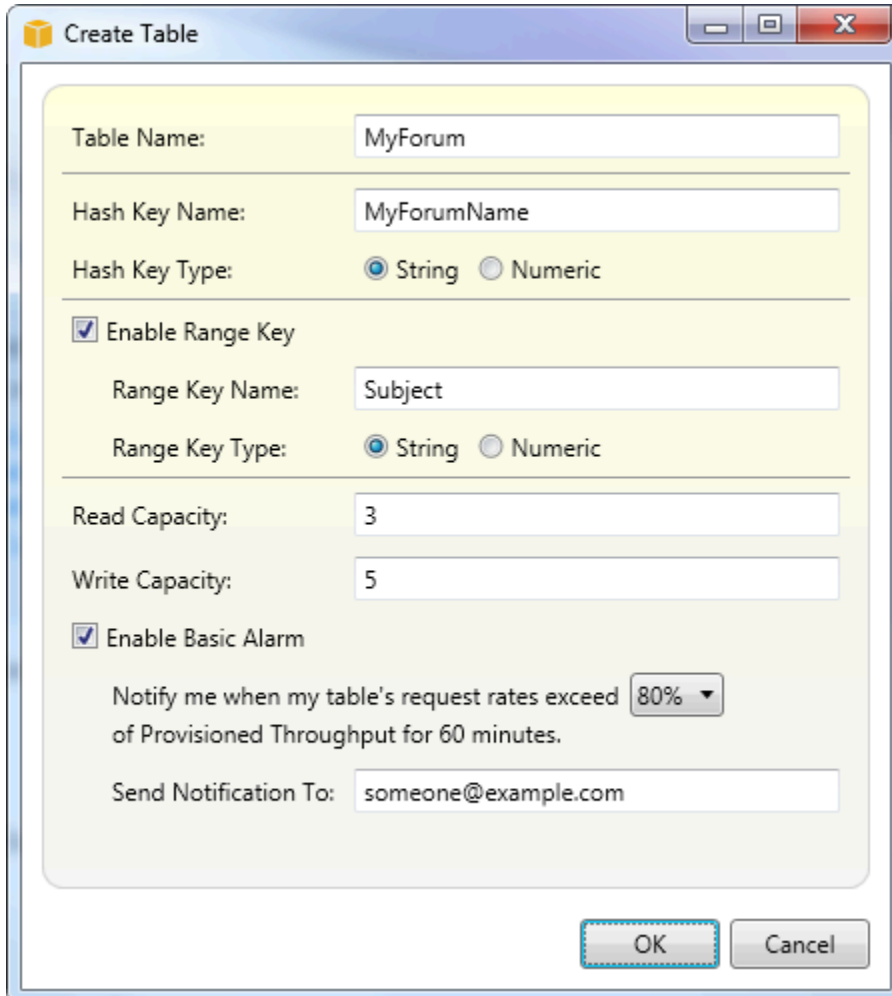
## Membuat Tabel DynamoDB

Anda dapat menggunakan Toolkit for Visual Studio untuk membuat tabel DynamoDB.

Untuk membuat tabel diAWSPenjelajah

1. MasukAWSExplorer, buka menu konteks (klik kanan) untukAmazon DynamoDB, dan kemudian pilihMembuat Tabel.
2. DiMembuat Tabelwizard, diNama Tabel, ketik nama untuk tabel.
3. DiNama Kunci bidang, ketik atribut kunci hash primer dan dariTipe Kunci tombol, pilih tipe kunci hash. DynamoDB membangun indeks hash tak berurutan menggunakan atribut kunci primer dan indeks rentang diurutkan opsional menggunakan atribut kunci primer rentang. Untuk informasi selengkapnya tentang atribut kunci hash primer, buka [atribut kunci hash primer](#), buka [Kunci Primer](#) bagian dalam Panduan Developer Amazon DynamoDB.
4. (Opsional) PilihAktifkan Kunci Rentang. DiNama Kunci bidang, ketik atribut kunci range, dan kemudian dariTipe Kunci tombol, pilih tipe kunci rentang.
5. DiKapasitas Bacabidang, ketik jumlah unit kapasitas baca. DiKapasitas Tulisbidang, ketik jumlah unit kapasitas tulis. Anda harus menentukan minimal tiga unit kapasitas baca dan lima unit kapasitas tulis. Untuk informasi selengkapnya tentang unit kapasitas baca dan tulis, buka [Throughput yang Disediakan di DynamoDB](#).

6. (Opsional) Pilih **Mengaktifkan Alarm Dasar** untuk memberi tahu Anda ketika tarif permintaan tabel Anda terlalu tinggi. Pilih persentase throughput yang disediakan per 60 menit yang harus dilampaui sebelum peringatan dikirim. Dalam **Kirim Pemberitahuan Ke**, ketik alamat email.
7. Klik **OKE** untuk membuat tabel.



The screenshot shows the 'Create Table' dialog box with the following configuration:

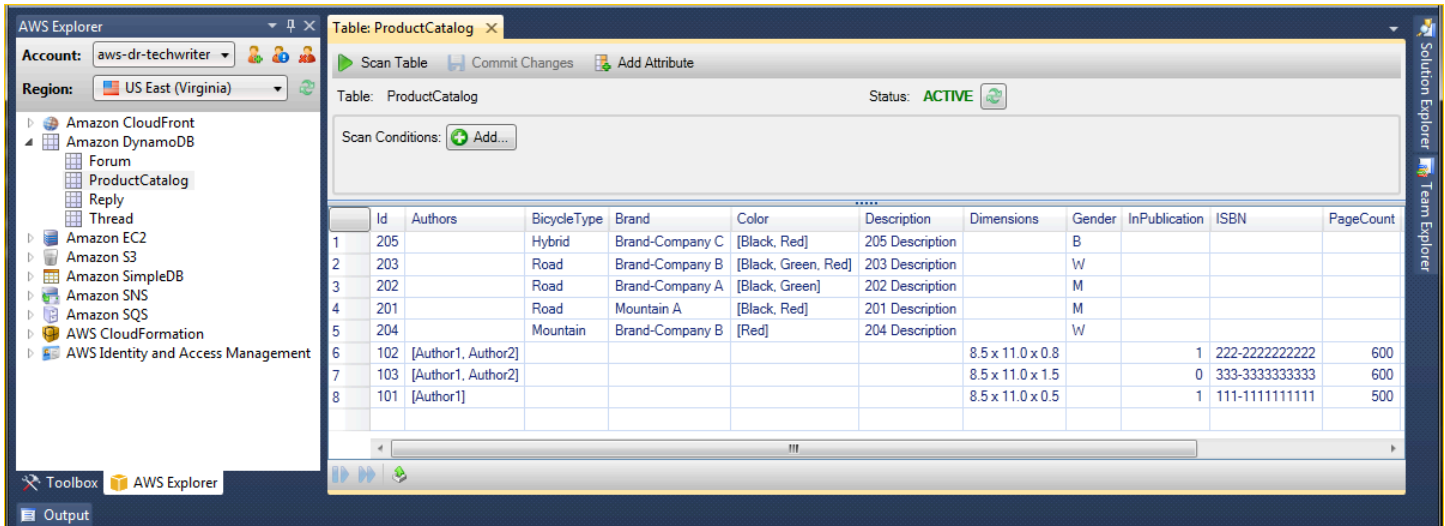
- Table Name: MyForum
- Hash Key Name: MyForumName
- Hash Key Type: String
- Enable Range Key
  - Range Key Name: Subject
  - Range Key Type: String
- Read Capacity: 3
- Write Capacity: 5
- Enable Basic Alarm
  - Notify me when my table's request rates exceed 80% of Provisioned Throughput for 60 minutes.
  - Send Notification To: someone@example.com

Untuk informasi selengkapnya tentang tabel DynamoDB, buka [Konsep Data - Tabel, Item, dan Atribut](#).

## Melihat Tabel DynamoDB sebagai Grid

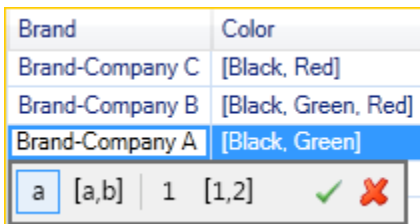
Untuk membuka tampilan grid salah satu tabel DynamoDB Anda, di **AWSExplorer**, klik dua kali subnode yang sesuai dengan tabel. Dari tampilan grid, Anda dapat melihat item, atribut, dan nilai yang tersimpan dalam tabel. Setiap baris sesuai dengan item dalam tabel. Kolom tabel sesuai dengan atribut. Setiap sel tabel memegang nilai-nilai yang terkait dengan atribut untuk item itu.

Atribut dapat memiliki nilai yang string atau angka. Beberapa atribut memiliki nilai yang terdiri darisetstring atau angka. Nilai ditampilkan sebagai daftar yang dipisahkan koma yang diapit tanda kurung.

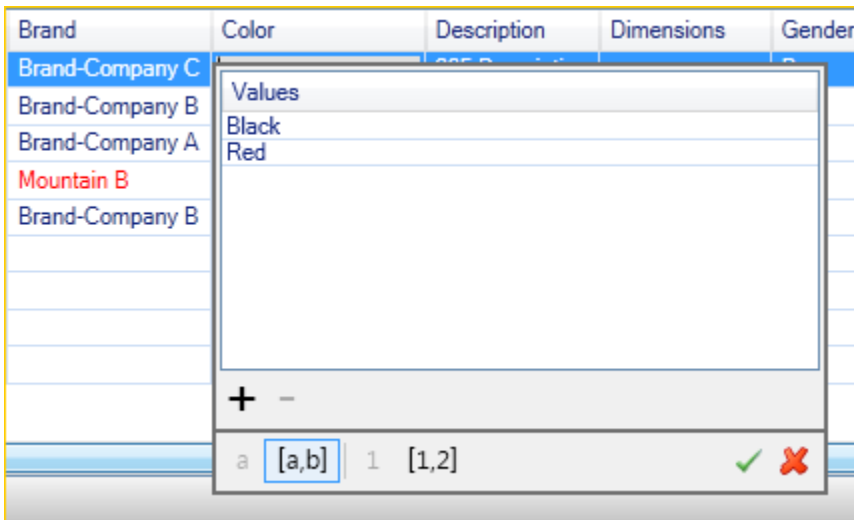


## Mengedit dan Menambahkan Atribut dan Nilai

Dengan mengklik dua kali sel, Anda dapat mengedit nilai untuk atribut item yang sesuai. Untuk atribut set-value, Anda juga dapat menambahkan atau menghapus nilai individual dari set.



Selain mengubah nilai atribut, Anda juga dapat, dengan beberapa keterbatasan, mengubah format nilai untuk atribut. Misalnya, nilai angka dapat dikonversi menjadi nilai string. Jika Anda memiliki nilai string, yang isinya adalah angka, seperti 125, editor sel memungkinkan Anda untuk mengubah format nilai dari string ke nomor. Anda juga dapat mengkonversi nilai tunggal ke set-nilai. Namun, Anda umumnya tidak dapat mengkonversi dari set-nilai ke nilai tunggal; pengecualian adalah ketika set-nilai memiliki, pada kenyataannya, hanya satu elemen dalam set.

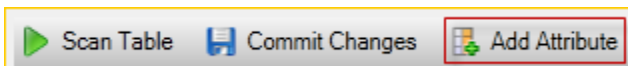


Setelah mengedit nilai atribut, pilih tanda centang hijau untuk mengonfirmasi perubahan Anda. Jika Anda ingin membuang perubahan Anda, pilih X merah.

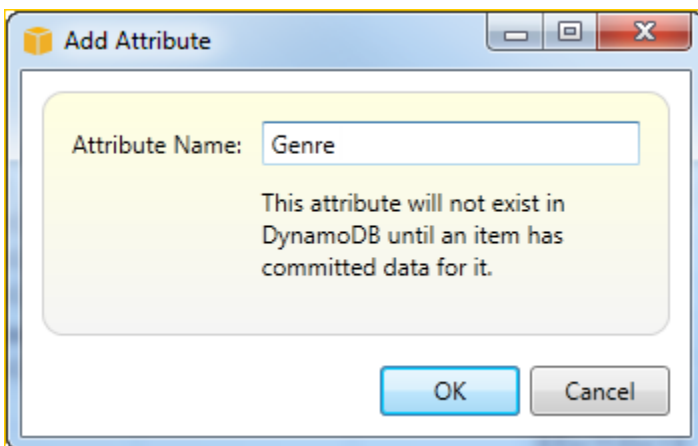
Setelah Anda mengonfirmasi perubahan Anda, nilai atribut akan ditampilkan dalam warna merah. Ini menunjukkan atribut telah diperbarui, tetapi bahwa nilai baru belum ditulis kembali ke database DynamoDB. Untuk menulis perubahan Anda kembali ke DynamoDB, pilih **Perubahan Komit**. Untuk membuang perubahan Anda, pilih **Tabel Pindaian** ketika Toolkit bertanya apakah Anda ingin melakukan perubahan sebelum Pindai, pilih **Tidak**.

### Menambahkan Atribut

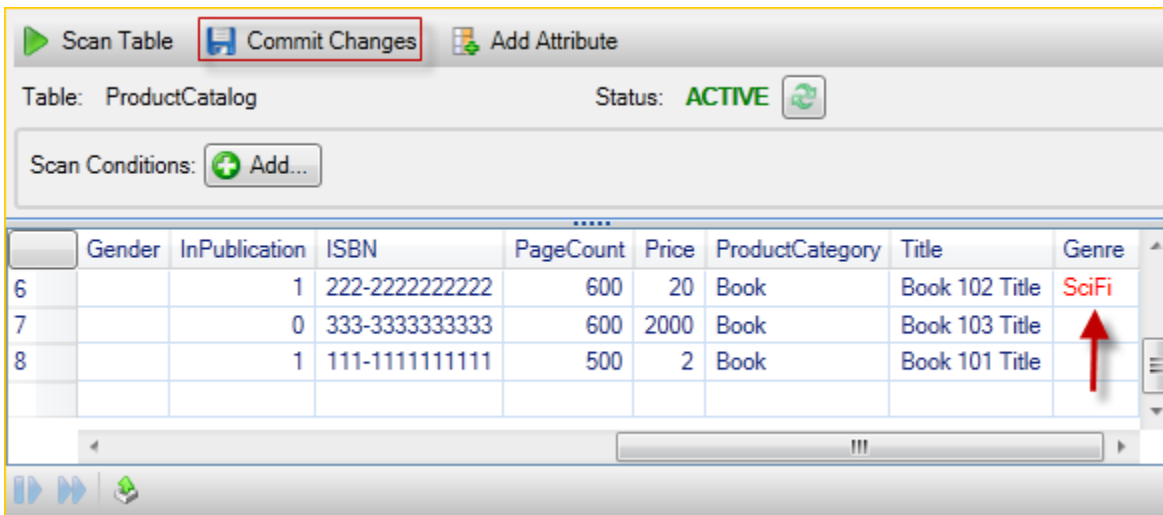
Dari tampilan grid, Anda juga dapat menambahkan atribut ke tabel. Untuk menambahkan atribut baru, pilih **Menambahkan Atribut**.



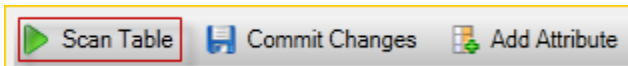
Di **Menambahkan Atribut** kotak dialog, ketik nama untuk atribut Anda, dan kemudian pilih **OKE**.



Untuk membuat atribut baru menjadi bagian dari tabel, Anda harus menambahkan nilai untuk setidaknya satu item dan kemudian memilih Perubahan Komit tombol. Untuk membuang atribut baru, cukup tutup tampilan grid tabel tanpa memilih Perubahan Komit.



## Memindai Tabel DynamoDB

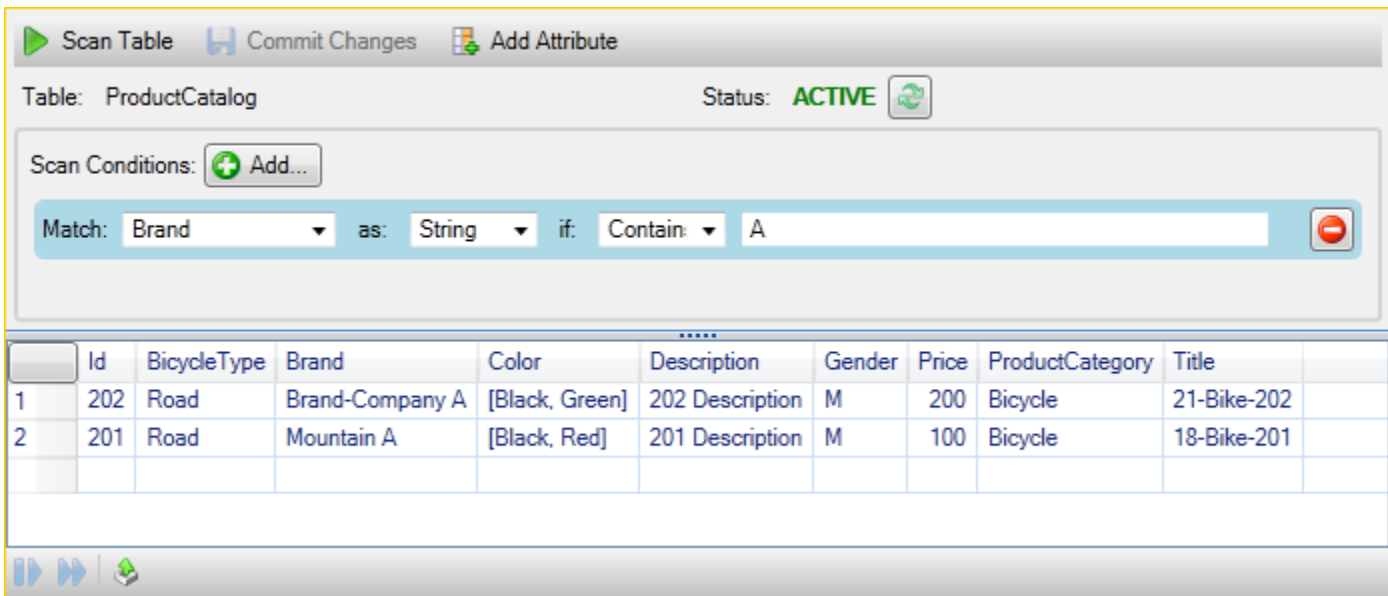


Anda dapat melakukan Scan pada tabel DynamoDB Anda dari Toolkit. Dalam Pindai, Anda menentukan serangkaian kriteria dan Pindai mengembalikan semua item dari tabel yang sesuai dengan kriteria Anda. Pemindaian adalah operasi yang mahal dan harus digunakan dengan hati-hati untuk menghindari mengganggu lalu lintas produksi prioritas yang lebih tinggi di atas meja. Untuk informasi selengkapnya tentang menggunakan operasi Pindai, buka Panduan Developer Amazon DynamoDB.

Untuk melakukan Pindai pada tabel DynamoDB dari AWS Penjelajah

1. Dalam tampilan grid, pilih kondisi pemindaian: tambah tombol.
2. Dalam editor klausa Pindai, pilih atribut yang akan dicocokkan, bagaimana nilai atribut harus ditafsirkan (string, angka, nilai set), bagaimana seharusnya dicocokkan (misalnya Dimulai Dengan atau Berisi), dan nilai literal yang seharusnya cocok.
3. Tambahkan lebih banyak klausal Pindai, sesuai kebutuhan, untuk pencarian Anda. Pemindaian akan mengembalikan hanya item yang sesuai dengan kriteria dari semua klausal Pindai Anda. Scan akan melakukan perbandingan case-sensitive ketika cocok dengan nilai string.
4. Pada bilah tombol di bagian atas tampilan grid, pilih Tabel Pindai.

Untuk menghapus klausa Pindai, pilih tombol merah dengan garis putih di sebelah kanan setiap klausa.



Untuk kembali ke tampilan tabel yang mencakup semua item, hapus semua klausul Pindai dan pilih Tabel Pindailagi.

Memindai Hasil

Di bagian bawah tampilan ada tiga tombol.



Dua tombol biru pertama memberikan pagination untuk hasil Pindai. Tombol pertama akan menampilkan halaman tambahan hasil. Tombol kedua akan menampilkan tambahan sepuluh halaman hasil. Dalam konteks ini, halaman sama dengan 1 MB konten.

Ekspor Hasil Pemindaian ke CSV

Tombol ketiga mengeksport hasil dari Scan saat ini ke file CSV.

## Menggunakan AWS CodeCommit dengan Visual Studio Team Explorer

Anda dapat menggunakan AWS Identity and Access Management (IAM) akun pengguna untuk membuat kredensi Git dan menggunakannya untuk membuat dan mengkloning repositori dari dalam Team Explorer.



## Tipe kredensialnya untukAWS CodeCommit

KebanyakanAWS Toolkit for Visual Studiopengguna menyadari pengaturanAWSprofil kredensial yang berisi akses dan kunci rahasia mereka. Profil kredensialnya ini digunakan dalam Toolkit for Visual Studio untuk mengaktifkan panggilan ke API layanan, misalnya, untuk merilis bucket Amazon S3 diAWSExplorer atau meluncurkan instans Amazon EC2. IntegrasiAWS CodeCommitdengan Team Explorer juga menggunakan profil kredensial ini. Namun, untuk bekerja dengan Git sendiri Anda memerlukan kredensi tambahan, khususnya, kredensi Git untuk koneksi HTTPS. Anda dapat membaca tentang kredensia ini (nama pengguna dan kata sandi) di[Pengaturan untuk Pengguna HTTPS Menggunakan Kredensia Git](#)diAWS CodeCommitPanduan Pengguna.

Anda dapat membuat kredensi Git untukAWS CodeCommithanya untuk akun pengguna IAM. Anda tidak dapat membuatnya untuk akun root. Anda dapat membuat hingga dua set kredensi ini untuk layanan dan, meskipun Anda dapat menandai satu set kredensi sebagai set tidak aktif dan tidak aktif masih dihitung terhadap batas dua set Anda. Perhatikan bahwa Anda dapat menghapus dan membuat kredensialnya kapan saja. Saat Anda menggunakanAWS CodeCommitdari dalam Visual Studio, tradisional AndaAWSkredensia digunakan untuk bekerja dengan layanan itu sendiri, misalnya, ketika Anda membuat dan daftar repositori. Ketika bekerja dengan repositori Git yang sebenarnya dihosting diAWS CodeCommit, Anda menggunakan kredensi Git.

Sebagai bagian dari dukungan untukAWS CodeCommit, Toolkit for Visual Studio secara otomatis membuat dan mengelola kredensi Git ini untuk Anda dan mengaitkannya dengan AndaAWSprofil kredensial. Anda tidak perlu khawatir tentang memiliki set kredensi yang tepat untuk melakukan operasi Git dalam Team Explorer. Setelah Anda terhubung ke Team Explorer denganAWSprofil kredensial, kredensi Git terkait digunakan secara otomatis setiap kali Anda bekerja dengan remote Git.

## Terhubung keAWS CodeCommit

Ketika Anda membuka jendela Team Explorer di Visual Studio 2015 atau yang lebih baru, Anda akan melihatAWS CodeCommitentri di bagian Penyedia Layanan yang Dihosting di Kelola Koneksi.



AWS CodeCommit  
Amazon, Inc.

AWS CodeCommit is a fully-managed source control service that makes it easy for companies to host secure and highly scalable private Git repositories.

[Connect...](#)

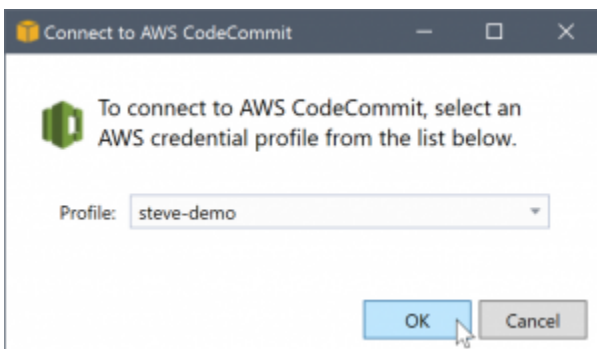
[Sign up](#)

MemilihDaftarmembuka halaman beranda Amazon Web Services di jendela peramban. Apa yang terjadi ketika Anda memilihHubungkantergantung pada apakah Toolkit for Visual Studio dapat

menemukan profil kredensial dengan AWS akses dan kunci rahasia untuk memungkinkannya melakukan panggilan ke AWS atas nama Anda. Anda mungkin telah menyiapkan profil kredensial dengan menggunakan halaman Memulai baru yang ditampilkan di IDE ketika Toolkit for Visual Studio tidak dapat menemukan kredensi yang disimpan secara lokal. Atau Anda mungkin telah menggunakan Toolkit for Visual Studio, yang AWS Tools for Windows PowerShell, atau AWS CLI dan sudah memiliki AWS profil kredensialnya tersedia untuk Toolkit for Visual Studio untuk digunakan.

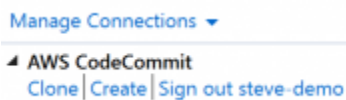
Bila Anda memilih Hubungkan, Toolkit for Visual Studio memulai proses untuk menemukan profil kredensial untuk digunakan dalam koneksi. Jika Toolkit for Visual Studio tidak dapat menemukan profil kredensial, itu akan membuka kotak dialog yang mengundang Anda untuk memasukkan akses dan kunci rahasia untuk Akun AWS. Kami sangat merekomendasikan agar Anda menggunakan akun pengguna IAM, dan bukan kredensialnya. Selain itu, seperti yang disebutkan sebelumnya, kredensi Git yang akhirnya Anda butuhkan hanya dapat dibuat untuk pengguna IAM. Setelah akses dan kunci rahasia disediakan dan profil kredensial dibuat, hubungan antara Team Explorer dan AWS CodeCommit siap digunakan.

Jika Toolkit for Visual Studio menemukan lebih dari satu AWS profil kredensialnya, Anda diminta untuk memilih akun yang ingin Anda gunakan dalam Team Explorer.



Jika Anda hanya memiliki satu profil kredensial, Toolkit for Visual Studio melewati kotak dialog pemilihan profil dan Anda terhubung segera:

Ketika koneksi dibuat antara Team Explorer dan AWS CodeCommit melalui profil kredensial Anda, kotak dialog undangan ditutup dan panel koneksi ditampilkan.



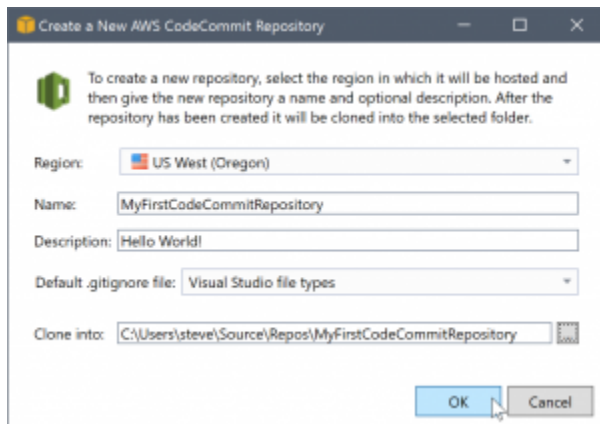
Karena Anda tidak memiliki repositori yang dikloning secara lokal, panel hanya menampilkan operasi yang dapat Anda lakukan: Klon, Buat, dan Keluar. Seperti penyedia lainnya, AWS CodeCommit di Team Explorer dapat terikat hanya satu AWS profil kredensialnya pada waktu tertentu. Untuk beralih

akun, Anda menggunakan Keluar untuk menghapus koneksi sehingga Anda dapat memulai koneksi baru menggunakan akun yang berbeda.

Sekarang setelah Anda membuat koneksi, Anda dapat membuat repositori dengan mengklik **Buat link**.

## Membuat Repositori

Ketika Anda mengklik **Buat link**, **Membuat Baru AWS CodeCommit Repositori** kotak dialog terbuka.



AWS CodeCommit repositori diatur oleh wilayah, jadi di Wilayah Anda dapat memilih wilayah di mana untuk meng-host repositori. Daftar ini memiliki semua daerah di mana AWS CodeCommit didukung. Anda memberikan Nama (wajib) dan Deskripsi (opsional) untuk repositori baru kami.

Perilaku default kotak dialog adalah untuk akhiran lokasi folder untuk repositori baru dengan nama repositori (saat Anda memasukkan nama, lokasi folder juga diperbarui). Untuk menggunakan nama folder yang berbeda, edit Mengkloning ke path folder setelah Anda selesai memasukkan nama repositori.

Anda juga dapat memilih untuk secara otomatis membuat sebuah awal `.gitignore` file untuk repositori. Parameter AWS Toolkit for Visual Studio menyediakan bawaan bawaan untuk jenis file Visual Studio. Anda juga dapat memilih untuk tidak memiliki file atau menggunakan file kustom yang ada yang ingin Anda gunakan kembali di seluruh repositori. Cukup pilih Menggunakan kustom dalam daftar dan arahkan ke file kustom untuk digunakan.

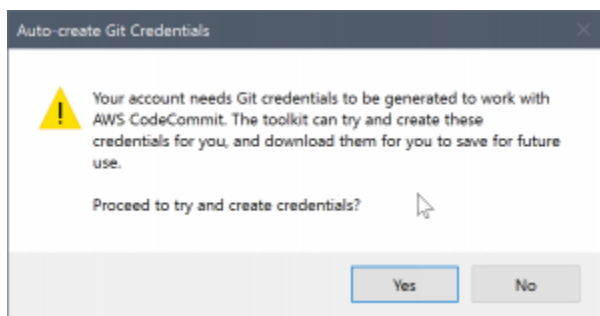
Setelah Anda memiliki nama repositori dan lokasi, Anda siap untuk mengklik **OK** dan mulai membuat repositori. Toolkit for Visual Studio meminta agar layanan membuat repositori dan kemudian mengkloning repositori baru secara lokal, menambahkan komit awal untuk file `.gitignore`, jika Anda menggunakannya. Pada titik inilah Anda mulai bekerja dengan remote Git, jadi Toolkit for Visual Studio sekarang membutuhkan akses ke kredensi Git yang dijelaskan sebelumnya.

## Menyiapkan Kredensialnya Git

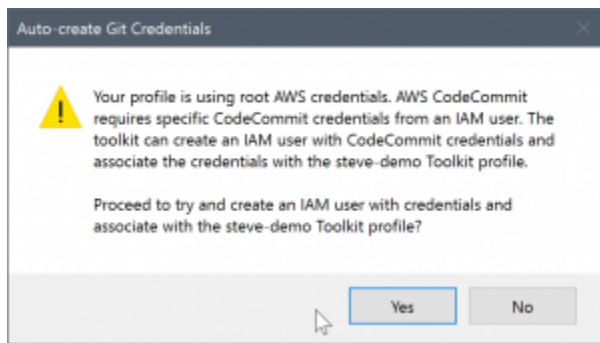
Untuk titik ini Anda telah menggunakan AWS Akses dan kunci rahasia untuk meminta layanan membuat repositori Anda. Sekarang Anda perlu bekerja dengan Git sendiri untuk melakukan operasi klon yang sebenarnya, dan Git tidak mengerti AWS Akses dan kunci rahasia. Sebagai gantinya, Anda perlu menyediakan kredensi nama pengguna dan kata sandi ke Git untuk digunakan pada koneksi HTTPS dengan remote.

Seperti dicatat dalam [Menyiapkan kredensialnya](#), kredensialnya Git yang akan Anda gunakan harus dikaitkan dengan pengguna IAM. Anda tidak dapat membuat mereka untuk kredensi root. Anda harus selalu menyiapkan AWS profil kredensial berisi akses pengguna IAM dan kunci rahasia, dan bukan kunci root. Toolkit for Visual Studio dapat mencoba untuk mengatur kredensi Git untuk AWS CodeCommit bagimu dan kaitkan mereka dengan AWS profil kredensial yang digunakan untuk terhubung di Team Explorer sebelumnya.

Bila Anda memilih OKE di Membuat Baru AWS CodeCommit Repositori kotak dialog dan berhasil membuat repositori, Toolkit for Visual Studio memeriksa AWS profil kredensial yang terhubung di Team Explorer untuk menentukan apakah kredensi Git untuk AWS CodeCommit ada dan terkait secara lokal dengan profil. Jika demikian, Toolkit for Visual Studio menginstruksikan Team Explorer untuk memulai operasi klon pada repositori baru. Jika kredensi Git tidak tersedia secara lokal, Toolkit for Visual Studio memeriksa jenis kredensi akun yang digunakan dalam koneksi di Team Explorer. Jika kredensialnya adalah untuk pengguna IAM, seperti yang kami sarankan, pesan berikut akan ditampilkan.

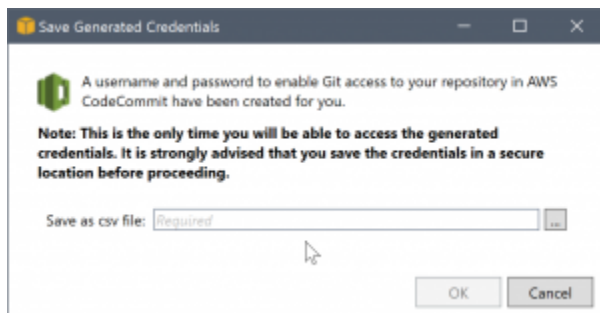


Jika kredensialnya adalah kredensia root, pesan berikut akan ditampilkan sebagai gantinya.



Dalam kedua kasus tersebut, Toolkit for Visual Studio menawarkan untuk mencoba melakukan pekerjaan untuk membuat kredensi Git yang diperlukan untuk Anda. Dalam skenario pertama, semua yang dibutuhkan untuk membuat adalah satu set kredensi Git untuk pengguna IAM. Ketika akun root digunakan, Toolkit for Visual Studio pertama mencoba untuk membuat pengguna IAM dan kemudian melanjutkan untuk membuat kredensi Git untuk pengguna baru itu. Jika Toolkit for Visual Studio harus membuat pengguna baru, itu berlaku AWS CodeCommit Power User mengelola kebijakan ke akun pengguna baru tersebut. Kebijakan ini memungkinkan akses hanya ke AWS CodeCommit dan memungkinkan semua operasi yang akan dilakukan dengan AWS CodeCommit kecuali untuk penghapusan repositori.

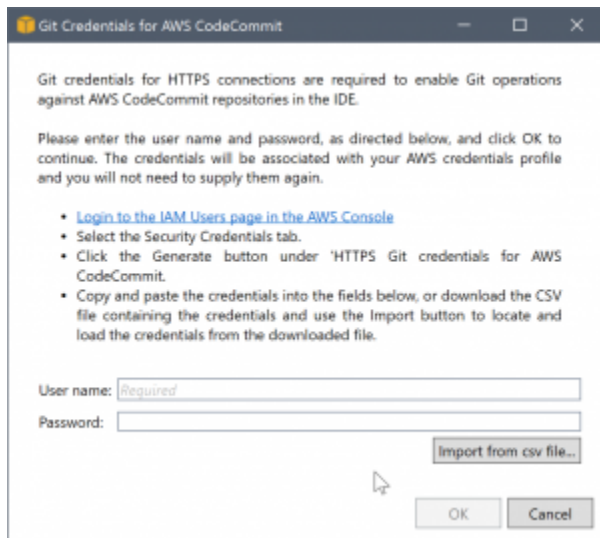
Saat membuat kredensi, Anda hanya dapat melihatnya satu kali. Oleh karena itu, Toolkit for Visual Studio meminta Anda untuk menyimpan kredensi yang baru dibuat sebagai `.csv` mengajukan sebelum melanjutkan.



Ini adalah sesuatu yang kami juga sangat menyarankan, dan pastikan untuk menyimpannya ke lokasi yang aman!

Mungkin ada kasus di mana Toolkit for Visual Studio tidak dapat secara otomatis membuat kredensi. Misalnya, Anda mungkin telah membuat jumlah maksimum set kredensi Git untuk AWS CodeCommit (dua), atau Anda mungkin tidak memiliki hak program yang cukup untuk Toolkit for Visual Studio untuk melakukan pekerjaan untuk Anda (jika Anda masuk sebagai pengguna IAM). Dalam kasus ini, Anda dapat masuk ke AWS Management Console untuk mengelola kredensi atau

mendapatkannya dari administrator Anda. Anda kemudian dapat memasukkannya ke dalam Kredensi Git untuk AWS CodeCommit kotak dialog, yang menampilkan Toolkit for Visual Studio.

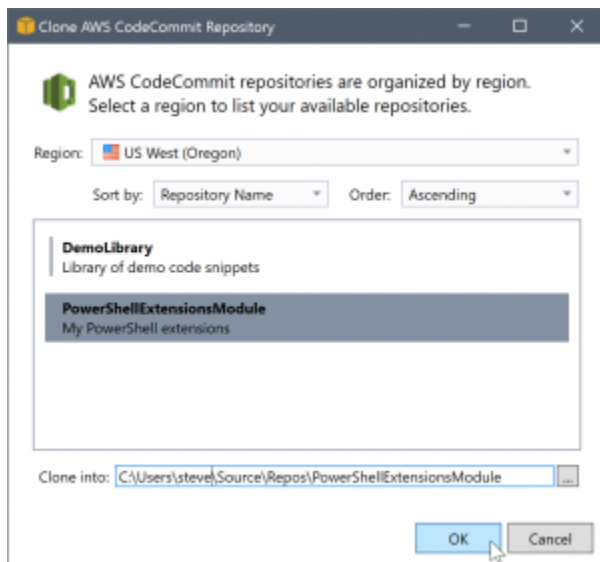


Sekarang kredensi untuk Git tersedia, operasi klon untuk hasil repositori baru (lihat indikasi kemajuan untuk operasi di dalam Team Explorer). Jika Anda memilih untuk memiliki default `.gitignore` file diterapkan, itu berkomitmen untuk repositori dengan komentar dari 'Komite Initial'.

Itu saja yang ada untuk menyiapkan kredensia dan membuat repositori dalam Team Explorer. Setelah kredensi yang diperlukan berada di tempat, semua yang Anda lihat saat membuat repositori baru di masa depan adalah Membuat Baru AWS CodeCommit Repositori kotak dialog itu sendiri.

## Kloning Repositori

Untuk mengkloning repositori yang ada, kembali ke panel koneksi untuk AWS CodeCommit di Team Explorer. Klik Klon link untuk membuka Klon AWS CodeCommit Repositori kotak dialog, dan kemudian pilih repositori untuk mengkloning dan lokasi pada disk di mana Anda ingin menempatkannya.



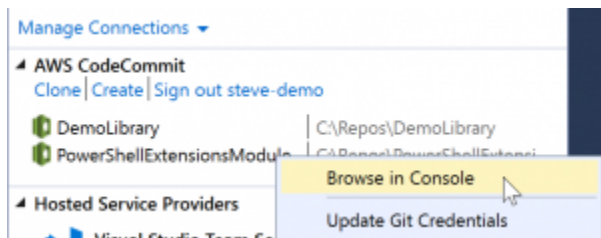
Setelah Anda memilih wilayah, Toolkit for Visual Studio query layanan untuk menemukan repositori yang tersedia di wilayah itu dan menampilkannya di bagian daftar pusat kotak dialog. Nama dan deskripsi opsional dari setiap repositori juga ditampilkan. Anda dapat menyusun ulang daftar untuk mengurutkannya dengan nama repositori atau tanggal modifikasi terakhir, dan untuk mengurutkan masing-masing dalam urutan menaik atau turun.

Setelah Anda memilih repositori Anda dapat memilih lokasi yang akan dikloning. Ini default ke lokasi repositori yang sama yang digunakan di plugin lain untuk Team Explorer, tetapi Anda dapat menelusuri atau memasukkan lokasi lain. Secara default, nama repositori disufiks ke jalur yang dipilih. Namun, jika Anda menginginkan jalur tertentu, cukup edit kotak teks setelah Anda memilih folder. Teks apa pun yang ada di dalam kotak saat Anda mengklik OKE akan menjadi folder tempat Anda menemukan repositori yang dikloning.

Setelah memilih repositori dan lokasi folder, Anda kemudian klik OKE untuk melanjutkan dengan operasi clone. Sama seperti dengan membuat repositori, Anda dapat melihat kemajuan operasi klon yang dilaporkan di Team Explorer.

## Bekerja dengan Repositori

Saat Anda mengkloning atau membuat repositori, perhatikan bahwa repositori lokal untuk koneksi tercantum dalam panel koneksi di Team Explorer di bawah tautan operasi. Entri ini memberi Anda cara mudah untuk mengakses repositori untuk menelusuri konten. Cukup klik kanan repositori dan pilih Menjelajahi Konsol.



Anda juga dapat menggunakan Perbarui Kredensial Git untuk memperbarui kredensi Git yang tersimpan terkait dengan profil kredensial. Ini berguna jika Anda telah memutar kredensialnya. Perintah membuka Kredensi Git untuk AWS CodeCommit kotak dialog di mana Anda dapat memasukkan atau mengimpor kredensi baru.

Operasi Git pada repositori bekerja seperti yang Anda harapkan. Anda dapat melakukan komit lokal dan, saat Anda siap untuk berbagi, Anda menggunakan opsi Sinkronisasi di Team Explorer. Karena kredensi Git sudah disimpan secara lokal dan terkait dengan kami yang terhubung AWS profil credential, kami tidak akan diminta untuk memasok mereka lagi untuk operasi terhadap AWS CodeCommit jarak jauh.

## Menggunakan CodeArtifact di Visual Studio

AWS CodeArtifact adalah layanan repositori artefak terkelola penuh yang memudahkan organisasi untuk menyimpan dan berbagi paket perangkat lunak yang digunakan untuk pengembangan aplikasi dengan aman. Anda dapat menggunakan CodeArtifact dengan alat build populer dan manajer paket seperti NuGet dan NET Core CLI dan Visual Studio. Anda juga dapat mengkonfigurasi CodeArtifact untuk menarik paket dari repositori publik eksternal seperti [Nuget.org](https://www.nuget.org).

Dalam CodeArtifact, paket Anda disimpan dalam repositori yang kemudian disimpan dalam domain. Parameter AWS Toolkit for Visual Studio menyederhanakan konfigurasi Visual Studio dengan repositori CodeArtifact Anda, sehingga mudah untuk mengonsumsi paket di Visual Studio dari kedua CodeArtifact langsung dan Nuget.org.

## Tambahkan repositori CodeArtifact Anda sebagai sumber paket NuGet

Untuk mengonsumsi paket dari CodeArtifact Anda, Anda perlu menambahkan repositori Anda sebagai sumber paket di Manajer Paket NuGet dalam Visual Studio

Untuk menambahkan repositori Anda sebagai sumber paket

1. Masuk ke AWS Explorer, arahkan ke repositori Anda di AWS CodeArtifact simpul.



2. Buka menu konteks (klik kanan) untuk repositori yang ingin ditambahkan, lalu pilih Titik Akhir Sumber NuGet.
3. Arahkan ke Sumber Paket di bawah Manajer Paket NuGet simpul dalam Alat > Ops menu.
4. Masuk Sumber Paket, pilih tanda plus (+), edit nama, dan tempel URL titik akhir sumber NuGet yang Anda salin sebelumnya di Sumber Bidang.
5. Pilih kotak centang di samping sumber paket yang baru ditambahkan untuk mengaktifkannya.

#### Note

Kami merekomendasikan untuk menambahkan koneksi eksternal ke NuGet.org ke CodeArtifact Anda dan menonaktifkan nuget.org sumber paket dalam Visual Studio. Saat menggunakan koneksi eksternal, semua dependensi ditarik dari NuGet.org disimpan dalam CodeArtifact. Jika NuGet.org turun untuk alasan apapun, paket yang Anda butuhkan masih akan tersedia. Untuk informasi selengkapnya tentang koneksi eksternal, lihat [Menambahkan koneksi eksternal](#) di AWS CodeArtifact Panduan Pengguna.

6. Pilih OKE untuk menutup menu.

Untuk informasi lebih lanjut tentang cara menggunakan CodeArtifact dengan Visual Studio, lihat [Menggunakan CodeArtifact dengan Visual Studio](#) di AWS CodeArtifact Panduan Pengguna.

## Amazon RDS dari AWS Penjelajah

Amazon Relational Database Service (Amazon RDS) adalah layanan yang memungkinkan Anda untuk menyediakan dan mengelola sistem basis data relasional SQL di cloud. Amazon RDS mendukung tiga jenis sistem basis data:

- MySQL Community Edition
- Oracle Database Enterprise Edition
- Microsoft SQL Server (Express, Standar, atau Edisi Web)

Untuk informasi selengkapnya, lihat [Panduan Pengguna Amazon RDS](#).

Banyak fungsi yang dibahas di sini juga tersedia melalui [AWS Konsol Manajemen](#) Amazon RDS.

Topik

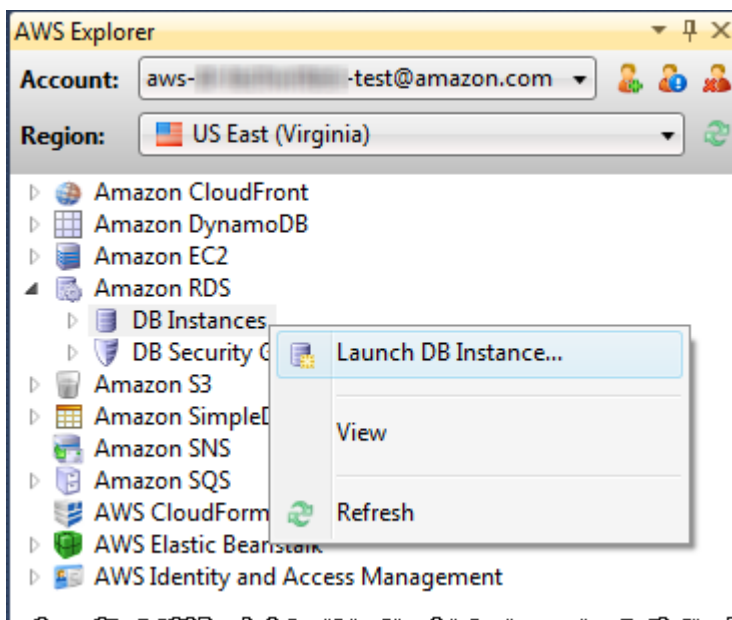
- [Luncurkan Instans Basis Data Amazon RDS](#)
- [Membuat Microsoft SQL Server Database dalam Instans RDS](#)
- [Grup keamanan Amazon RDS](#)

## Luncurkan Instans Basis Data Amazon RDS

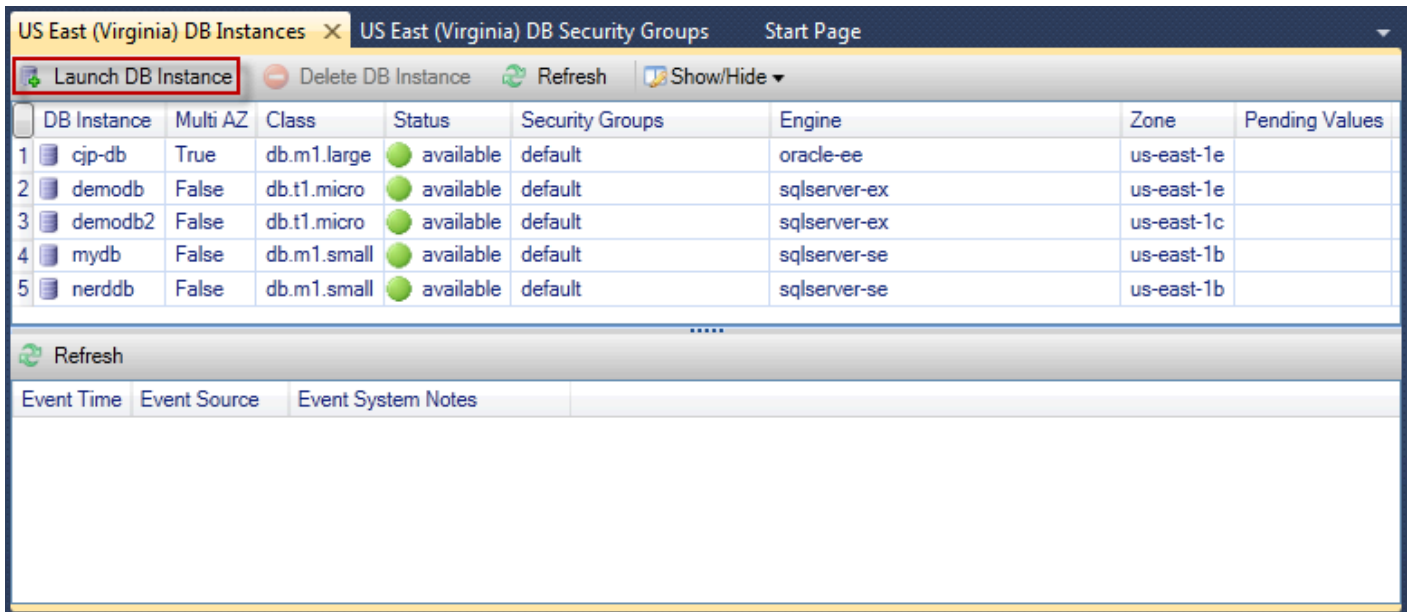
Dengan AWSExplorer, Anda dapat meluncurkan instance dari salah satu mesin database yang didukung oleh Amazon RDS. Walkthrough berikut menunjukkan pengalaman pengguna untuk meluncurkan instance Microsoft SQL Server Standard Edition, tetapi pengalaman pengguna serupa untuk semua mesin yang didukung.

### Meluncurkan instans Amazon RDS

1. Masuk AWSExplorer, buka menu konteks (klik kanan) untuk Amazon RDS node dan pilih Luncurkan Instans DB.

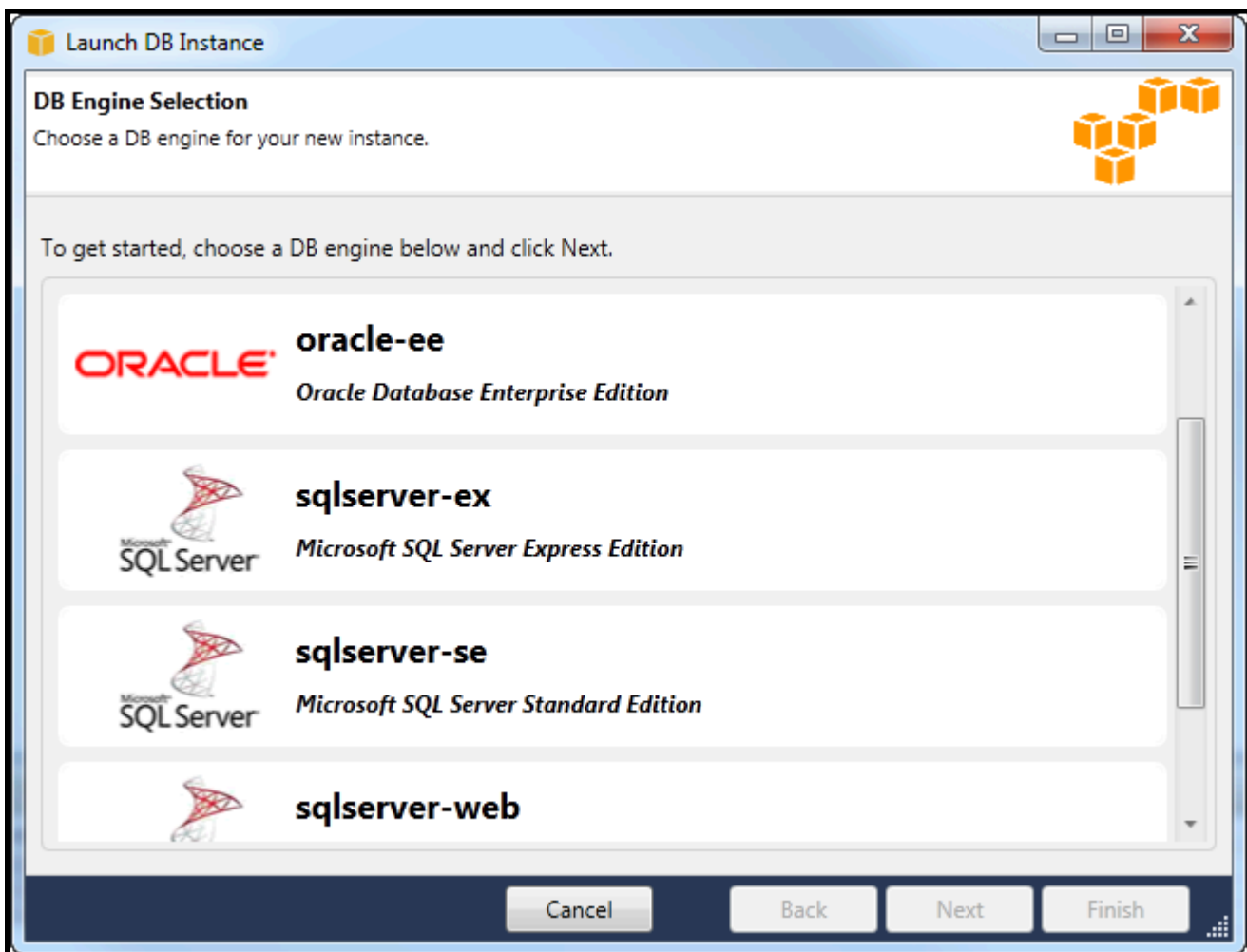


Atau, pada Instans DB tab, pilih Luncurkan Instans DB.



DB Instance	Multi AZ	Class	Status	Security Groups	Engine	Zone	Pending Values
1 cjp-db	True	db.m1.large	available	default	oracle-ee	us-east-1e	
2 demodb	False	db.t1.micro	available	default	sqlserver-ex	us-east-1e	
3 demodb2	False	db.t1.micro	available	default	sqlserver-ex	us-east-1c	
4 mydb	False	db.m1.small	available	default	sqlserver-se	us-east-1b	
5 nerddb	False	db.m1.small	available	default	sqlserver-se	us-east-1b	

2. DiPemilihan Mesin DBkotak dialog, pilih tipe mesin basis data untuk diluncurkan. Untuk panduan ini, pilih Microsoft SQL Server Standard Edition (sqlserver-se), dan kemudian pilihSelanjutnya.



### 3. DiOpsi Instans Mesin DBkotak dialog, pilih opsi konfigurasi.

DiOpsi dan Kelas Instans Mesin DBbagian, Anda dapat menentukan pengaturan berikut.

#### Model lisensi

Tipe mesin	Lisensi
Microsoft SQL Server	lisensi-termasuk
MySql	umum-publik-lisensi
Oracle	Bring-your-License

Model lisensi bervariasi, tergantung pada jenis mesin basis data. Lisensi tipe mesin Microsoft SQL Server lisensi-termasuk MySql general-public-license Oracle membawa-lisensi Anda sendiri

#### Versi Instans DB

Pilih versi mesin basis data yang ingin Anda gunakan. Jika hanya satu versi yang didukung, versi dipilih untuk Anda.

#### Kelas Instans DB

Memilih kelas instans untuk mesin basis data. Harga untuk kelas misalnya bervariasi. Untuk informasi selengkapnya, lihat [Harga Amazon RDS](#).

#### Lakukan penyebaran multi AZ

Pilih opsi ini untuk membuat penyebaran Multi-AZ untuk meningkatkan daya tahan dan ketersediaan data. Amazon RDS menyediakan dan menyimpan salinan standby database Anda di Availability Zone yang berbeda untuk kegagalan otomatis jika terjadi pemadaman terjadwal atau tidak direncanakan. Untuk informasi tentang penetapan harga untuk penerapan Multi-AZ, lihat bagian harga pada [Amazon RDS](#) Halaman detail. Opsi ini tidak didukung untuk Microsoft SQL Server.

#### Tingkatkan versi minor secara otomatis

Pilih opsi ini untuk memiliki AWS secara otomatis melakukan pembaruan versi minor pada instans RDS Anda untuk Anda.

Dilinstans basis data RDSbagian, Anda dapat menentukan pengaturan berikut.

Penyimpanan yang dialokasikan

Mesin	Minimum (GB)	Maksimum (GB)
MySQL	5	1024
Oracle Enterprise Edition	10	1024
Microsoft SQL Server Express Edition	30	1024
Microsoft SQL Server Standard Edition	250	1024
Edisi Web Microsoft SQL	30	1024

Minimal dan maksimum untuk penyimpanan yang dialokasikan tergantung pada jenis mesin database. Mesin Minimum (GB) Maksimum (GB) MySQL 5 1024 Oracle Enterprise Edition 10 1024 Microsoft SQL Server Express Edition 30 1024 Microsoft SQL Server Edisi Standar 250 1024 Microsoft SQL Server Edisi Web 30 1024

Pengidentifikasi instans DB

Menentukan nama untuk instans basis data. Nama ini tidak peka huruf besar/kecil. Ini akan ditampilkan dalam bentuk huruf kecil diAWSPenjelajah.

Nama Pengguna Utama

Ketik nama untuk administrator contoh database.

Kata sandi pengguna utama

Ketik sandi untuk administrator instans basis data.

Konfirmasi Kata Sandi

Ketik kata sandi lagi untuk memverifikasi itu benar.

1. DiOpsikan kotak dialog, Anda dapat menentukan pengaturan berikut.

#### Port Basis Data

Ini adalah port TCP yang akan digunakan instance untuk berkomunikasi di jaringan. Jika komputer Anda mengakses Internet melalui firewall, tetapkan nilai ini ke port tempat firewall Anda mengizinkan lalu lintas.

#### Availability Zone

Gunakan opsi ini jika Anda ingin instance diluncurkan di Availability Zone tertentu di wilayah Anda. Contoh basis data yang Anda tentukan mungkin tidak tersedia di semua Availability Zone di wilayah tertentu.

#### Grup keamanan RDS

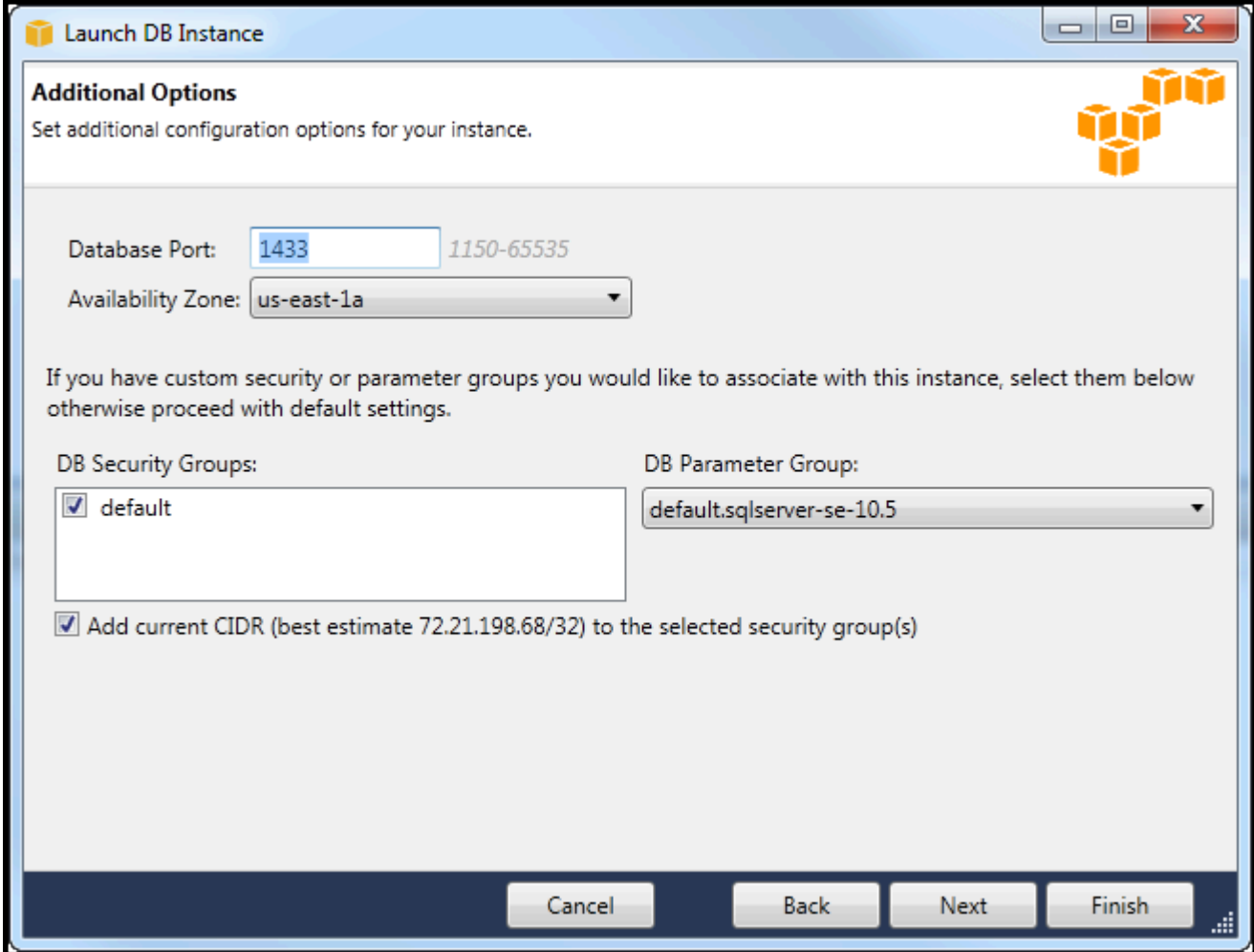
Pilih grup keamanan RDS (atau grup) untuk diasosiasikan dengan instans Anda. Grup keamanan RDS menentukan alamat IP, instans Amazon EC2, dan Akun AWS yang diizinkan

untuk mengakses instans Anda. Untuk informasi selengkapnya tentang grup keamanan RDS, lihat [Grup Keamanan Amazon RDS](#). Toolkit for Visual Studio mencoba menentukan alamat IP Anda saat ini dan menyediakan opsi untuk menambahkan alamat ini ke grup keamanan yang terkait dengan instans Anda. Namun, jika komputer Anda mengakses Internet melalui firewall, alamat IP yang dihasilkan Toolkit untuk komputer Anda mungkin tidak akurat. Untuk menentukan alamat IP mana yang akan digunakan, konsultasikan administrator sistem Anda.

### Grup Parameter DB

(Opsional) Dari daftar drop-down ini, pilih grup parameter DB untuk diasosiasikan dengan instans Anda. DB kelompok parameter memungkinkan Anda untuk mengubah konfigurasi default untuk instance. Untuk informasi selengkapnya, kunjungi [Panduan Pengguna Amazon Relational Database Service](#) dan [Artikel ini](#).

Bila Anda telah menetapkan pengaturan pada kotak dialog ini, pilih **Selanjutnya**.



**Launch DB Instance**

**Additional Options**  
Set additional configuration options for your instance.

Database Port:  1150-65535

Availability Zone:

If you have custom security or parameter groups you would like to associate with this instance, select them below otherwise proceed with default settings.

DB Security Groups:

- default

DB Parameter Group:

Add current CIDR (best estimate 72.21.198.68/32) to the selected security group(s)

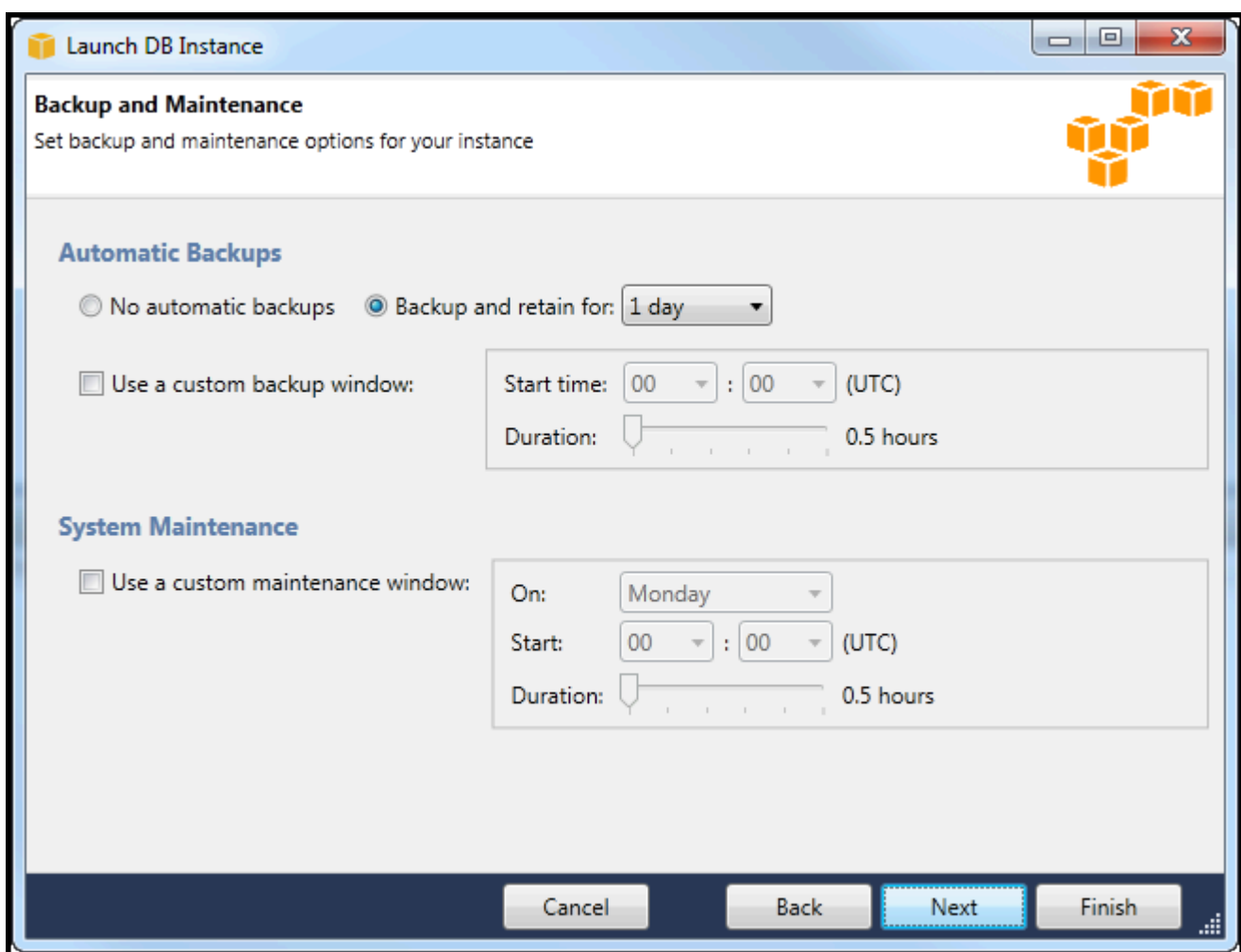
Cancel Back Next Finish

- ParameterBackup dan Pemeliharaankotak dialog memungkinkan Anda untuk menentukan apakah Amazon RDS harus mencadangkan instans Anda dan jika demikian, untuk berapa lama cadangan harus dipertahankan. Anda juga dapat menentukan jendela waktu selama backup harus terjadi.

Kotak dialog ini juga memungkinkan Anda untuk menentukan apakah Anda ingin Amazon RDS melakukan pemeliharaan sistem pada instans Anda. Pemeliharaan mencakup patch rutin dan upgrade versi minor.

Jendela waktu yang Anda tentukan untuk pemeliharaan sistem tidak dapat tumpang tindih dengan jendela yang ditentukan untuk backup.

Pilih Selanjutnya.



- Kotak dialog terakhir di wizard memungkinkan Anda untuk meninjau pengaturan untuk instance Anda. Jika Anda perlu memodifikasi pengaturan, gunakanKembalitombol. Jika semua pengaturan sudah benar, pilihLuncurkan.



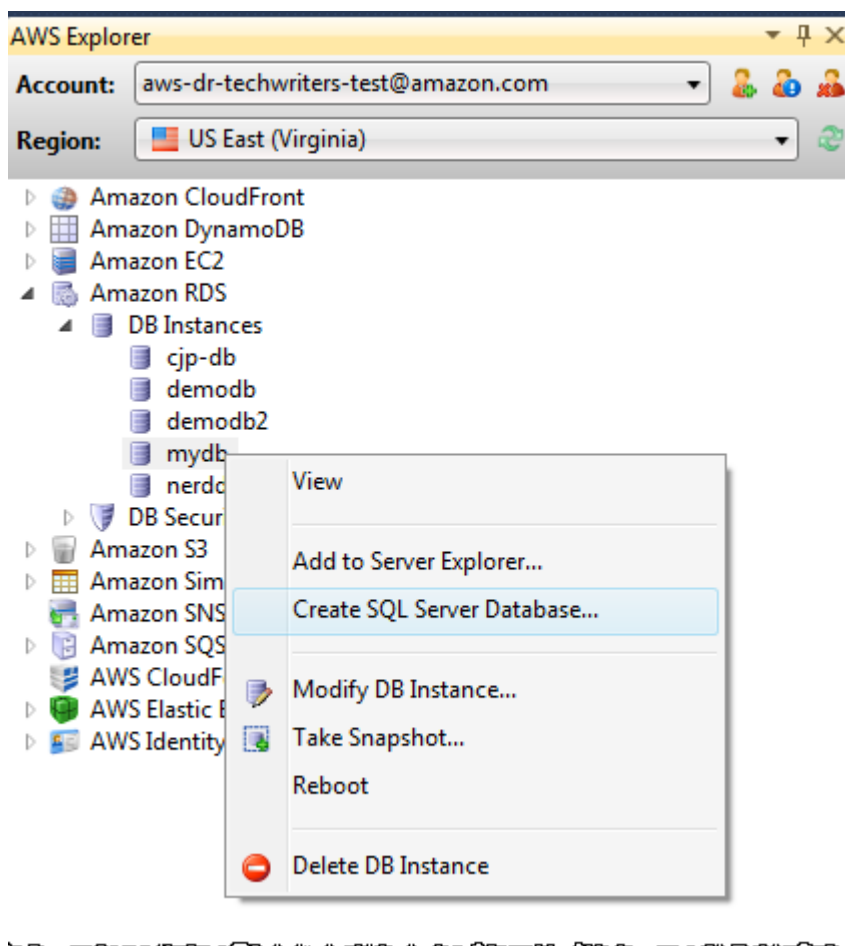
## Membuat Microsoft SQL Server Database dalam Instans RDS

Microsoft SQL Server dirancang sedemikian rupa sehingga, setelah meluncurkan instans Amazon RDS, Anda perlu membuat database SQL Server di instans RDS.

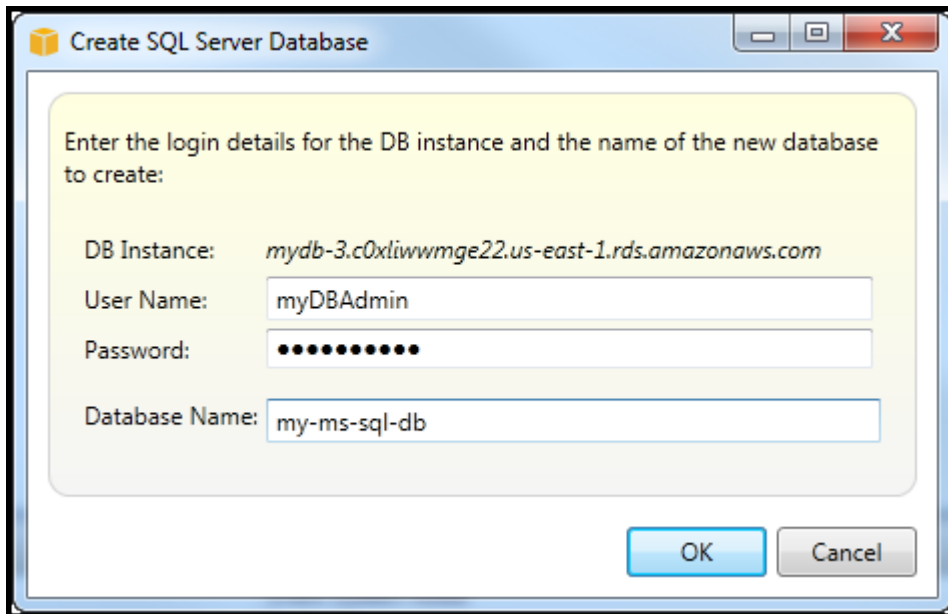
Untuk informasi tentang cara membuat instans Amazon RDS, lihat [Meluncurkan Instans Basis Data Amazon RDS](#).

Untuk membuat basis data Microsoft SQL Server

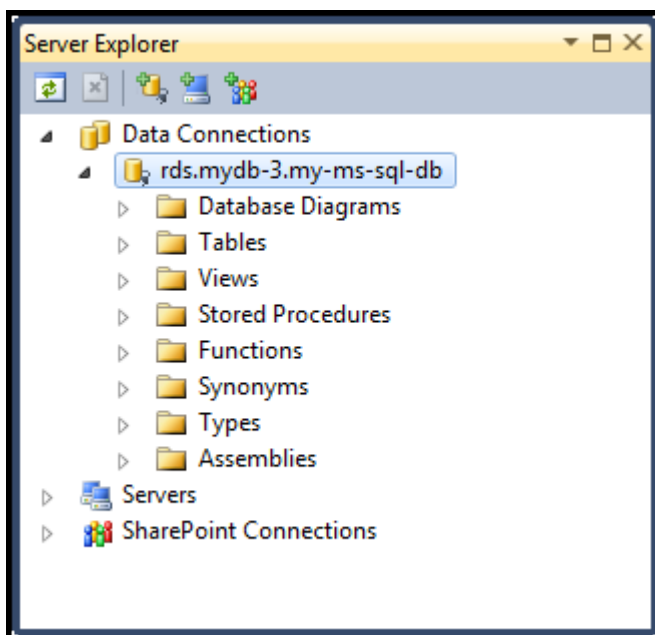
1. Masuk ke AWS Explorer, buka menu konteks (klik kanan) untuk node yang sesuai dengan instans RDS Anda untuk Microsoft SQL Server, dan pilih **Membuat Basis Data SQL Server**.



2. Di **Membuat Basis Data SQL Server** kotak dialog, ketik kata sandi yang Anda tentukan saat Anda membuat contoh RDS, ketik nama untuk database Microsoft SQL Server, dan kemudian pilih **OKE**.



3. Toolkit for Visual Studio menciptakan database Microsoft SQL Server dan menambahkannya ke Visual Studio Server Explorer.



## Grup keamanan Amazon RDS

Grup keamanan Amazon RDS memungkinkan Anda mengelola akses jaringan ke instans Amazon RDS Anda. Dengan grup keamanan, Anda menetapkan set alamat IP menggunakan notasi CIDR, dan hanya lalu lintas jaringan yang berasal dari alamat ini yang diakui oleh instans Amazon RDS Anda.

Meskipun berfungsi dengan cara yang sama, grup keamanan Amazon RDS berbeda dari grup keamanan Amazon EC2. Dimungkinkan untuk menambahkan grup keamanan EC2 ke grup keamanan RDS Anda. Instans EC2 yang merupakan anggota grup keamanan EC2 kemudian dapat mengakses instans RDS yang merupakan anggota grup keamanan RDS.

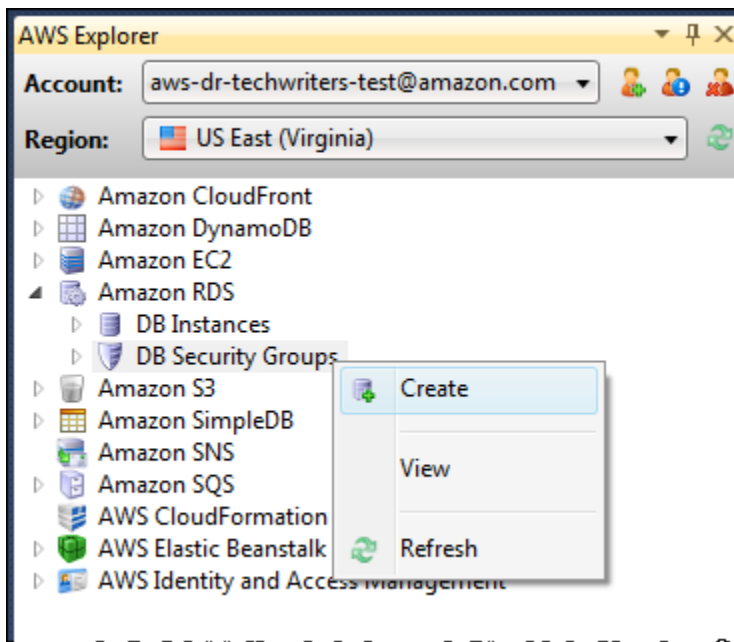
Untuk informasi selengkapnya tentang grup keamanan Amazon RDS, buka [Grup keamanan RDS](#). Untuk informasi lebih lanjut tentang grup keamanan Amazon EC2, buka [Panduan Pengguna EC2](#).

## Buat Grup Keamanan Amazon RDS

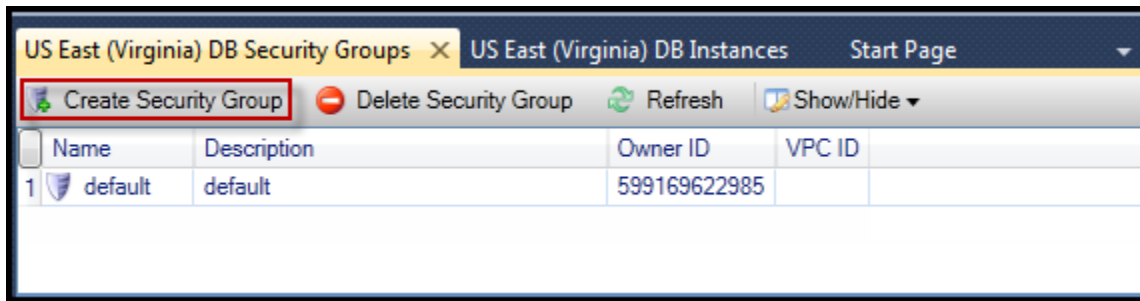
Anda dapat menggunakan Toolkit for Visual Studio untuk membuat grup keamanan RDS. Jika Anda menggunakan AWSToolkit untuk meluncurkan instance RDS, wizard akan memungkinkan Anda untuk menentukan grup keamanan RDS untuk digunakan dengan instance Anda. Anda dapat menggunakan prosedur berikut untuk membuat grup keamanan tersebut sebelum memulai wizard.

Untuk membuat grup keamanan Amazon RDS

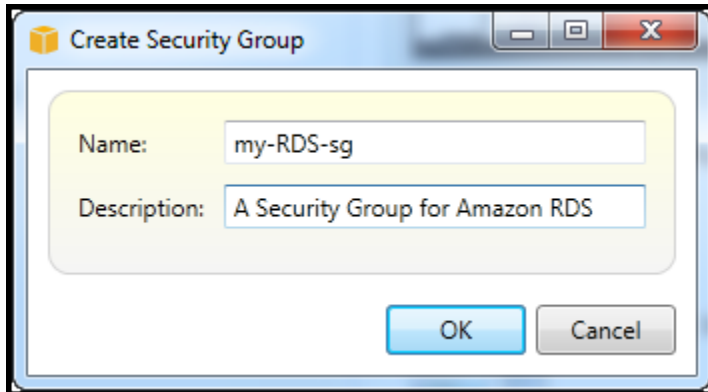
1. Masuk AWSExplorer, memperluas Amazon RDS node, buka menu konteks (klik kanan) untuk Grup keamanan DB subnode dan pilih Buat.



Atau, pada Kelompok Keamanan tab, pilih Buat Grup Keamanan. Jika tab ini tidak ditampilkan, buka menu konteks (klik kanan) untuk Grup keamanan DB subnode dan pilih Lihat.



2. DiBuat Grup Keamanankotak dialog, ketik nama dan deskripsi untuk grup keamanan, lalu pilihOKE.



## Mengatur Izin Akses untuk Grup Keamanan Amazon RDS

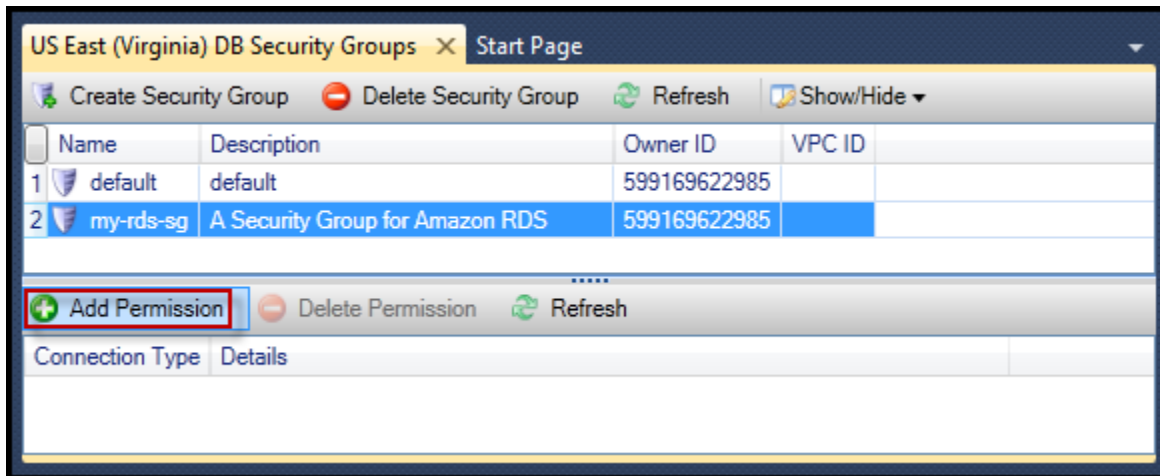
Secara default, grup keamanan Amazon RDS baru tidak menyediakan akses jaringan. Untuk mengaktifkan akses ke instans Amazon RDS yang menggunakan grup keamanan, gunakan prosedur berikut untuk mengatur izin aksesnya.

### Menetapkan akses untuk grup keamanan Amazon RDS

1. PadaKelompok Keamanantab, pilih grup keamanan dari tampilan daftar. Jika grup keamanan Anda tidak muncul dalam daftar, pilihRefresh. Jika grup keamanan Anda masih tidak muncul dalam daftar, pastikan Anda melihat daftar yang benarAWSwilayah. Grup keamanantab diAWSToolkit adalah wilayah khusus.

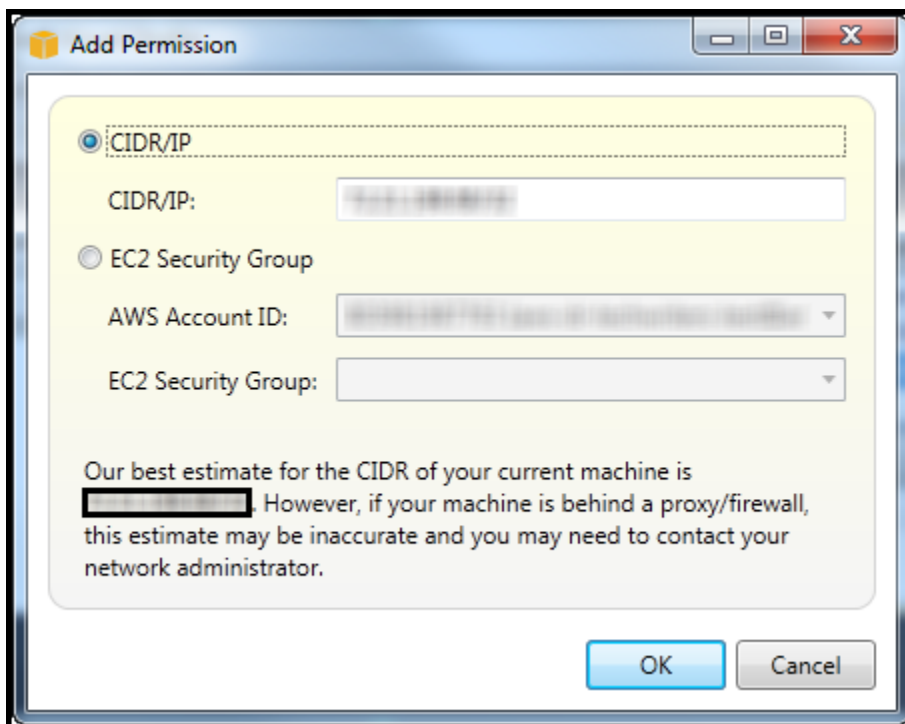
Jika tidakGrup keamanantab muncul, diAWSExplorer, buka menu konteks (klik kanan) untukGrup keamanan DBsubnode dan pilihLihat.

2. PilihTambah Izin.



Tambah Izintombol padaKelompok Keamanan tab

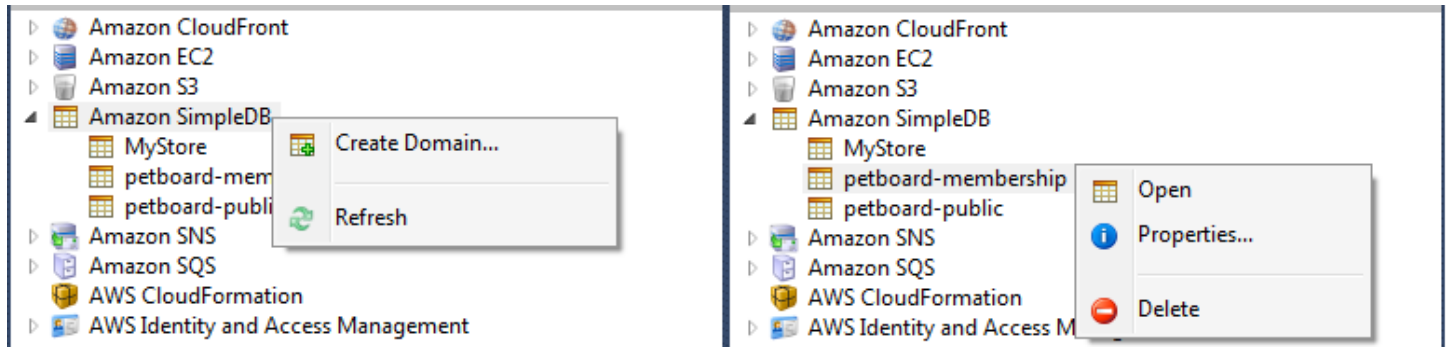
3. DiTambah Izinkotak dialog, Anda dapat menggunakan notasi CIDR untuk menentukan alamat IP mana yang dapat mengakses instans RDS Anda, atau Anda dapat menentukan grup keamanan EC2 mana yang dapat mengakses instans RDS Anda. Bila Anda memilihGrup keamanan EC2, Anda dapat menentukan akses untuk semua instans EC2 yang terkait denganAkun AWSmemiliki akses, atau Anda dapat memilih grup keamanan EC2 dari daftar tarik-turun.



ParameterAWSToolkit mencoba menentukan alamat IP Anda dan mengisi kotak dialog secara otomatis dengan spesifikasi CIDR yang sesuai. Namun, jika komputer Anda mengakses Internet melalui firewall, CIDR yang ditentukan oleh Toolkit mungkin tidak akurat.

## Menggunakan Amazon SimpleDB dari AWS Penjelajah

AWSExplorer menampilkan semua domain Amazon SimpleDB yang terkait dengan yang aktif AWS akun. From AWSExplorer, Anda dapat membuat atau menghapus domain Amazon SimpleDB.



Create, delete, or open Amazon SimpleDB domains associated with your account

### Mengeksekusi Query dan Mengedit Hasil

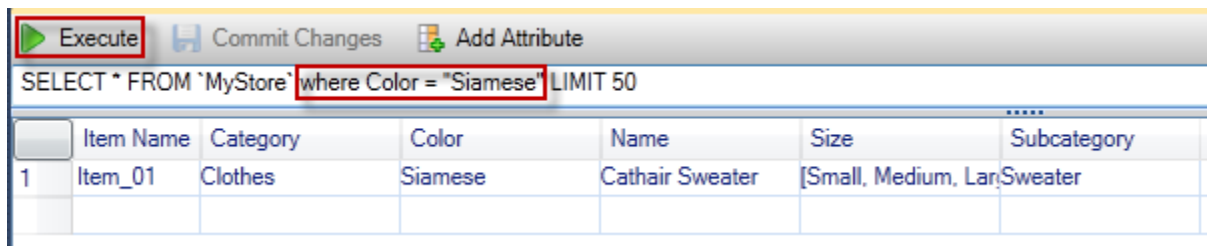
AWSExplorer juga dapat menampilkan tampilan grid domain Amazon SimpleDB tempat Anda dapat melihat item, atribut, dan nilai dalam domain tersebut. Anda dapat mengeksekusi query sehingga hanya subset dari item domain yang ditampilkan. Dengan mengklik dua kali sel, Anda dapat mengedit nilai untuk atribut yang sesuai item tersebut. Anda juga dapat menambahkan atribut baru ke domain.

Domain yang ditampilkan di sini berasal dari sampel Amazon SimpleDB yang disertakan dengan AWS SDK for .NET.

Item Name	Category	Color	Make	Model	Name	Size	Subcategory	Year
Item_01	Clothes	Siamese			Cathair Sweater	[Small, Medium, Lar	Sweater	
Item_02	Clothes	Paisley Acid Wash			Designer Jeans	[32x32, 30x32, 32x3	Pants	
Item_03	Clothes	[Yellow, Pink]			Sweatpants	Medium	Pants	
Item_04	Car Parts		Audi	S4	Turbos		Engine	[2002, 2001, 2000]
Item_05	Car Parts		Audi	S4	O2 Sensor		Emissions	[2001, 2000, 2002]

### Amazon SimpleDB grid view

Untuk menjalankan kueri, edit kueri di kotak teks di bagian atas tampilan grid, dan kemudian pilih Eksekusi. Tampilan disaring untuk hanya menampilkan item yang cocok dengan kueri.

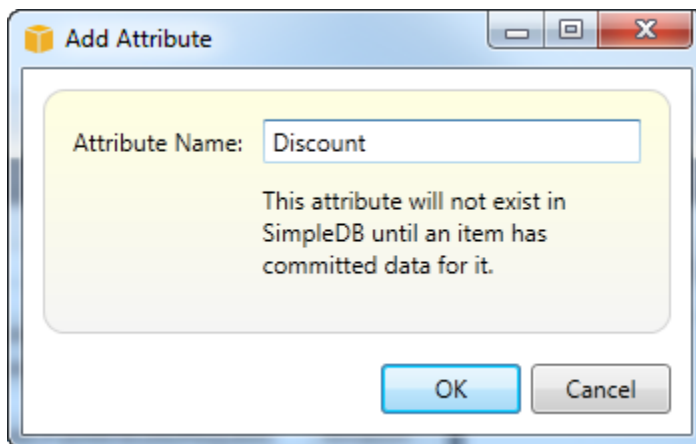


Execute query from AWS Explorer

Untuk mengedit nilai yang terkait dengan atribut, klik dua kali sel yang sesuai, edit nilainya, lalu pilih **Komit Perubahan**.

Menambahkan Atribut

Untuk menambahkan atribut, di bagian atas tampilan, pilih **Menambahkan atribut**.



Menambahkan atribut dialog box

Untuk membuat atribut bagian dari domain, Anda harus menambahkan nilai untuk itu ke setidaknya satu item dan kemudian memilih **Komit Perubahan**.



Commit changes for a new attribute

Pemberian Nomor Halaman Hasil Kueri

Ada tiga tombol di bagian bawah tampilan.



### Paginate and export buttons

Dua tombol pertama memberikan pagination untuk hasil query. Untuk menampilkan halaman hasil tambahan, pilih tombol pertama. Untuk menampilkan sepuluh halaman hasil tambahan, pilih tombol kedua. Dalam konteks ini, halaman sama dengan 100 baris atau jumlah hasil yang ditentukan oleh nilai LIMIT, jika disertakan dalam query.

### Ekspor ke CSV

Tombol terakhir mengekspor hasil saat ini ke file CSV.

## Menggunakan Amazon SQSAWSPenjelajah

Amazon Simple Queue Service (Amazon SQS) adalah layanan antrian fleksibel yang memungkinkan pesan lewat antara proses eksekusi yang berbeda dalam aplikasi perangkat lunak. Antrian Amazon SQS terletak diAWSinfrastruktur, tetapi proses yang lewat pesan dapat ditempatkan secara lokal, pada instans Amazon EC2, atau pada beberapa kombinasi dari ini. Amazon SQS sangat ideal untuk mengkoordinasikan distribusi pekerjaan di beberapa komputer.

Toolkit for Visual Studio memungkinkan Anda untuk melihat antrian Amazon SQS yang terkait dengan akun aktif, membuat dan menghapus antrian, dan mengirim pesan melalui antrian. (Dengan akun aktif, kami berarti akun yang dipilih diAWSExplorer.)

Untuk informasi selengkapnya tentang Amazon SQS, kunjungi[Pengantar SQS](#)diAWSdokumentasi.

## Membuat Antrean

Anda dapat membuat antrean Amazon SQSAWSExplorer. ARN dan URL untuk antrian akan didasarkan pada nomor akun untuk akun aktif dan nama antrian yang Anda tentukan saat pembuatan.

### Untuk membuat antrean

1. MasukAWSExplorer, buka menu konteks (klik kanan) untukAmazon SQSnode, dan kemudian pilihMembuat Antrean.
2. DiMembuat Antreankotak dialog, tentukan nama antrian, batas waktu visibilitas default, dan penundaan pengiriman default. Batas waktu visibilitas default dan penundaan pengiriman default



ditentukan dalam hitungan detik. Batas waktu visibilitas default adalah jumlah waktu bahwa pesan tidak terlihat oleh proses penerimaan potensial setelah proses tertentu telah memperoleh pesan. Keterlambatan pengiriman default adalah jumlah waktu dari saat pesan dikirim ke saat pertama kali terlihat oleh proses penerimaan potensial.

3. Pilih OKE. Antrian baru akan muncul sebagai subnode di bawah Amazon SQS simpul.

## Menghapus Antrean

Anda dapat menghapus antrian yang ada dari AWSExplorer. Jika Anda menghapus antrian, pesan yang terkait dengan antrian tidak lagi tersedia.

Untuk menghapus antrean

1. Masuk AWSExplorer, buka menu konteks (klik kanan) untuk antrean yang ingin Anda hapus, lalu pilih Hapus.

## Mengelola Properti Antrean

Anda dapat melihat dan mengedit properti untuk salah satu antrian yang ditampilkan dalam AWSExplorer. Anda juga dapat mengirim pesan ke antrean dari tampilan properti ini.

Mengelola properti antrian

- Masuk AWSExplorer, buka menu konteks (klik kanan) untuk antrean yang propertinya ingin Anda kelola, lalu pilih Antrean Lihat.

Dari tampilan properti antrian, Anda dapat mengedit batas waktu visibilitas, ukuran pesan maksimum, periode retensi pesan, dan penundaan pengiriman default. Penundaan pengiriman default dapat diganti saat Anda mengirim pesan. Pada screenshot berikut, teks dikaburkan adalah komponen nomor akun dari ARN antrian dan URL.

Save Send Refresh

Visibility timeout (Seconds): 30 Created timestamp: 10/20/2011 1:34:49 PM

Maximum message size (Bytes): 65536 Last modified timestamp: 10/20/2011 1:34:49 PM

Message retention period (Seconds): 345600 Number of messages: 0


Default Delivery Delay (Seconds): 120 Number of messages not visible: 0

Queue ARN: arn:aws:sqs:us-east-1: :my-tk-queue

Queue URL: https://queue.amazonaws.com/ /my-tk-queue

**Message Sampling**

Message Id	Message Body	Sender Id	Sent

 Changes can take up to 60 seconds to propagate throughout the SQS system.

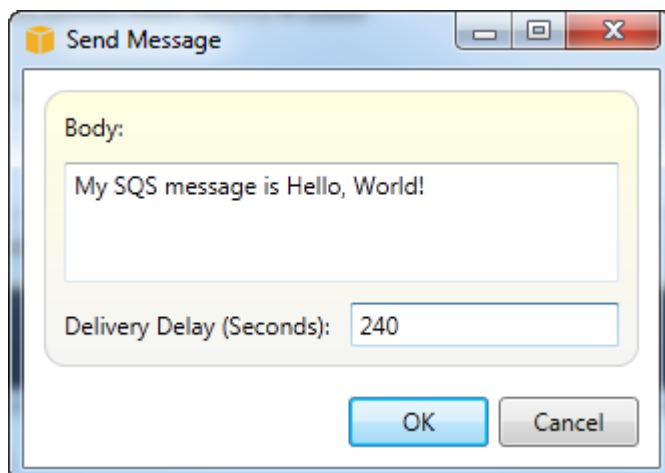
SQS queue properties view

## Mengirim Pesan ke Antrian

Dari tampilan properti antrian, Anda dapat mengirim pesan ke antrian.

Cara mengirim pesan

1. Tampilan antrian di bagian atas properti, pilih tombol.
2. Ketik pesan. (Opsional) Masukkan penundaan pengiriman yang akan menimpa penundaan pengiriman default untuk antrian. Pada contoh berikut, kami telah mengganti delay dengan nilai 240 detik. Pilih OKE.



Mengirim Pesan dialog box

3. Tunggu sekitar 240 detik (empat menit). Pesan akan muncul di Pesan pengambilan sampel bagian dari tampilan properti antrian.

Save Send Refresh

Visibility timeout (Seconds):  Created timestamp: 10/20/2011 1:34:49 PM

Maximum message size (Bytes):  Last modified timestamp: 10/20/2011 1:34:49 PM

Message retention period (Seconds):  Number of messages: 1

Default Delivery Delay (Seconds):  Number of messages not visible: 0

Queue ARN: arn:aws:sqs:us-east-1:.....:my-tk-queue

Queue URL: https://queue.amazonaws.com/...../my-tk-queue

**Message Sampling**

Message Id	Message Body	Sender Id	Sent
d58475df-2f92-49ec-a400-957bafcc5daf	My SQS message is Hello, World!	.....	10/20/2011 2:33:02 PM

⚠ Changes can take up to 60 seconds to propagate throughout the SQS system.

### SQS properties view with sent message

Cap waktu dalam tampilan properti antrian adalah saat Anda memilih mengirim tombol. Ini tidak termasuk penundaan. Oleh karena itu, waktu pesan muncul dalam antrian dan tersedia untuk penerima mungkin lebih lambat dari stempel waktu ini. Cap waktu ditampilkan di waktu lokal komputer Anda.

## Manajemen Identitas dan Akses

AWS Identity and Access Management (IAM) memungkinkan Anda mengelola akses ke Akun AWS dan sumber daya. Dengan IAM, Anda dapat membuat beberapa pengguna di primer Anda (akar) Akun AWS. Pengguna ini dapat memiliki kredensialnya sendiri: kata sandi, ID kunci akses, dan kunci rahasia, tetapi semua pengguna IAM berbagi satu nomor akun.

Anda dapat mengelola tingkat akses sumber daya setiap pengguna IAM dengan melampirkan kebijakan IAM ke pengguna. Misalnya, Anda dapat melampirkan kebijakan ke pengguna IAM yang memberikan akses pengguna ke layanan Amazon S3 dan sumber daya terkait di akun Anda, tetapi yang tidak menyediakan akses ke layanan atau sumber daya lainnya.

Untuk manajemen akses yang lebih efisien, Anda dapat membuat grup IAM, yang merupakan koleksi pengguna. Saat Anda melampirkan kebijakan ke grup, kebijakan tersebut akan memengaruhi semua pengguna yang merupakan anggota grup tersebut.

Selain mengelola izin di tingkat pengguna dan grup, IAM juga mendukung konsep IAM role. Seperti pengguna dan grup, Anda dapat melampirkan kebijakan ke peran IAM. Anda kemudian dapat mengaitkan IAM role dengan instans Amazon EC2. Aplikasi yang dijalankan di instans EC2 dapat diakses AWS menggunakan izin yang disediakan oleh peran IAM. Untuk informasi selengkapnya tentang cara menggunakan IAM role dengan Toolkit, lihat [Buat IAM Role..](#) Untuk informasi selengkapnya tentang IAM, kunjungi [Panduan Pengguna IAM.](#)

## Membuat dan Mengonfigurasi Pengguna IAM

Pengguna IAM memungkinkan Anda untuk memberikan orang lain akses ke Akun AWS. Karena Anda dapat melampirkan kebijakan ke pengguna IAM, Anda dapat membatasi sumber daya yang dapat diakses pengguna IAM dan operasi yang dapat mereka lakukan pada sumber daya tersebut.

Sebagai praktik terbaik, semua pengguna yang mengakses Akun AWS harus melakukannya sebagai pengguna IAM—bahkan pemilik akun. Hal ini memastikan bahwa jika kredensi untuk salah satu pengguna IAM dikompromikan, hanya kredensi tersebut dapat dinonaktifkan. Tidak perlu menonaktifkan atau mengubah kredensi root untuk akun tersebut.

Dari Toolkit for Visual Studio, Anda dapat menetapkan izin untuk pengguna IAM baik dengan melampirkan kebijakan IAM ke pengguna atau dengan menetapkan pengguna ke grup. Pengguna IAM yang ditugaskan ke grup memperoleh izin mereka dari kebijakan yang dilampirkan ke grup. Untuk informasi selengkapnya, lihat [Buat Grup IAM](#) dan [Menambahkan Pengguna IAM ke Grup IAM.](#)

Dari Toolkit for Visual Studio, Anda juga dapat menghasilkan AWS kredensial (access key ID dan secret key) untuk pengguna IAM. Untuk informasi selengkapnya, lihat [Menghasilkan Kredensial untuk Pengguna IAM](#)

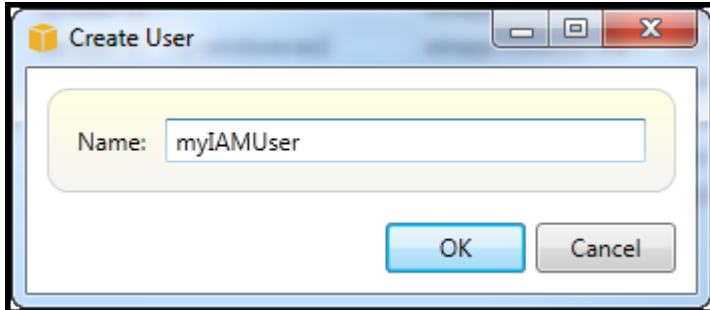


Toolkit for Visual Studio mendukung menentukan kredensial pengguna IAM untuk mengakses layanan melalui AWSExplorer. Karena pengguna IAM biasanya tidak memiliki akses penuh ke semua Amazon Web Services, beberapa fungsi di AWSExplorer mungkin tidak tersedia. Jika Anda menggunakan AWSExplorer untuk mengubah sumber daya sementara akun aktif adalah pengguna IAM dan kemudian beralih akun aktif ke akun root, perubahan mungkin tidak terlihat sampai Anda menyegarkan tampilan di AWSExplorer. Untuk menyegarkan tampilan, pilih tombol refresh ( ).

Untuk informasi tentang cara mengonfigurasi pengguna IAM dari AWS Management Console, pergi ke [Bekerja dengan Pengguna dan Grup](#) dalam Panduan Pengguna IAM.

Untuk membuat pengguna IAM

1. MasukAWSExplorer, memperluasAWS Identity and Access Managementnode, buka menu konteks (klik kanan) untukPenggunalalu pilihMembuat pengguna.
2. DiMembuat penggunakotak dialog, ketik nama untuk pengguna IAM dan pilihOKE. Ini adalah IAM[nama ramah](#). Untuk informasi tentang kendala pada nama untuk pengguna IAM, buka[Panduan Pengguna IAM](#).



Create an IAM user

Pengguna baru akan muncul sebagai subnode di bawahPenggundi bawahAWS Identity and Access Managementsimpul.

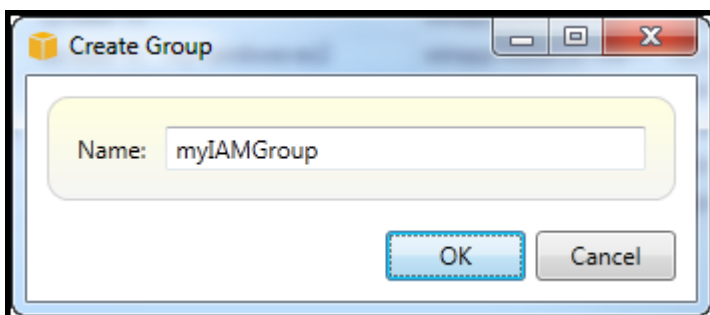
Untuk informasi tentang cara membuat kebijakan dan melampirkannya ke pengguna, lihat[Membuat Kebijakan IAM](#).

## Buat Grup IAM

Grup menyediakan cara menerapkan kebijakan IAM ke koleksi pengguna. Untuk informasi tentang cara mengelola pengguna dan grup IAM, buka[Bekerja dengan Pengguna dan Grup](#)dalam Panduan Pengguna IAM.

Untuk menciptakan sebuah grup IAM

1. MasukAWSExplorer, di bawahManajemen Identitas dan Akses, buka menu konteks (klik kanan) untukGrupdan pilihlahBuat Grup.
2. DiBuat Grupkotak dialog, ketik nama grup IAM dan pilihOKE.



## Create IAM group

Grup IAM baru akan muncul di bawah Grupsubnode dari Manajemen Identitas dan Akses.

Untuk informasi tentang membuat kebijakan dan melampirkannya ke grup IAM, lihat [Membuat Kebijakan IAM](#).

## Menambahkan Pengguna IAM ke Grup IAM

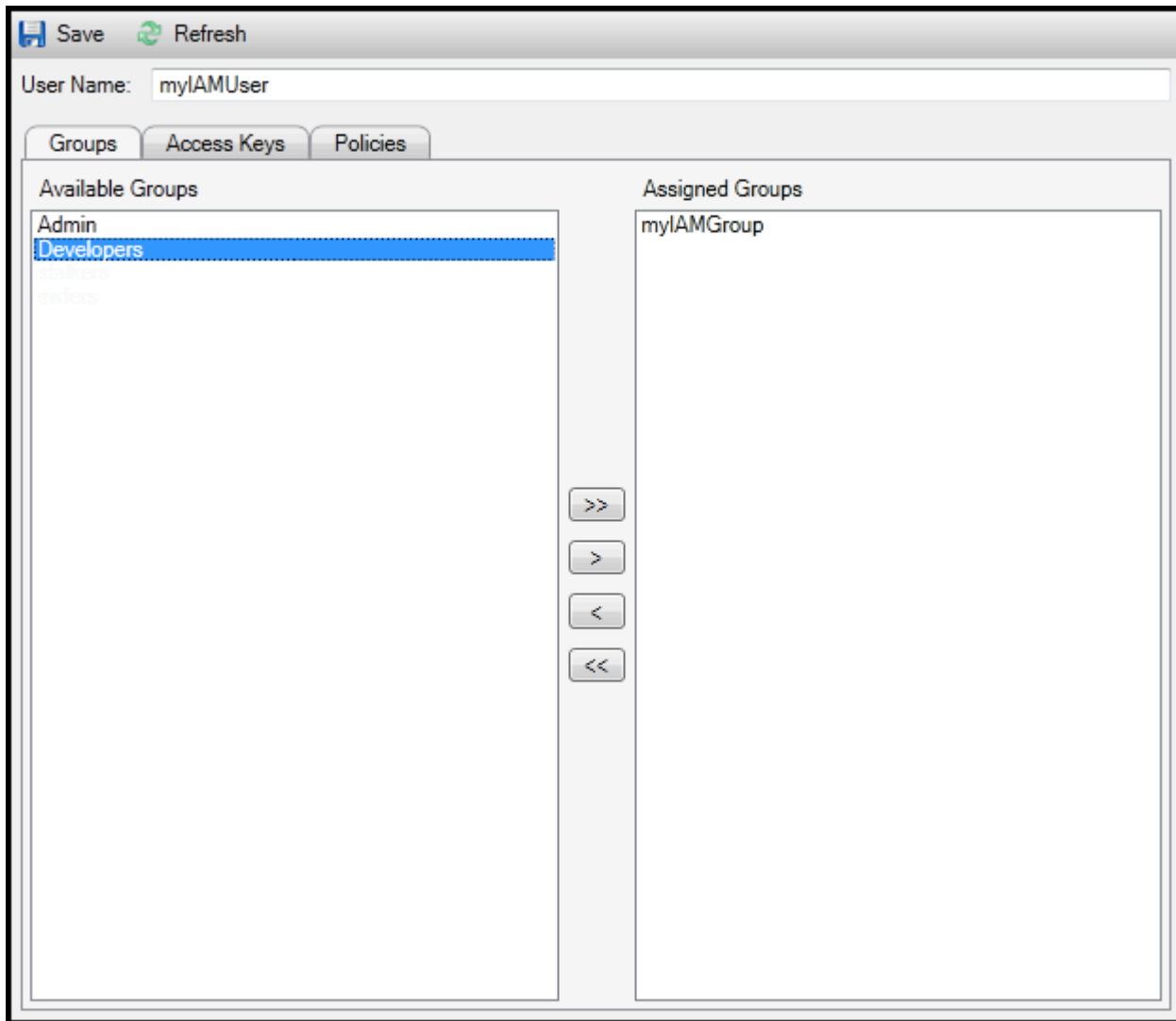
Pengguna IAM yang merupakan anggota grup IAM memperoleh izin akses dari kebijakan yang dilampirkan ke grup. Tujuan dari grup IAM adalah untuk membuatnya lebih mudah untuk mengelola izin di seluruh koleksi pengguna IAM.

Untuk informasi tentang bagaimana kebijakan yang dilampirkan ke grup IAM berinteraksi dengan kebijakan yang melekat pada pengguna IAM yang merupakan anggota grup IAM tersebut, kunjungi [Mengelola Kebijakan IAM di Panduan Pengguna IAM](#).

Masuk AWSExplorer, Anda menambahkan pengguna IAM ke grup IAM dari Penggunasubnode, bukan Grupsubnode.

Untuk menambahkan pengguna IAM ke grup IAM

1. Masuk AWSExplorer, di bawah Manajemen Identitas dan Akses, buka menu konteks (klik kanan) untuk Penggunadan pilihlah Mengedit.



Assign an IAM user to a IAM group

2. Panel kiriGruptab menampilkan grup IAM yang tersedia. Panel kanan menampilkan kelompok yang pengguna IAM yang ditentukan sudah menjadi anggota.

Untuk menambahkan pengguna IAM ke grup, di panel kiri, pilih grup IAM dan kemudian pilih>tombol.

Untuk menghapus pengguna IAM dari grup, di panel kanan, pilih grup IAM dan kemudian pilih<tombol.

Untuk menambahkan pengguna IAM ke semua grup IAM, pilih>>tombol. Demikian pula, untuk menghapus pengguna IAM dari semua grup, pilih<<tombol.

Untuk memilih beberapa kelompok, pilih mereka secara berurutan. Anda tidak perlu menahan tombol Control. Untuk menghapus grup dari pilihan Anda, cukup pilih untuk kedua kalinya.

3. Setelah selesai menetapkan pengguna IAM ke grup IAM, pilih **Simpan**.

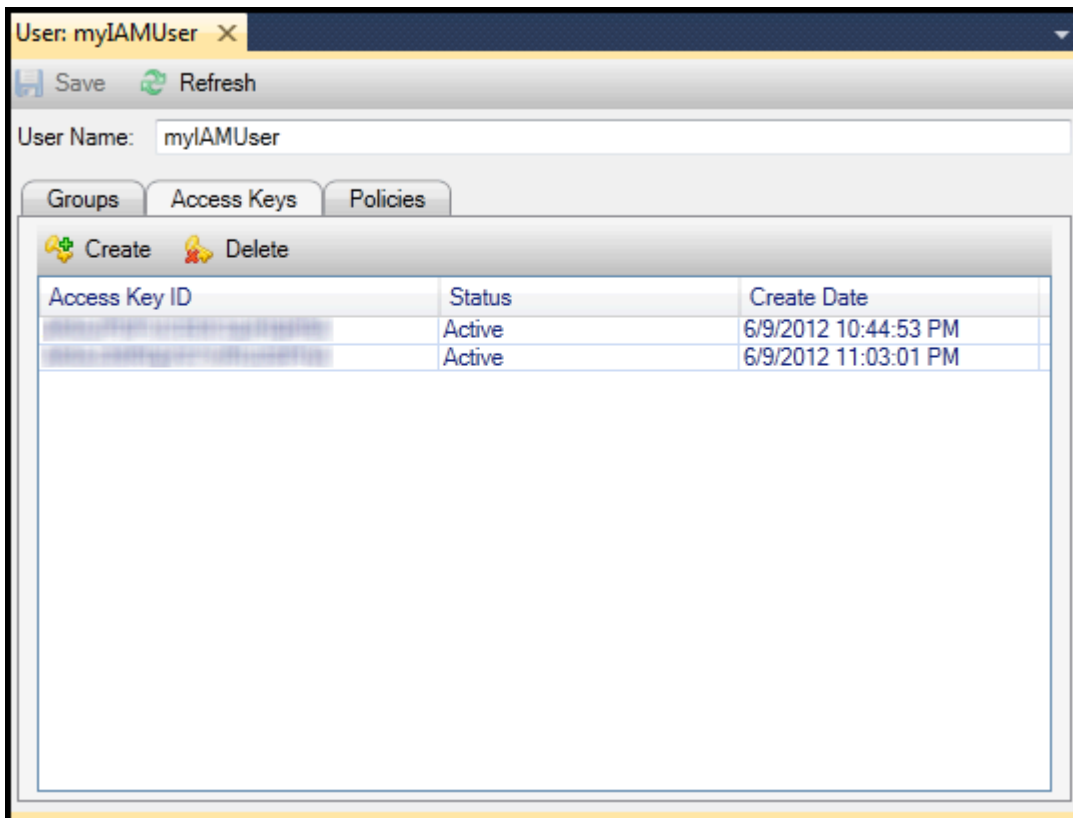
## Menghasilkan Kredensial untuk Pengguna IAM

Dengan Toolkit for Visual Studio, Anda dapat menghasilkan ID kunci akses dan kunci rahasia yang digunakan untuk membuat panggilan API AWS. Kunci ini juga dapat ditentukan untuk mengakses Amazon Web Services melalui Toolkit. Untuk informasi selengkapnya tentang cara menentukan kredensial untuk digunakan dengan Toolkit, lihat [kredensial](#). Untuk informasi selengkapnya tentang cara menangani kredensial dengan aman, lihat [Praktik Terbaik untuk Mengelola AWS Tombol akses](#).

Toolkit tidak dapat digunakan untuk menghasilkan kata sandi bagi pengguna IAM.

Untuk menghasilkan kredensial untuk pengguna IAM

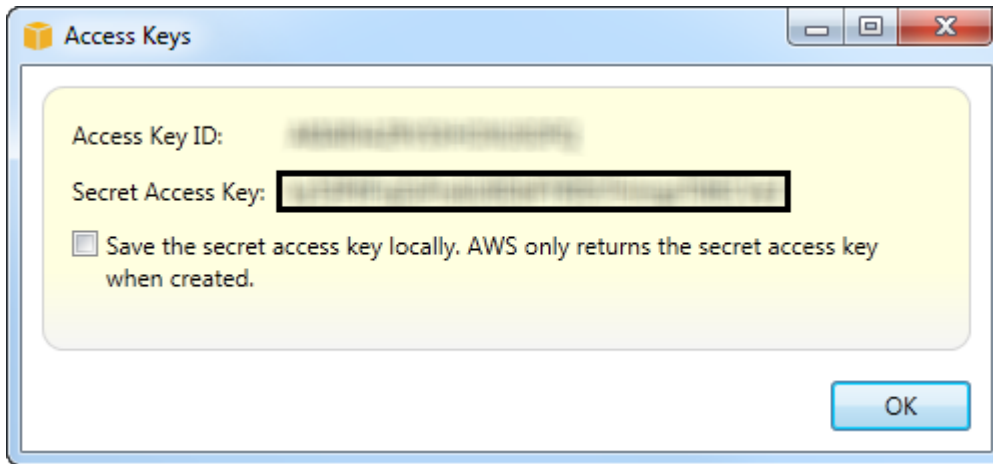
1. Masuk ke **AWSExplorer**, buka menu konteks (klik kanan) dan pilih **Mengedit**.



2. Untuk menghasilkan kredensial, pada **Tombol akses**, pilih **Buat**.

Anda dapat menghasilkan hanya dua set kredensial per pengguna IAM. Jika Anda sudah memiliki dua set kredensial dan perlu membuat set tambahan, Anda harus menghapus salah satu set yang ada.



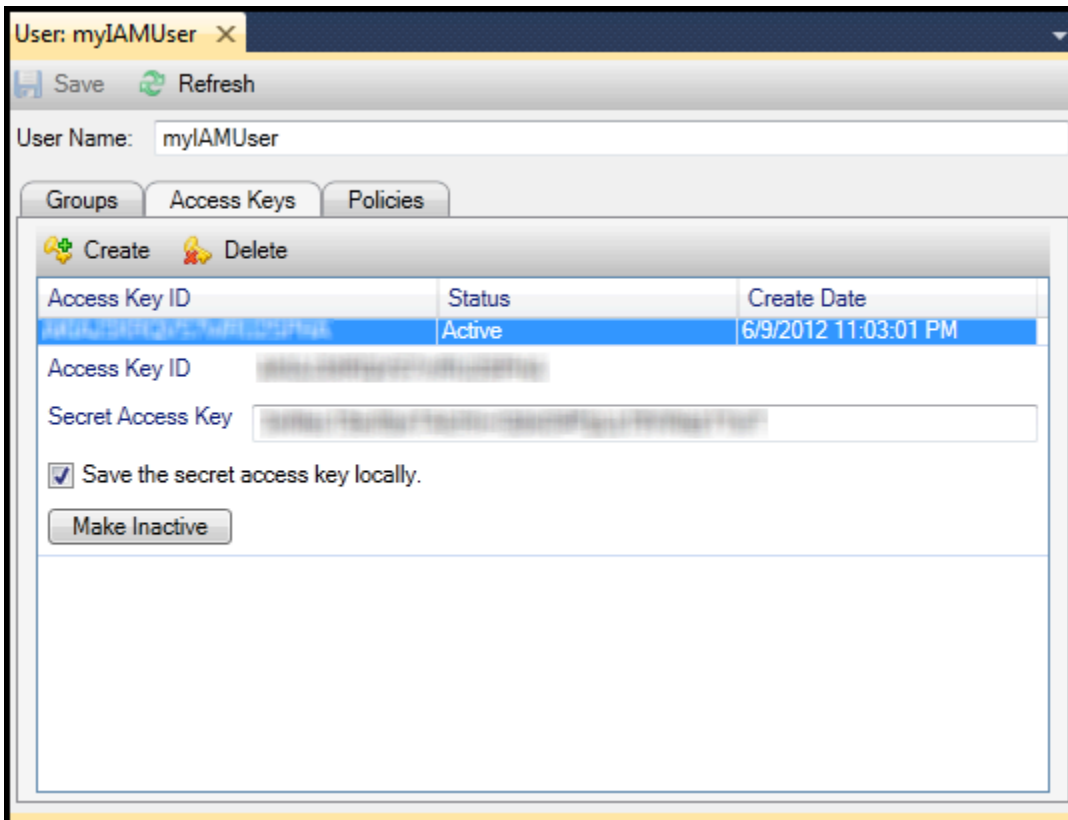


reate credentials for IAM user

Jika Anda ingin Toolkit menyimpan salinan terenkripsi kunci akses rahasia Anda ke drive lokal Anda, pilih Simpan kunci akses rahasia secara lokal. AWS hanya mengembalikan kunci akses rahasia saat dibuat. Anda juga dapat menyalin kunci akses rahasia dari kotak dialog dan menyimpannya di lokasi yang aman.

3. Pilih OKE.

Setelah Anda menghasilkan kredensialnya, Anda dapat melihatnya dari Tombol akses tab. Jika Anda memilih opsi untuk memiliki Toolkit menyimpan kunci rahasia secara lokal, itu akan ditampilkan di sini.



## Create credentials for IAM user

Jika Anda menyimpan kunci rahasia itu sendiri dan juga ingin Toolkit menyimpannya, diSecret Access Key kotak, ketik kunci akses rahasia, dan kemudian pilih Menyimpan kunci akses rahasia secara lokal.

Untuk menonaktifkan kredenensi, pilih Membuat Tidak Aktif. (Anda mungkin melakukan ini jika Anda mencurigai kredensialnya telah dikompromikan. Anda dapat mengaktifkan kembali kredensi jika Anda menerima jaminan bahwa mereka aman.)

## Buat IAM Role

Toolkit for Visual Studio mendukung pembuatan dan konfigurasi IAM role. Sama seperti pengguna dan grup, Anda dapat melampirkan kebijakan ke peran IAM. Anda kemudian dapat mengaitkan IAM role dengan instans Amazon EC2. Hubungan dengan instans EC2 ditangani melalui profil instans, yang merupakan wadah logis untuk peran tersebut. Aplikasi yang dijalankan di instans EC2 secara otomatis diberikan tingkat akses yang ditentukan oleh kebijakan yang terkait dengan IAM role. Hal ini tetap berlaku meskipun aplikasi belum ditentukan lainnya AWS kredenensi.

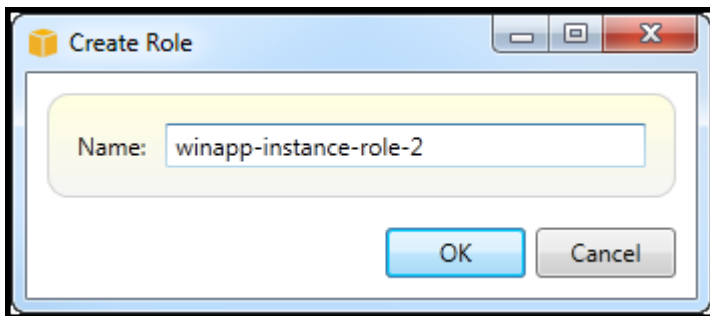
Misalnya, Anda dapat membuat peran dan melampirkan kebijakan ke peran yang membatasi akses ke Amazon S3 saja. Setelah mengaitkan peran ini dengan instans EC2, Anda kemudian dapat

menjalankan aplikasi pada instans tersebut dan aplikasi akan memiliki akses ke Amazon S3, tetapi tidak ke layanan atau sumber daya lainnya. Keuntungan dari pendekatan ini adalah Anda tidak perlu khawatir dengan aman mentransfer dan menyimpan AWS kredensial pada instans EC2.

Untuk informasi lebih lanjut tentang IAM role, buka [Bekerja dengan Peran IAM dalam Panduan Pengguna IAM](#). Untuk contoh program mengakses AWS menggunakan peran IAM yang terkait dengan instans Amazon EC2, buka AWS panduan pengembang untuk [Java](#), [.NET](#), [PHP](#), dan Ruby ([Mengatur Kredensial Menggunakan IAM](#), [Membuat IAM Role](#), dan [Cara menggunakan Kebijakan IAM](#)).

Untuk membuat IAM role

1. Masuk ke AWSExplorer, di bawah Manajemen Identitas dan Akses, buka menu konteks (klik kanan) untuk Peran lalu pilih **Membuat peran**.
2. Di **Membuat peran** kotak dialog, ketik nama untuk peran IAM dan pilih **OK**.



Create IAM role

Peran IAM baru akan muncul di bawah Peran di Manajemen Identitas dan Akses.

Untuk informasi tentang cara membuat kebijakan dan melampirkannya ke peran, lihat [Membuat Kebijakan IAM](#).

## Membuat Kebijakan IAM

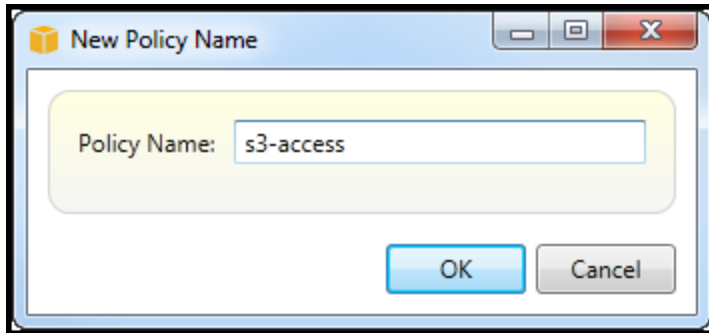
Kebijakan sangat mendasar bagi IAM. Kebijakan dapat dikaitkan dengan IAM pengguna seperti pengguna, grup, atau peran. Kebijakan menentukan tingkat akses yang diaktifkan untuk pengguna, grup, atau peran.

Untuk membuat kebijakan IAM

Masuk ke AWSExplorer, memperluas AWS Identity and Access Management node, kemudian memperluas node untuk jenis entitas (Grup, Peran, atau Pengguna) yang akan Anda lampirkan kebijakan. Misalnya, buka menu konteks untuk peran IAM dan pilih **Mengedit**.

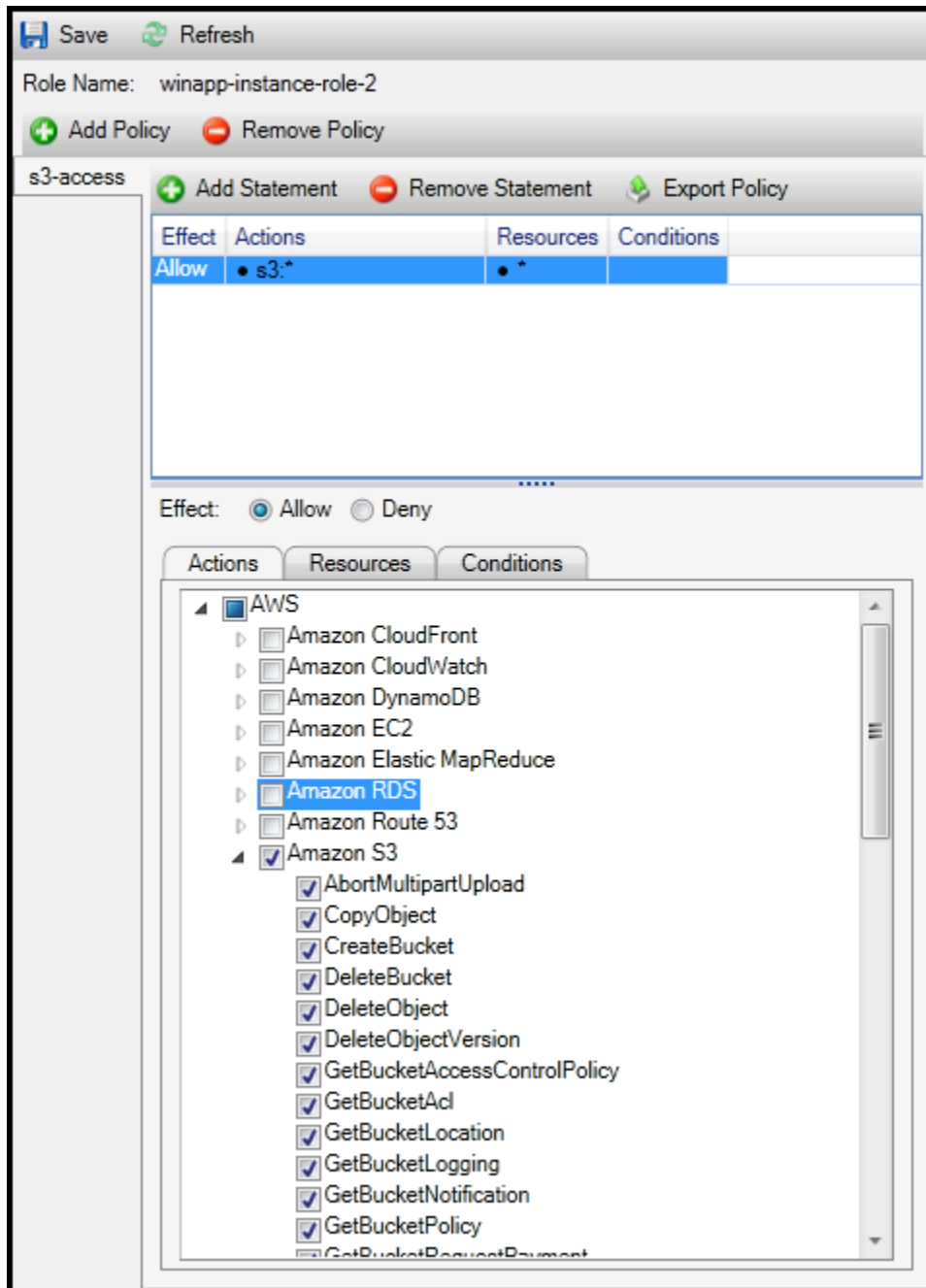
Tab yang terkait dengan peran akan muncul diAWSExplorer. PilihTambahkan Kebijakanlink.

DiNama Kebijakan Barukotak dialog, ketik nama untuk kebijakan (misalnya, s3-akses).



New Policy Name dialog box

Dalam editor kebijakan, tambahkan pernyataan kebijakan untuk menentukan tingkat akses untuk memberikan peran (dalam contoh ini, winapp-instance-role-2 terkait dengan kebijakan. Dalam contoh ini, kebijakan menyediakan akses penuh ke Amazon S3, tetapi tidak ada akses ke sumber daya lainnya.



### Specify IAM policy

Untuk kontrol akses yang lebih tepat, Anda dapat memperluas subnode di editor kebijakan untuk mengizinkan atau melarang tindakan yang terkait dengan Amazon Web Services.

Ketika Anda telah mengedit kebijakan, pilih Simpan link.

# AWS Lambda

Kembangkan dan terapkan fungsi C# Lambda berbasis .NET Core Anda dengan. AWS Toolkit for Visual Studio AWS Lambda adalah layanan komputasi yang memungkinkan Anda menjalankan kode tanpa menyediakan atau mengelola server. Toolkit for Visual Studio AWS Lambda mencakup template proyek .NET Core untuk Visual Studio.

Untuk informasi selengkapnya AWS Lambda, lihat Panduan Pengembang [AWS Lambda](#).

Untuk informasi selengkapnya tentang .NET Core, lihat panduan Microsoft. [.NET Core](#). [Untuk prasyarat.NET Core dan petunjuk penginstalan untuk platform Windows, macOS, dan Linux, lihat .NET Core Downloads.](#)

Topik berikut menjelaskan cara bekerja dengan AWS Lambda menggunakan Toolkit for Visual Studio.

Topik

- [AWS Lambda Proyek Dasar](#)
- [AWS Lambda Proyek Dasar Membuat Gambar Docker](#)
- [Tutorial: Membangun dan Menguji Aplikasi Tanpa Server dengan AWS Lambda](#)
- [Tutorial: Membuat Aplikasi Amazon Rekognition Lambda](#)
- [Tutorial: Menggunakan Amazon Logging Frameworks dengan AWS Lambda untuk Membuat Log Aplikasi](#)

## AWS Lambda Proyek Dasar

Anda dapat membuat fungsi Lambda menggunakan template proyek Microsoft .NET Core, di file. AWS Toolkit for Visual Studio

### Buat Proyek Lambda Inti Visual Studio .NET

Anda dapat menggunakan template dan cetak biru Lambda-Visual Studio untuk membantu mempercepat inisialisasi proyek Anda. Cetak biru Lambda berisi fungsi pra-tertulis yang menyederhanakan pembuatan fondasi proyek yang fleksibel.

**Note**

Layanan Lambda memiliki batas data pada jenis paket yang berbeda. Untuk informasi rinci tentang batas data, lihat topik [kuota Lambda di Panduan Pengguna Lambda AWS](#) .

Untuk membuat proyek Lambda di Visual Studio

1. Dari Visual Studio, perluas menu File, perluas Baru, lalu pilih Project.
2. Dari kotak dialog Proyek Baru, atur kotak drop-down Bahasa, Platform, dan Jenis proyek ke “Semua”, lalu ketik `aws lambda` di bidang Pencarian. Pilih template Proyek AWS Lambda (.NET Core - C #).
3. Di bidang Nama, masukkan **AWSLambdaSample**, tentukan lokasi file yang Anda inginkan, lalu pilih Buat untuk melanjutkan.
4. Dari halaman Select Blueprint, pilih cetak biru Fungsi Kosong, lalu pilih Selesai untuk membuat proyek Visual Studio.

## Tinjau File Proyek

Ada dua file proyek untuk ditinjau: `aws-lambda-tools-defaults.json` dan `Function.cs`.

Contoh berikut menunjukkan `aws-lambda-tools-defaults.json` file, yang secara otomatis dibuat sebagai bagian dari proyek Anda. Anda dapat mengatur opsi build dengan menggunakan bidang dalam file ini.

**Note**

Template proyek di Visual Studio berisi banyak bidang yang berbeda, perhatikan hal-hal berikut:

- `function-handler`: menentukan metode yang berjalan saat fungsi Lambda berjalan
- Menentukan nilai di bidang `function-handler` akan mengisi nilai tersebut di wizard Publish.
- Jika Anda mengganti nama fungsi, kelas, atau perakitan maka Anda juga perlu memperbarui bidang yang sesuai dalam `aws-lambda-tools-defaults.json` file.

```
{
  "Information": [
    "This file provides default values for the deployment wizard inside Visual Studio
    and the AWS Lambda commands added to the .NET Core CLI.",
    "To learn more about the Lambda commands with the .NET Core CLI execute the
    following command at the command line in the project root directory.",
    "dotnet lambda help",
    "All the command line options for the Lambda command can be specified in this
    file."
  ],
  "profile": "default",
  "region": "us-west-2",
  "configuration": "Release",
  "function-architecture": "x86_64",
  "function-runtime": "dotnet8",
  "function-memory-size": 512,
  "function-timeout": 30,
  "function-handler": "AWSLambdaSample::AWSLambdaSample.Function::FunctionHandler"
}
```

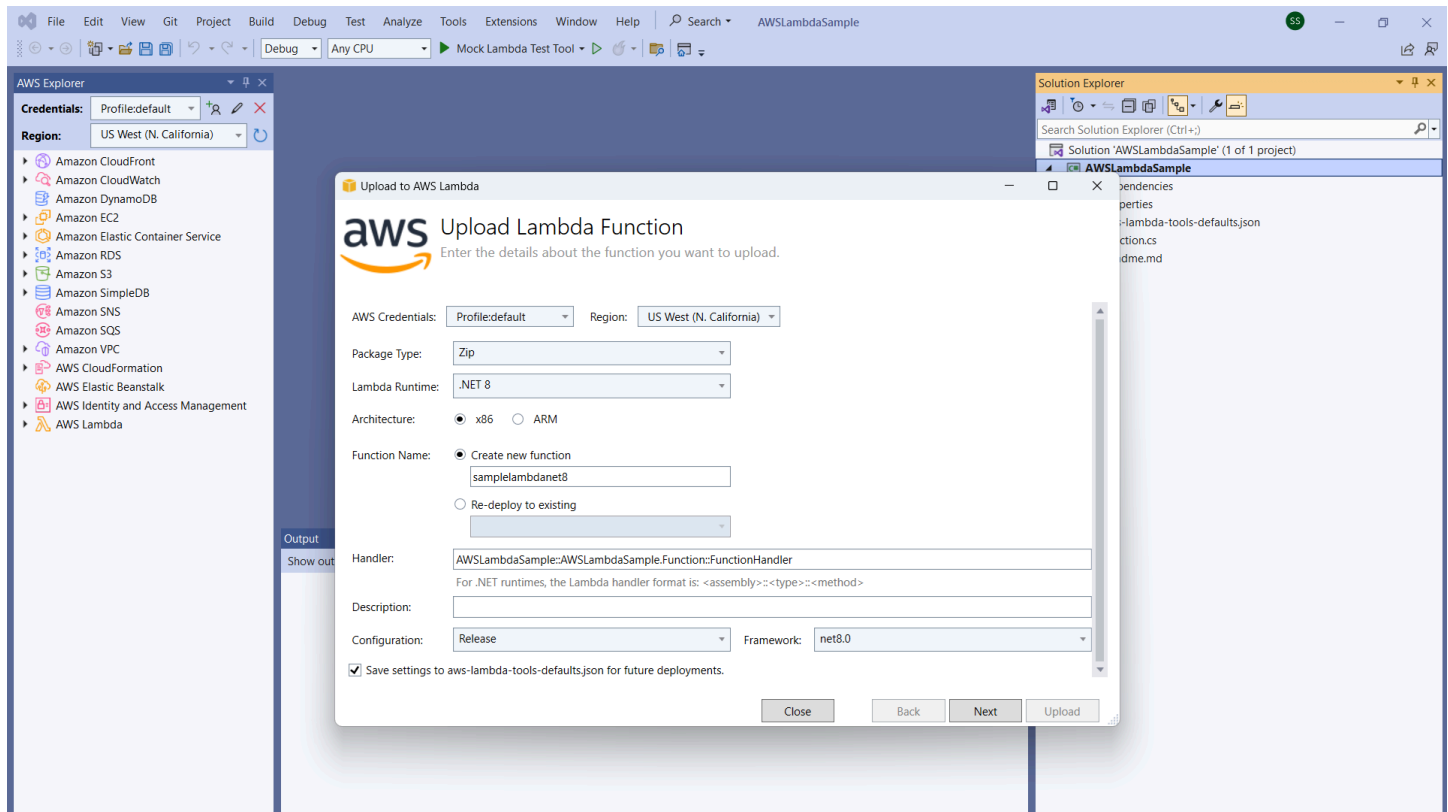
Periksa `Function.cs` file. `Function.cs` mendefinisikan fungsi `c #` untuk mengekspos sebagai fungsi Lambda. Ini `FunctionHandler` adalah fungsi Lambda yang berjalan saat fungsi Lambda berjalan. Dalam proyek ini, ada satu fungsi yang didefinisikan: `FunctionHandler`, yang memanggil `ToUpper()` teks input.

Proyek Anda sekarang siap dipublikasikan ke Lambda.

## Penerbitan ke Lambda


Prosedur dan gambar berikut menunjukkan cara mengunggah fungsi Anda ke Lambda menggunakan AWS Toolkit for Visual Studio






## Menerbitkan fungsi Anda ke Lambda

1. Arahkan ke AWS Explorer dengan memperluas View dan memilih AWS Explorer.
2. Di Solution Explorer, buka menu konteks untuk (klik kanan) proyek yang ingin Anda publikasikan, lalu pilih Publish to AWS Lambda untuk membuka jendela Upload Lambda Function.
3. Dari jendela Upload Lambda Function, lengkapi kolom berikut:
  - a. Jenis Paket: Pilih **Zip**. File ZIP akan dibuat sebagai hasil dari proses pembuatan dan akan diunggah ke Lambda. Atau, Anda dapat memilih Package Type **Image**. [Tutorial: Proyek Lambda Dasar Membuat Gambar Docker](#) menjelaskan cara mempublikasikan menggunakan Package Type. **Image**
  - b. Lambda Runtime: Pilih Lambda Runtime Anda dari menu tarik-turun.
  - c. Arsitektur: Pilih radial untuk arsitektur pilihan Anda.
  - d. Nama Fungsi: Pilih radial untuk Buat fungsi baru, lalu masukkan nama tampilan untuk instance Lambda Anda. Nama ini direferensikan oleh AWS Explorer dan AWS Management Console display.
  - e. Handler: Gunakan bidang ini untuk menentukan fungsi handler. Misalnya: **AWSLambdaSample::AWSLambdaSample.Function::FunctionHandler**.

- f. (Opsional) Deskripsi: Masukkan teks deskriptif untuk ditampilkan dengan instance Anda, dari dalam. AWS Management Console
  - g. Konfigurasi: Pilih konfigurasi pilihan Anda dari menu tarik-turun.
  - h. Framework: Pilih framework pilihan Anda dari menu drop-down.
  - i. Simpan pengaturan: Pilih kotak ini untuk menyimpan pengaturan Anda saat ini `aws-lambda-tools-defaults.json` sebagai default untuk penerapan masa depan.
  - j. Pilih Berikutnya untuk melanjutkan ke jendela Advanced Function Details.
4. Di jendela Advanced Function Details, lengkapi bidang-bidang berikut:
- a. Nama Peran: Pilih peran yang terkait dengan akun Anda. Peran ini menyediakan kredensi sementara untuk setiap panggilan AWS layanan yang dibuat oleh kode dalam fungsi. Jika Anda tidak memiliki peran, gulir untuk menemukan Peran Baru berdasarkan Kebijakan AWS Terkelola di pemilih tarik-turun, lalu pilih. `AWSLambdaBasicExecutionRole` Peran ini memiliki izin akses minimal.
-  **Note**


Akun Anda harus memiliki izin untuk menjalankan `ListPolicies` tindakan IAM, atau daftar Nama Peran akan kosong dan Anda tidak akan dapat melanjutkan.
- b. (Opsional) Jika fungsi Lambda Anda mengakses sumber daya di VPC Amazon, pilih subnet dan grup keamanan.
  - c. (Opsional) Tetapkan variabel lingkungan apa pun yang dibutuhkan fungsi Lambda Anda. Kunci secara otomatis dienkripsi oleh kunci layanan default yang gratis. Atau, Anda dapat menentukan AWS KMS kunci, yang dikenakan biaya. [KMS](#) adalah layanan terkelola yang dapat Anda gunakan untuk membuat dan mengontrol kunci enkripsi yang digunakan untuk mengenkripsi data Anda. Jika Anda memiliki AWS KMS kunci, Anda dapat memilihnya dari daftar.
5. Pilih Upload untuk membuka jendela Uploading Function dan memulai proses upload.

 **Note**

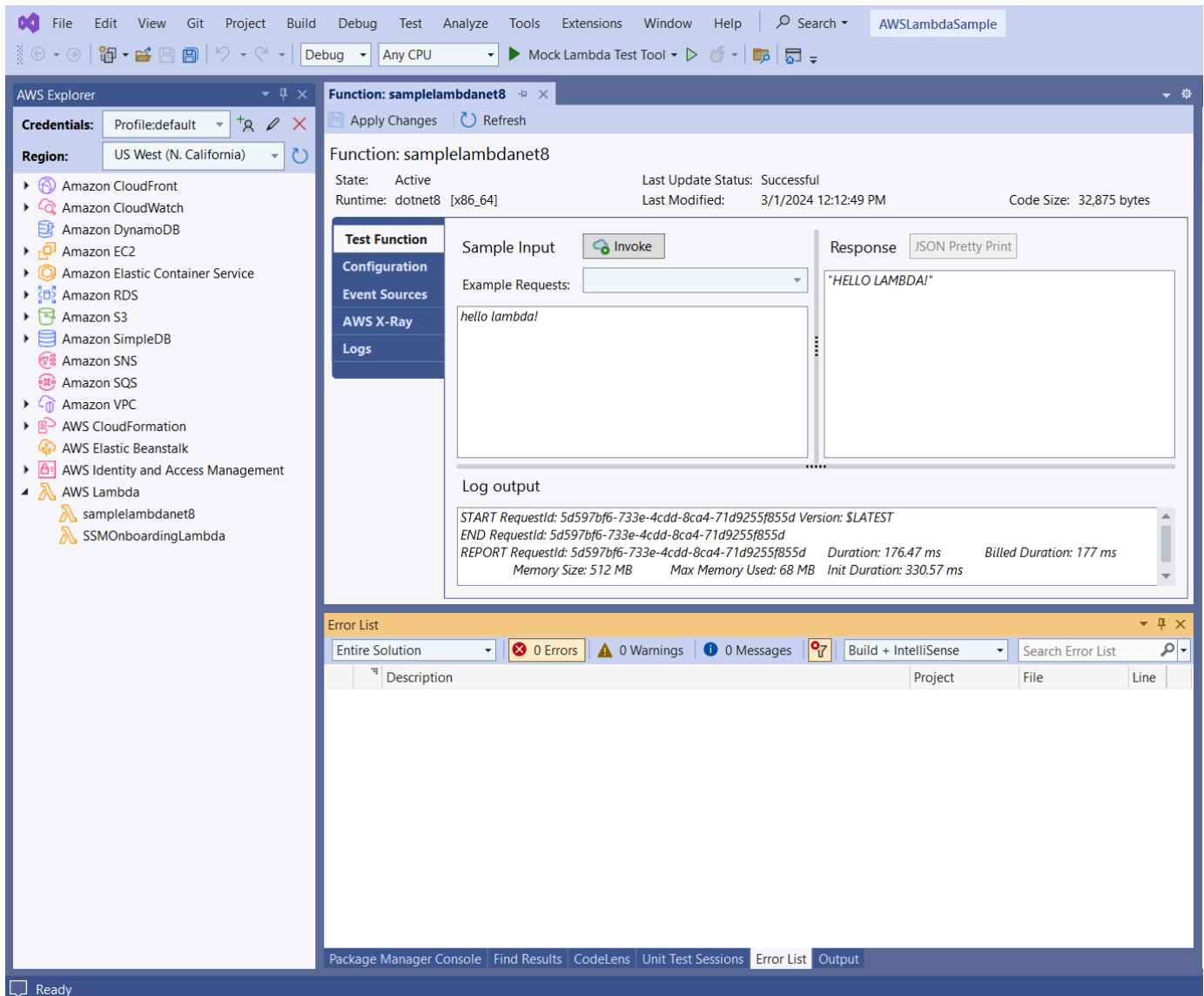
Halaman Uploading Function ditampilkan saat fungsi diunggah ke. AWS Agar wizard tetap terbuka setelah mengunggah sehingga Anda dapat melihat laporan, hapus Wizard tutup otomatis jika berhasil diselesaikan di bagian bawah formulir sebelum unggahan selesai.

Setelah fungsi diunggah, fungsi Lambda Anda aktif. Halaman Function: view terbuka dan menampilkan konfigurasi fungsi Lambda baru Anda.

6. Dari tab Test Function, masukkan `hello lambda!` di kolom input teks dan kemudian pilih Invoke untuk memanggil fungsi Lambda Anda secara manual. Teks Anda muncul di tab Respons, dikonversi ke huruf besar.

 Note

Anda dapat membuka kembali tampilan Function: kapan saja dengan mengklik dua kali pada instance yang digunakan yang terletak di AWS Explorer di bawah node. AWS Lambda



7. (Opsional) Untuk mengonfirmasi bahwa Anda berhasil mempublikasikan fungsi Lambda Anda, masuk ke AWS Management Console dan kemudian pilih Lambda. Konsol menampilkan semua fungsi Lambda yang Anda publikasikan, termasuk yang baru saja Anda buat.

## Membersihkan

Jika Anda tidak akan terus mengembangkan dengan contoh ini, hapus fungsi yang Anda terapkan sehingga Anda tidak ditagih untuk sumber daya yang tidak digunakan di akun Anda.

**Note**

Lambda secara otomatis memantau fungsi Lambda untuk Anda, melaporkan metrik melalui Amazon CloudWatch. Untuk memantau dan memecahkan masalah fungsi Anda, lihat topik [Pemecahan Masalah dan Pemantauan AWS Fungsi Lambda dengan Amazon CloudWatch](#) di Panduan Pengembang. AWS Lambda

Untuk menghapus fungsi Anda

1. Dari AWS Explorer memperluas AWS Lambda node.
2. Klik kanan instance yang Anda gunakan, lalu pilih Hapus.

## AWS Lambda Proyek Dasar Membuat Gambar Docker

Anda dapat menggunakan Toolkit for Visual Studio untuk menyebarkan fungsi AWS Lambda Anda sebagai image Docker. Menggunakan Docker, Anda memiliki kontrol lebih besar atas runtime Anda. Misalnya, Anda dapat memilih runtime kustom seperti .NET 8.0. Anda menerapkan gambar Docker Anda dengan cara yang sama seperti gambar kontainer lainnya. Tutorial ini sangat mirip dengan [Tutorial: Proyek Lambda Dasar](#), dengan dua perbedaan:

- Sebuah Dockerfile disertakan dalam proyek.
- Konfigurasi penerbitan alternatif dipilih.

Untuk informasi tentang gambar kontainer Lambda, lihat [Paket Penerapan Lambda](#) di Panduan Pengembang. AWS Lambda

Untuk informasi tambahan tentang bekerja dengan Lambda AWS Toolkit for Visual Studio, lihat [Menggunakan AWS Lambda Template dalam AWS Toolkit for Visual Studio](#) topik di Panduan Pengguna ini.

## Buat Proyek Lambda Inti Visual Studio .NET

Anda dapat menggunakan template dan cetak biru Lambda Visual Studio untuk membantu mempercepat inisialisasi proyek Anda. Cetak biru Lambda berisi fungsi pra-tertulis yang menyederhanakan pembuatan fondasi proyek yang fleksibel.

## Untuk membuat proyek Visual Studio .NET Core Lambda

1. Dari Visual Studio, perluas menu File, perluas Baru, lalu pilih Project.
2. Dari kotak dialog Proyek Baru, atur kotak drop-down Bahasa, Platform, dan Jenis proyek ke “Semua”, lalu ketik **aws lambda** di bidang Pencarian. Pilih template Proyek AWS Lambda (.NET Core - C #).
3. Di bidang Nama Proyek, masukkan **AWSLambdaDocker**, tentukan lokasi file Anda, lalu pilih Buat.
4. Pada halaman Select Blueprint, pilih blueprint .NET 8 (Container Image), lalu pilih Finish untuk membuat proyek Visual Studio. Anda sekarang dapat meninjau struktur dan kode proyek.

## Meninjau File Proyek

Bagian berikut memeriksa tiga file proyek yang dibuat oleh cetak biru .NET 8 (Container Image):

1. Dockerfile
2. aws-lambda-tools-defaults.json
3. Function.cs

### 1. Dockerfile

A `Dockerfile` melakukan tiga tindakan utama:

- **FROM:** Menetapkan gambar dasar untuk digunakan untuk gambar ini. Gambar dasar ini menyediakan .NET Runtime, Lambda runtime, dan skrip shell yang menyediakan titik masuk untuk proses Lambda.NET.
- **WORKDIR:** Menetapkan direktori kerja internal gambar sebagai `/var/task`.
- **COPY:** Akan menyalin file yang dihasilkan dari proses pembuatan dari lokasi lokalnya ke direktori kerja gambar.

Berikut ini adalah `Dockerfile` tindakan opsional yang dapat Anda tentukan:

- **ENTRYPOINT:** Gambar dasar sudah menyertakan `ENTRYPOINT`, yang merupakan proses start-up yang dijalankan saat gambar dimulai. Jika Anda ingin menentukan sendiri, maka Anda mengesampingkan titik masuk dasar itu.

- **CMD:** Menginstruksikan kode kustom AWS mana yang ingin Anda eksekusi. Ini mengharapkan nama yang sepenuhnya memenuhi syarat untuk metode kustom Anda. Baris ini perlu disertakan langsung di Dockerfile atau dapat ditentukan selama proses publikasi.

```
# Example of alternative way to specify the Lambda target method rather than during
the publish process.
CMD [ "AWSLambdaDocker::AWSLambdaDocker.Function::FunctionHandler"]
```

Berikut ini adalah contoh dari Dockerfile yang dibuat oleh cetak biru.NET 8 (Container Image).

```
FROM public.ecr.aws/lambda/dotnet:8

WORKDIR /var/task

# This COPY command copies the .NET Lambda project's build artifacts from the host
machine into the image.
# The source of the COPY should match where the .NET Lambda project publishes its build
artifacts. If the Lambda function is being built
# with the AWS .NET Lambda Tooling, the `--docker-host-build-output-dir` switch
controls where the .NET Lambda project
# will be built. The .NET Lambda project templates default to having `--docker-host-
build-output-dir`
# set in the aws-lambda-tools-defaults.json file to "bin/Release/lambda-publish".
#
# Alternatively Docker multi-stage build could be used to build the .NET Lambda project
inside the image.
# For more information on this approach checkout the project's README.md file.
COPY "bin/Release/lambda-publish" .
```

## 2. aws-lambda-tools-defaults.json

`aws-lambda-tools-defaults.json` File ini digunakan untuk menentukan nilai default untuk wizard penyebaran Toolkit for Visual Studio dan .NET Core CLI. Daftar berikut menjelaskan bidang yang dapat Anda atur dalam `aws-lambda-tools-defaults.json` file Anda.

- **profile:** menetapkan AWS profil Anda.
- **region:** menetapkan AWS wilayah tempat sumber daya Anda disimpan.
- **configuration:** menetapkan konfigurasi yang digunakan untuk mempublikasikan fungsi Anda.
- **package-type:** menyetel tipe paket penerapan ke gambar kontainer atau arsip file.zip.

- `function-memory-size`: mengatur alokasi memori untuk fungsi Anda dalam MB.
- `function-timeout`: Timeout adalah jumlah waktu maksimum dalam hitungan detik yang dapat dijalankan oleh fungsi Lambda. Anda dapat menyesuaikan ini dengan penambahan 1 detik hingga nilai maksimum 15 menit.
- `docker-host-build-output-dir`: menyetel direktori keluaran dari proses pembuatan yang berkorelasi dengan instruksi di `Dockerfile`
- `image-command`: adalah nama yang sepenuhnya memenuhi syarat untuk metode Anda, kode yang Anda inginkan untuk menjalankan fungsi Lambda. Sintaksnya adalah: `{Assembly}:: {Namespace}. {ClassName}:: {MethodName}`. Untuk informasi selengkapnya, lihat [Tanda tangan Handler](#). Pengaturan `image-command` di sini telah mengisi nilai ini di wizard Publish Visual Studio nanti.

Berikut ini adalah contoh dari sebuah `aws-lambda-tools-defaults.json` yang dibuat oleh cetak biru .NET 8 (Container Image).

```
{
  "Information": [
    "This file provides default values for the deployment wizard inside Visual Studio",
    "and the AWS Lambda commands added to the .NET Core CLI.",
    "To learn more about the Lambda commands with the .NET Core CLI execute the",
    "following command at the command line in the project root directory.",
    "dotnet lambda help",
    "All the command line options for the Lambda command can be specified in this",
    "file."
  ],
  "profile": "default",
  "region": "us-west-2",
  "configuration": "Release",
  "package-type": "image",
  "function-memory-size": 512,
  "function-timeout": 30,
  "image-command": "AWSLambdaDocker::AWSLambdaDocker.Function::FunctionHandler",
  "docker-host-build-output-dir": "./bin/Release/lambda-publish"
}
```



### 3. Function.cs

Function.csFile mendefinisikan fungsi c # yang akan diekspos sebagai fungsi Lambda. FunctionHandlerIni adalah fungsi Lambda yang berjalan saat fungsi Lambda berjalan. Dalam proyek ini, FunctionHandler panggilan ToUpper() pada teks input.

### Publikasikan ke Lambda

Gambar Docker yang dihasilkan oleh proses pembuatan diunggah ke Amazon Elastic Container Registry (Amazon ECR) Registry ECR). Amazon ECR adalah registri kontainer Docker yang dikelola sepenuhnya yang Anda gunakan untuk menyimpan, mengelola, dan menyebarkan gambar kontainer Docker. Amazon ECR menghosting gambar, yang kemudian dirujuk oleh Lambda untuk menyediakan fungsionalitas Lambda yang diprogram saat dipanggil.


Untuk mempublikasikan fungsi Anda ke Lambda

1. Dari Solution Explorer, buka menu konteks untuk (klik kanan) proyek, lalu pilih Publish AWS Lambda untuk membuka jendela Upload Lambda Function.
2. Dari halaman Upload Lambda Function, lakukan hal berikut:

The screenshot shows the 'Upload to AWS Lambda' dialog box. The title bar reads 'Upload to AWS Lambda'. The main header features the AWS logo and the text 'Upload Lambda Function' with the subtitle 'Enter the details about the function you want to upload.' Below this, there are several configuration sections: 'AWS Credentials' with a dropdown for 'Profile: Default' and 'Region' set to 'US West (Oregon)'; 'Package Type' set to 'Image'; 'Lambda Runtime' set to 'Not Applicable to Image based Functions'; 'Architecture' with radio buttons for 'x86' (selected) and 'ARM'; 'Function Name' with radio buttons for 'Create new function' (selected) and 'Re-deploy to existing', with a text input field containing 'LambdafunctionDocker'; 'Description' with an empty text area; 'Image Command' with the text 'AWSLambdaDocker::AWSLambdaDocker.Function::FunctionHandler'; 'Image Repo' set to 'awslambdadocker' and 'Image Tag' set to 'latest'. At the bottom, there are four buttons: 'Close', 'Back', 'Next', and 'Upload'.


- a. Untuk Package Type, **Image** telah secara otomatis dipilih sebagai Package Type Anda karena wizard publikasi mendeteksi sebuah `Dockerfile` dalam proyek Anda.
- b. Untuk Nama Fungsi, masukkan nama tampilan untuk instance Lambda Anda. Nama ini adalah nama referensi yang ditampilkan di AWS Explorer di Visual Studio dan AWS Management Console.
- c. Untuk Deskripsi, masukkan teks untuk ditampilkan dengan instance Anda di AWS Management Console.
- d. Untuk Image Command, masukkan path yang sepenuhnya memenuhi syarat ke metode yang Anda inginkan untuk menjalankan fungsi Lambda:

**AWSLambdaDocker::AWSLambdaDocker.Function::FunctionHandler**

 Note

Nama metode apa pun yang dimasukkan di sini akan mengganti instruksi CMD apa pun di dalam `Dockerfile`. Memasuki Perintah Gambar hanya opsional JIKA Anda `Dockerfile` menyertakan a CMD untuk menginstruksikan cara meluncurkan fungsi Lambda.

- e. Untuk Image Repo, masukkan nama Amazon Elastic Container Registry yang baru atau yang sudah ada. Gambar Docker yang dibuat oleh proses pembuatan diunggah ke registri ini. Definisi Lambda yang sedang diterbitkan akan merujuk pada gambar Amazon ECR itu.
  - f. Untuk Tag Gambar, masukkan tag Docker untuk diasosiasikan dengan gambar Anda di repositori.
  - g. Pilih Selanjutnya.
3. Pada halaman Detail Fungsi Lanjutan, di Nama Peran pilih peran yang terkait dengan akun Anda. Peran ini digunakan untuk memberikan kredensial sementara untuk setiap panggilan Amazon Web Services yang dibuat oleh kode dalam fungsi. Jika Anda tidak memiliki peran, pilih Peran Baru berdasarkan Kebijakan AWS Terkelola, lalu pilih `AWSLambdaBasicExecutionRole`.

 Note

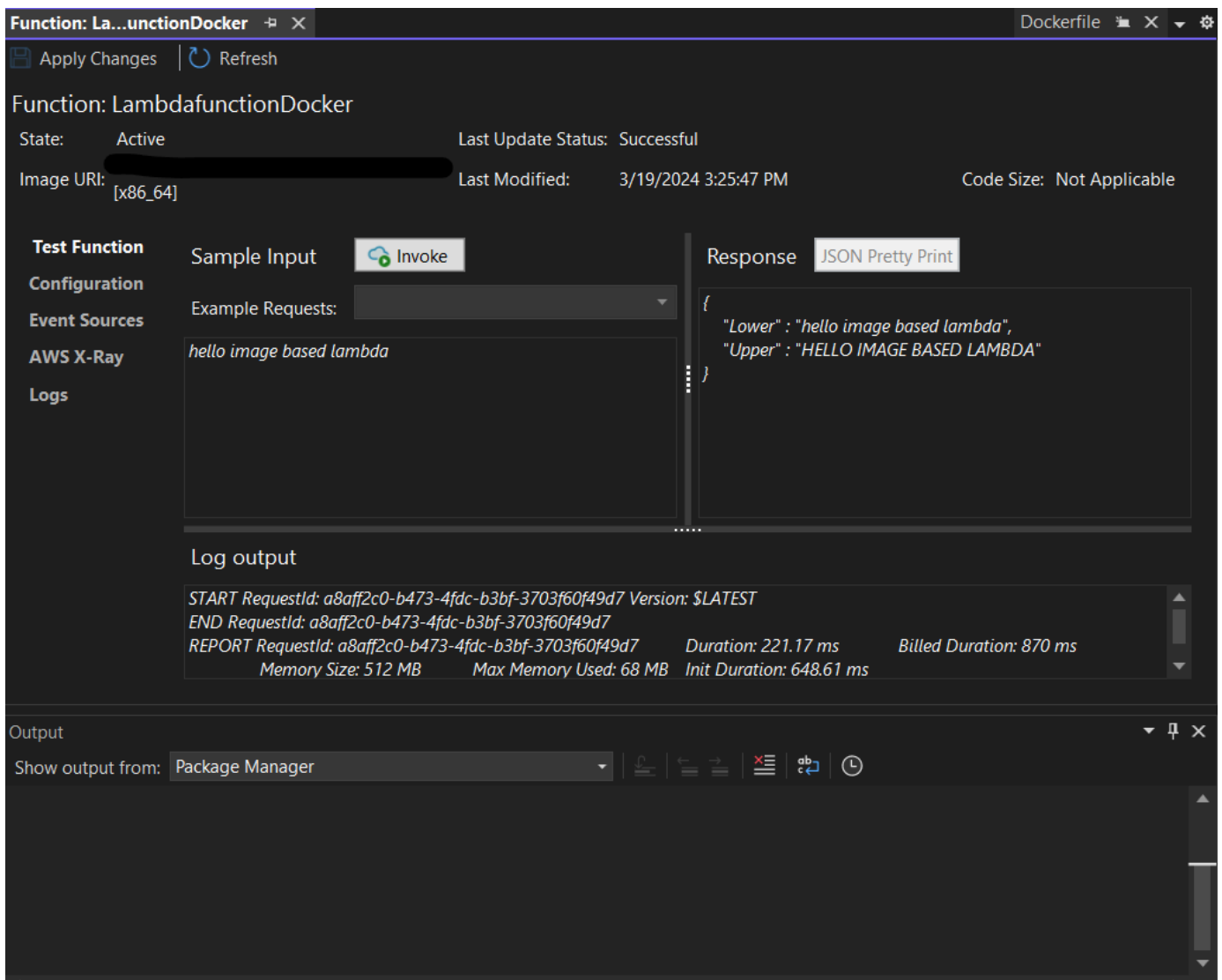
Akun Anda harus memiliki izin untuk menjalankan `ListPolicies` tindakan IAM, atau daftar Nama Peran akan kosong.

4. Pilih Unggah untuk memulai proses pengunggahan dan penerbitan.

**Note**

Halaman Uploading Function ditampilkan saat fungsi sedang mengunggah. Proses publikasi kemudian membangun gambar berdasarkan parameter konfigurasi, membuat repositori Amazon ECR jika perlu, mengunggah gambar ke dalam repositori, dan membuat Lambda yang mereferensikan repo itu dengan gambar itu. Setelah fungsi diunggah, halaman Fungsi akan terbuka dan menampilkan konfigurasi fungsi Lambda baru Anda.

5. Untuk memanggil fungsi Lambda secara manual, pada tab Test Function, **hello image based lambda** masukkan ke bidang input teks bebas permintaan dan kemudian pilih Invoke. Teks Anda, dikonversi ke huruf besar, akan muncul di Respons.



Function: La...unctionDocker

Apply Changes Refresh

Function: LambdafunctionDocker

State: Active Last Update Status: Successful

Image URI: [x86\_64] Last Modified: 3/19/2024 3:25:47 PM Code Size: Not Applicable

**Test Function** Sample Input **Invoke** Response **JSON Pretty Print**

Configuration Example Requests: hello image based lambda

Event Sources

AWS X-Ray

Logs

Log output

```
START RequestId: a8aff2c0-b473-4fdc-b3bf-3703f60f49d7 Version: $LATEST
END RequestId: a8aff2c0-b473-4fdc-b3bf-3703f60f49d7
REPORT RequestId: a8aff2c0-b473-4fdc-b3bf-3703f60f49d7    Duration: 221.17 ms    Billed Duration: 870 ms
Memory Size: 512 MB    Max Memory Used: 68 MB    Init Duration: 648.61 ms
```

Output

Show output from: Package Manager

6. Untuk melihat repositori, di AWS Explorer, di bawah Amazon Elastic Container Service, pilih Repositori.

Anda dapat membuka kembali tampilan Function: kapan saja dengan mengklik dua kali pada instance yang digunakan yang terletak di AWS Explorer di bawah node. AWS Lambda

#### Note

Jika jendela AWS Explorer Anda tidak terbuka, Anda dapat merambungkannya melalui View -> AWS Explorer

7. Perhatikan opsi konfigurasi khusus gambar tambahan pada tab Konfigurasi. Tab ini menyediakan cara untuk mengganti ENTRYPOINT, CMD, dan WORKDIR itu mungkin telah ditentukan dalam Dockerfile. Deskripsi adalah deskripsi yang Anda masukkan (jika ada) selama upload/publish.

## Membersihkan

Jika Anda tidak akan terus mengembangkan dengan contoh ini, ingatlah untuk menghapus fungsi dan gambar ECR yang digunakan sehingga Anda tidak ditagih untuk sumber daya yang tidak digunakan di akun Anda.

- Fungsi dapat dihapus dengan mengklik kanan instance yang Anda gunakan yang terletak di AWS Explorer di bawah node. AWS Lambda
- Repositori dapat dihapus di AWS Explorer di bawah Amazon Elastic Container Service -> Repositori.

## Langkah Berikutnya

Untuk informasi tentang membuat dan menguji gambar Lambda, lihat [Menggunakan Gambar Kontainer dengan Lambda](#).

[Untuk informasi tentang penerapan gambar kontainer, izin, dan pengaturan konfigurasi utama, lihat Mengonfigurasi Fungsi.](#)

# Tutorial: Membangun dan Menguji Aplikasi Tanpa Server dengan AWS Lambda

Anda dapat membangun aplikasi Lambda tanpa server dengan menggunakan template. AWS Toolkit for Visual Studio Template proyek Lambda menyertakan satu untuk Aplikasi AWS Tanpa Server, yang merupakan AWS Toolkit for Visual Studio implementasi dari Model Aplikasi AWS [Tanpa Server](#) (SAM). AWS Dengan menggunakan jenis proyek ini, Anda dapat mengembangkan kumpulan AWS Lambda fungsi dan menyebarkannya dengan AWS sumber daya yang diperlukan sebagai keseluruhan aplikasi, menggunakan AWS CloudFormation untuk mengatur penyebaran.

Untuk prasyarat dan informasi tentang pengaturan AWS Toolkit for Visual Studio, lihat [Menggunakan Template AWS Lambda di Toolkit for Visual Studio](#). AWS

## Topik

- [Buat Proyek Aplikasi AWS Tanpa Server Baru](#)
- [Meninjau file Aplikasi Tanpa Server](#)
- [Menerapkan Aplikasi Tanpa Server](#)
- [Uji Aplikasi Tanpa Server](#)

## Buat Proyek Aplikasi AWS Tanpa Server Baru

AWS Proyek Aplikasi Tanpa Server membuat fungsi Lambda dengan template tanpa server. AWS CloudFormation AWS CloudFormation template memungkinkan Anda untuk menentukan sumber daya tambahan seperti database, menambahkan peran IAM, dan menyebarkan beberapa fungsi pada satu waktu. Ini berbeda dari proyek AWS Lambda, yang berfokus pada pengembangan dan penerapan fungsi Lambda tunggal.

Prosedur berikut menjelaskan cara membuat Proyek Aplikasi AWS Tanpa Server baru.

1. Dari Visual Studio, perluas menu File, perluas Baru, lalu pilih Project.
2. Di kotak dialog Proyek Baru, pastikan bahwa kotak drop-down Bahasa, Platform, dan Jenis Proyek diatur ke “Semua...” dan masukkan **aws lambda** di bidang Pencarian.
3. Pilih template AWS Serverless Application with Tests (.NET Core - C#).

**Note**

Ada kemungkinan bahwa template Aplikasi AWS Tanpa Server dengan Tes (.NET Core - C#) mungkin tidak terisi di bagian atas hasil.

4. Klik Berikutnya untuk membuka dialog Configure your new project.
5. Dari dialog Konfigurasi proyek baru Anda, masukkan **ServerlessPowertools** untuk Nama, lalu lengkapi bidang yang tersisa sesuai preferensi Anda. Pilih Buat tombol untuk melanjutkan ke dialog Select Blueprint.
6. Dari dialog Select Blueprint pilih Powertools untuk AWS Lambda cetak biru, lalu pilih Selesai untuk membuat proyek Visual Studio.

## Meninjau file Aplikasi Tanpa Server

Bagian berikut memberikan tampilan rinci pada tiga file Aplikasi Tanpa Server yang dibuat untuk proyek Anda:

1. template tanpa server
2. Functions.cs
3. aws-lambda-tools-defaults.json

### 1. tanpa servers.template

`serverless.templateFile` adalah AWS CloudFormation template untuk mendeklarasikan fungsi Tanpa Server dan sumber daya lainnya. AWS File yang disertakan dengan proyek ini berisi deklarasi untuk satu fungsi Lambda yang akan diekspos melalui Amazon API Gateway sebagai HTTP `*Get*` operasi. Anda dapat mengedit template ini untuk menyesuaikan fungsi yang ada atau menambahkan lebih banyak fungsi dan sumber daya lain yang diperlukan oleh aplikasi Anda.

Berikut ini adalah contoh `serverless.template` file:

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Transform": "AWS::Serverless-2016-10-31",
  "Description": "An AWS Serverless Application.",
  "Resources": {
    "Get": {
```

```

    "Type": "AWS::Serverless::Function",
    "Properties": {
      "Architectures": [
        "x86_64"
      ],
      "Handler": "ServerlessPowertools::ServerlessPowertools.Functions::Get",
      "Runtime": "dotnet8",
      "CodeUri": "",
      "MemorySize": 512,
      "Timeout": 30,
      "Role": null,
      "Policies": [
        "AWSLambdaBasicExecutionRole"
      ],
      "Environment": {
        "Variables": {
          "POWERTOOLS_SERVICE_NAME": "ServerlessGreeting",
          "POWERTOOLS_LOG_LEVEL": "Info",
          "POWERTOOLS_LOGGER_CASE": "PascalCase",
          "POWERTOOLS_TRACER_CAPTURE_RESPONSE": true,
          "POWERTOOLS_TRACER_CAPTURE_ERROR": true,
          "POWERTOOLS_METRICS_NAMESPACE": "ServerlessGreeting"
        }
      },
      "Events": {
        "RootGet": {
          "Type": "Api",
          "Properties": {
            "Path": "/",
            "Method": "GET"
          }
        }
      }
    }
  },
  "Outputs": {
    "ApiURL": {
      "Description": "API endpoint URL for Prod environment",
      "Value": {
        "Fn::Sub": "https://${ServerlessRestApi}.execute-api.
${AWS::Region}.amazonaws.com/Prod/"
      }
    }
  }
}

```

```
}  
}
```

Perhatikan bahwa banyak bidang `...AWS::Serverless::Function...` deklarasi mirip dengan bidang penyebaran proyek Lambda. Powertools Logging, Metrics, dan Tracing dikonfigurasi melalui variabel lingkungan berikut:

- `POWERTOOLS_SERVICE_NAME= ServerlessGreeting`
- `PowerTools_log_level=Info`
- `POWERTOOLS_LOGGER_CASE = PascalCase`
- `PowerTools_TRACER_CAPTURE_RESPONSE=Benar`
- `PowerTools_TRACER_CAPTURE_ERROR=Benar`
- `POWERTOOLS_METRICS_NAMESPACE= ServerlessGreeting`

Untuk definisi dan detail tambahan tentang variabel lingkungan, lihat situs web [Powertools untuk AWS Lambda referensi](#).

## 2. Functions.cs

`Functions.cs` adalah file kelas yang berisi metode C# yang dipetakan ke satu fungsi yang dideklarasikan dalam file template. Fungsi Lambda merespons HTTP Get metode dari API Gateway. Berikut ini adalah contoh `Functions.cs` file:

```
public class Functions  
{  
    [Logging(LogEvent = true, CorrelationIdPath = CorrelationIdPaths.ApiGatewayRest)]  
    [Metrics(CaptureColdStart = true)]  
    [Tracing(CaptureMode = TracingCaptureMode.ResponseAndError)]  
    public APIGatewayProxyResponse Get(APIGatewayProxyRequest request, ILambdaContext  
context)  
    {  
        Logger.LogInformation("Get Request");  
  
        var greeting = GetGreeting();  
  
        var response = new APIGatewayProxyResponse  
        {  
            StatusCode = (int)HttpStatusCode.OK,  

```



```
        Body = greeting,
        Headers = new Dictionary (string, string) { { "Content-Type", "text/
plain" } }
    };

    return response;
}

[Tracing(SegmentName = "GetGreeting Method")]
private static string GetGreeting()
{
    Metrics.AddMetric("GetGreeting_Invocations", 1, MetricUnit.Count);

    return "Hello Powertools for AWS Lambda (.NET)";
}
}
```

### 3. aws-lambda-tools-defaults.json

`aws-lambda-tools-defaults.json` menyediakan nilai default untuk wizard AWS penerapan di dalam Visual Studio dan AWS Lambda perintah yang ditambahkan ke .NET Core CLI. Berikut ini adalah contoh `aws-lambda-tools-defaults.json` file yang disertakan dengan proyek ini:


```
{
  "profile": "Default",
  "region": "us-east-1",
  "configuration": "Release",
  "s3-prefix": "ServerlessPowertools/",
  "template": "serverless.template",
  "template-parameters": ""
}
```

## Menerapkan Aplikasi Tanpa Server

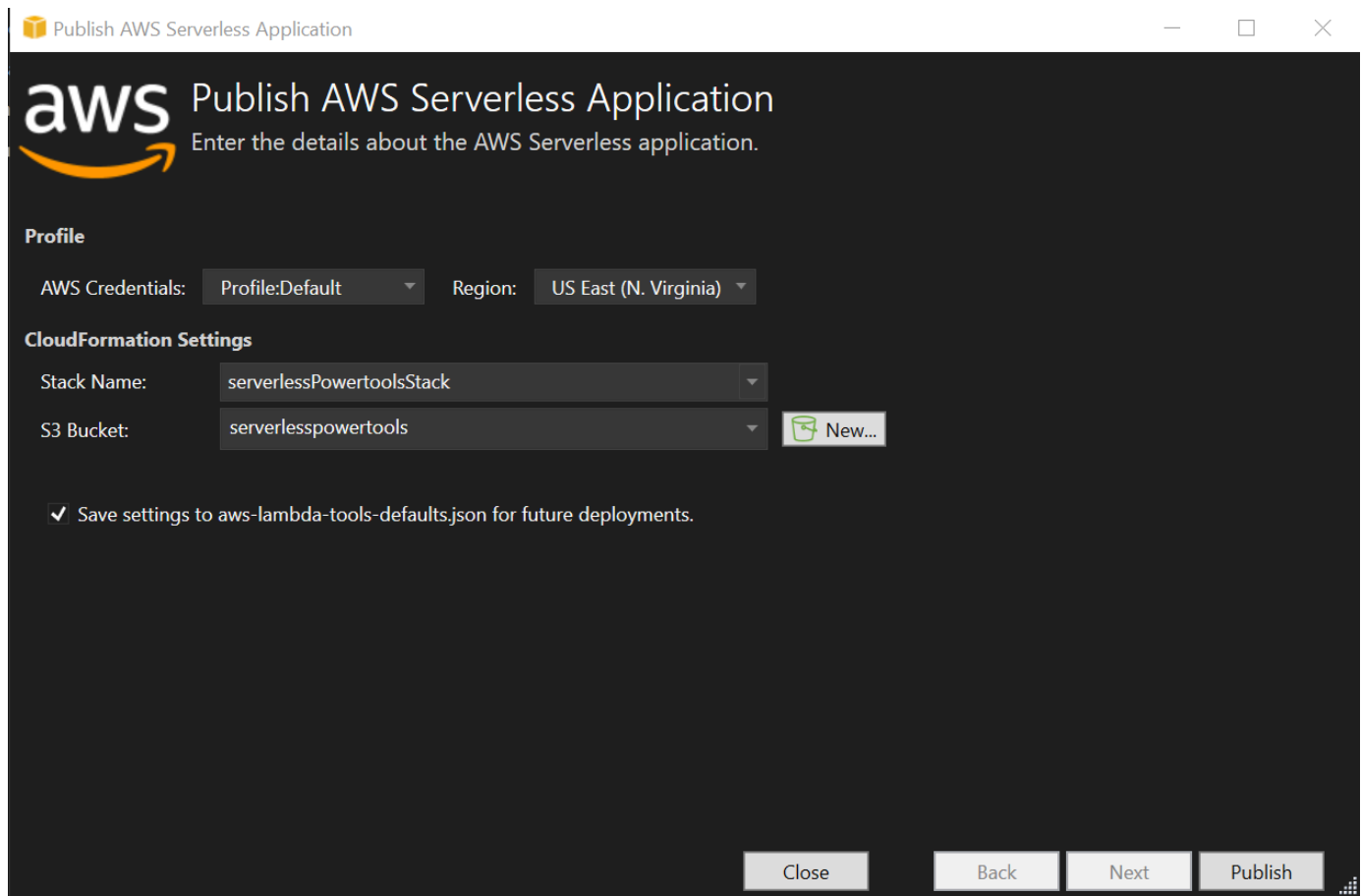
Untuk menerapkan aplikasi tanpa server Anda, selesaikan langkah-langkah berikut

1. Dari Solution Explorer, buka menu konteks untuk (klik kanan) proyek Anda dan pilih Publish to AWS Lambda untuk membuka dialog Publish AWS Serverless Application.
2. Dari dialog Publish AWS Serverless Application, masukkan nama untuk wadah AWS CloudFormation tumpukan di bidang Stack Name.

3. Di bidang S3 Bucket, pilih bucket Amazon S3 yang akan diunggah bundel aplikasi Anda atau pilih New... tombol dan masukkan nama bucket Amazon S3 baru. Kemudian pilih Publish to publish untuk menyebarkan aplikasi Anda.

 Note

AWS CloudFormation Tumpukan dan Bucket Amazon S3 Anda harus ada di wilayah yang sama AWS . Pengaturan yang tersisa untuk proyek Anda ditentukan dalam `serverless.template` file.



4. Jendela tampilan Stack terbuka selama proses penerbitan, saat penerapan selesai, bidang Status menampilkan:CREATE\_COMPLETE.

The screenshot displays the AWS Toolkit for Visual Studio interface. At the top, there are tabs for 'aws-lambda-to...-defaults.json', 'Functions.cs', 'serverless.template', 'Readme.md', and 'serverlessPowertools'. Below the tabs, there are buttons for 'Connect to Instance', 'Delete Stack', 'Cancel Update', and 'Refresh'. The main area shows the stack details for 'serverlessPowertoolsStack', including its name, status ('CREATE COMPLETE'), creation time (3/29/2024 12:44:49 PM), and description ('An AWS Serverless Application.'). The AWS Serverless URL is provided as <https://.amazonaws.com/Prod>. Below this, there is a table of events.

Resources	Time	Type	Logical ID	Physical ID	Status	Reason
Monitoring	3/29/2024 12:45:26 PM	AWS::CloudFormation::Stack	serverlessPowertoolsStack	arn:aws:cloudformation:us-east-1:508...	CREATE_COMPLETE	
Template	3/29/2024 12:45:25 PM	AWS::ApiGateway::Stage	ServerlessRestApiProdStage	Prod	CREATE_COMPLETE	
Parameters	3/29/2024 12:45:25 PM	AWS::ApiGateway::Stage	ServerlessRestApiProdStage	Prod	CREATE_IN_PROGRESS	Resour
Outputs	3/29/2024 12:45:24 PM	AWS::ApiGateway::Stage	ServerlessRestApiProdStage		CREATE_IN_PROGRESS	
	3/29/2024 12:45:23 PM	AWS::Lambda::Function	Get	serverlessPowertoolsStack-Get-Lgaks	CREATE_COMPLETE	
	3/29/2024 12:45:23 PM	AWS::ApiGateway::Deployment	ServerlessRestApiDeployment9d78fb6c57	qpdntli	CREATE_COMPLETE	
	3/29/2024 12:45:23 PM	AWS::ApiGateway::Deployment	ServerlessRestApiDeployment9d78fb6c57	qpdntli	CREATE_IN_PROGRESS	Resour
	3/29/2024 12:45:22 PM	AWS::Lambda::Permission	GetRootGetPermissionProd	serverlessPowertoolsStack-GetRootGe	CREATE_COMPLETE	
	3/29/2024 12:45:22 PM	AWS::Lambda::Permission	GetRootGetPermissionProd	serverlessPowertoolsStack-GetRootGe	CREATE_IN_PROGRESS	Resour
	3/29/2024 12:45:21 PM	AWS::ApiGateway::Deployment	ServerlessRestApiDeployment9d78fb6c57		CREATE_IN_PROGRESS	
	3/29/2024 12:45:21 PM	AWS::Lambda::Permission	GetRootGetPermissionProd		CREATE_IN_PROGRESS	
	3/29/2024 12:45:21 PM	AWS::ApiGateway::RestApi	ServerlessRestApi	bhntmpmjoj	CREATE_COMPLETE	
	3/29/2024 12:45:20 PM	AWS::ApiGateway::RestApi	ServerlessRestApi	bhntmpmjoj	CREATE_IN_PROGRESS	Resour
	3/29/2024 12:45:19 PM	AWS::ApiGateway::RestApi	ServerlessRestApi		CREATE_IN_PROGRESS	
	3/29/2024 12:45:18 PM	AWS::Lambda::Function	Get	serverlessPowertoolsStack-Get-Lgaks	CREATE_IN_PROGRESS	Eventu
	3/29/2024 12:45:17 PM	AWS::Lambda::Function	Get	serverlessPowertoolsStack-Get-Lgaks	CREATE_IN_PROGRESS	Resour
	3/29/2024 12:45:16 PM	AWS::Lambda::Function	Get		CREATE_IN_PROGRESS	
	3/29/2024 12:45:15 PM	AWS::IAM::Role	GetRole	serverlessPowertoolsStack-GetRole-D	CREATE_COMPLETE	
	3/29/2024 12:44:59 PM	AWS::IAM::Role	GetRole	serverlessPowertoolsStack-GetRole-D	CREATE_IN_PROGRESS	Resour
	3/29/2024 12:44:58 PM	AWS::IAM::Role	GetRole		CREATE_IN_PROGRESS	
	3/29/2024 12:44:55 PM	AWS::CloudFormation::Stack	serverlessPowertoolsStack	arn:aws:cloudformation:us-east-1:508...	CREATE_IN_PROGRESS	User In
	3/29/2024 12:44:49 PM	AWS::CloudFormation::Stack	serverlessPowertoolsStack	arn:aws:cloudformation:us-east-1:508...	REVIEW_IN_PROGRESS	User In

## Uji Aplikasi Tanpa Server

Ketika pembuatan tumpukan selesai, Anda dapat melihat aplikasi Anda menggunakan URL AWS Tanpa Server. Jika Anda telah menyelesaikan tutorial ini tanpa menambahkan fungsi atau parameter tambahan, mengakses URL AWS tanpa server Anda akan menampilkan frasa berikut di browser web Anda: Hello Powertools for AWS Lambda (.NET)

## Tutorial: Membuat Aplikasi Amazon Rekognition Lambda

Tutorial ini menunjukkan cara membuat aplikasi Lambda yang menggunakan Amazon Rekognition untuk menandai objek Amazon S3 dengan label yang terdeteksi.

Untuk prasyarat dan informasi tentang pengaturan AWS Toolkit for Visual Studio, lihat [Menggunakan Template AWS Lambda di Toolkit for Visual Studio](#). AWS

## Buat Proyek Rekognition Gambar Lambda Inti Visual Studio .NET

Prosedur berikut menjelaskan cara membuat aplikasi Amazon Rekognition Lambda dari aplikasi AWS Toolkit for Visual Studio

### Note

Setelah pembuatan, aplikasi Anda memiliki solusi dengan dua proyek: proyek sumber yang berisi kode fungsi Lambda Anda untuk diterapkan ke Lambda, dan proyek pengujian menggunakan XUnit untuk menguji fungsi Anda secara lokal.

Terkadang Visual Studio tidak dapat menemukan semua NuGet referensi untuk proyek Anda. Ini karena cetak biru memerlukan dependensi yang harus diambil dari NuGet. Ketika proyek baru dibuat, Visual Studio hanya menarik referensi lokal dan bukan referensi jarak jauh dari NuGet. Untuk memperbaiki NuGet kesalahan: klik kanan referensi Anda dan pilih Pulihkan Paket.

1. Dari Visual Studio, perluas menu File, perluas Baru, lalu pilih Project.
2. Di kotak dialog Proyek Baru, pastikan bahwa kotak drop-down Bahasa, Platform, dan Jenis Proyek diatur ke "Semua..." dan masukkan **aws lambda** di bidang Pencarian.
3. Pilih template AWS Lambda with Tests (.NET Core - C #).
4. Klik Berikutnya untuk membuka dialog Configure your new project.
5. Dari dialog Konfigurasi proyek baru Anda, masukkan "ImageRekognition" untuk Nama, lalu lengkapi bidang yang tersisa sesuai preferensi Anda. Pilih Buat tombol untuk melanjutkan ke dialog Select Blueprint.
6. Dari dialog Select Blueprint, pilih cetak biru Deteksi Label Gambar, lalu pilih Selesai untuk membuat proyek Visual Studio.

### Note

Cetak biru ini menyediakan kode untuk mendengarkan peristiwa Amazon S3 dan menggunakan Amazon Rekognition untuk mendeteksi label dan menambahkannya ke objek S3 sebagai tag.

## Meninjau File Proyek

Bagian berikut memeriksa file-file proyek ini:

1. `Function.cs`
2. `aws-lambda-tools-defaults.json`

### 1. `Function.cs`

Di dalam `Function.cs` file, segmen kode pertama adalah atribut assembly, yang terletak di bagian atas file. Secara default, Lambda hanya menerima parameter input dan mengembalikan tipe `System.IO.Stream`. Anda harus mendaftarkan serializer untuk menggunakan kelas yang diketik untuk parameter input dan tipe pengembalian. Atribut assembly mendaftarkan serializer Lambda JSON, yang `Newtonsoft.Json` digunakan untuk mengonversi aliran ke kelas yang diketik. Anda dapat mengatur serializer di tingkat perakitan atau metode.

Berikut ini adalah contoh atribut assembly:

```
// Assembly attribute to enable the Lambda function's JSON input to be converted into
// a .NET class.
[assembly:
    LambdaSerializer(typeof(Amazon.Lambda.Serialization.SystemTextJson.DefaultLambdaJsonSerializer))
```

Kelas ini memiliki dua konstruktor. Yang pertama adalah konstruktor default yang digunakan saat Lambda memanggil fungsi Anda. Konstruktor ini menciptakan klien layanan Amazon S3 dan Amazon Rekognition. Konstruktor juga mengambil AWS kredensi untuk klien ini dari peran IAM yang Anda tetapkan ke fungsi saat Anda menerapkannya. AWS Wilayah untuk klien diatur ke wilayah yang menjalankan fungsi Lambda Anda. Dalam cetak biru ini, Anda hanya ingin menambahkan tag ke objek Amazon S3 jika layanan Amazon Rekognition memiliki tingkat kepercayaan minimum tentang label tersebut. Konstruktor ini memeriksa variabel lingkungan `MinConfidence` untuk menentukan tingkat kepercayaan yang dapat diterima. Anda dapat mengatur variabel lingkungan ini saat Anda menerapkan fungsi Lambda.

Berikut ini adalah contoh konstruktor kelas pertama di `Function.cs`:

```
public Function()
{
    this.S3Client = new AmazonS3Client();
    this.RekognitionClient = new AmazonRekognitionClient();
}
```

```

var environmentMinConfidence =
System.Environment.GetEnvironmentVariable(MIN_CONFIDENCE_ENVIRONMENT_VARIABLE_NAME);
if(!string.IsNullOrEmpty(environmentMinConfidence))
{
    float value;
    if(float.TryParse(environmentMinConfidence, out value))
    {
        this.MinConfidence = value;
        Console.WriteLine($"Setting minimum confidence to {this.MinConfidence}");
    }
    else
    {
        Console.WriteLine($"Failed to parse value {environmentMinConfidence} for
minimum confidence. Reverting back to default of {this.MinConfidence}");
    }
}
else
{
    Console.WriteLine($"Using default minimum confidence of {this.MinConfidence}");
}
}

```

Contoh berikut menunjukkan bagaimana konstruktor kedua dapat digunakan untuk pengujian. Proyek pengujian mengonfigurasi klien S3 dan Rekognition sendiri dan meneruskannya di:

```

public Function(IAmazonS3 s3Client, IAmazonRekognition rekognitionClient, float
minConfidence)
{
    this.S3Client = s3Client;
    this.RekognitionClient = rekognitionClient;
    this.MinConfidence = minConfidence;
}

```

Berikut ini adalah contoh `FunctionHandler` metode di dalam `Function.cs` file.

```

public async Task FunctionHandler(S3Event input, ILambdaContext context)
{
    foreach(var record in input.Records)
    {
        if(!SupportedImageTypes.Contains(Path.GetExtension(record.S3.Object.Key)))
        {
            Console.WriteLine($"Object {record.S3.Bucket.Name}:{record.S3.Object.Key}
is not a supported image type");
        }
    }
}

```

```
        continue;
    }

    Console.WriteLine($"Looking for labels in image {record.S3.Bucket.Name}:
{record.S3.Object.Key}");
    var detectResponses = await this.RekognitionClient.DetectLabelsAsync(new
DetectLabelsRequest
    {
        MinConfidence = MinConfidence,
        Image = new Image
        {
            S3Object = new Amazon.Rekognition.Model.S3Object
            {
                Bucket = record.S3.Bucket.Name,
                Name = record.S3.Object.Key
            }
        }
    });

    var tags = new List();
    foreach(var label in detectResponses.Labels)
    {
        if(tags.Count < 10)
        {
            Console.WriteLine($"\\tFound Label {label.Name} with confidence
{label.Confidence}");
            tags.Add(new Tag { Key = label.Name, Value =
label.Confidence.ToString() });
        }
        else
        {
            Console.WriteLine($"\\tSkipped label {label.Name} with confidence
{label.Confidence} because maximum number of tags reached");
        }
    }

    await this.S3Client.PutObjectTaggingAsync(new PutObjectTaggingRequest
    {
        BucketName = record.S3.Bucket.Name,
        Key = record.S3.Object.Key,
        Tagging = new Tagging
        {
            TagSet = tags
        }
    });
}
```

```
    });  
  }  
  return;  
}
```

`FunctionHandler` adalah metode yang dipanggil Lambda setelah membangun instance. Perhatikan bahwa parameter input adalah tipe `S3Event` dan bukan `aStream`. Anda dapat melakukan ini karena serializer Lambda JSON yang terdaftar. `S3Event` berisi semua informasi tentang acara yang dipicu di Amazon S3. Fungsi loop melalui semua objek S3 yang merupakan bagian dari acara dan memberitahu Rekognition untuk mendeteksi label. Setelah label terdeteksi, mereka ditambahkan sebagai tag ke objek S3.

### Note

Kode berisi panggilan ke `Console.WriteLine()`. Saat fungsi berjalan di Lambda, semua panggilan untuk `Console.WriteLine()` mengalihkan ke Amazon Logs. CloudWatch

## 2. aws-lambda-tools-defaults.json

`aws-lambda-tools-defaults.json` berisi nilai default yang telah ditetapkan cetak biru untuk mengisi beberapa bidang di wizard penerapan. Ini juga membantu dalam mengatur opsi baris perintah untuk integrasi dengan .NET Core CLI.

Untuk mengakses integrasi .NET Core CLI, navigasikan ke direktori dan ketik proyek fungsi. **dotnet lambda help**

### Note

Penangan fungsi menunjukkan metode apa yang Lambda panggil sebagai respons terhadap fungsi yang dipanggil. Format bidang ini adalah: `<assembly-name>::<full-type-name>::<method-name>`. Namespace harus disertakan dengan nama tipe.

## Menyebarkan Fungsi

Prosedur berikut menjelaskan cara menerapkan fungsi Lambda Anda.

1. Dari Solution Explorer, klik kanan proyek Lambda dan pilih Publish to AWS Lambda untuk membuka jendela Upload to. AWS Lambda



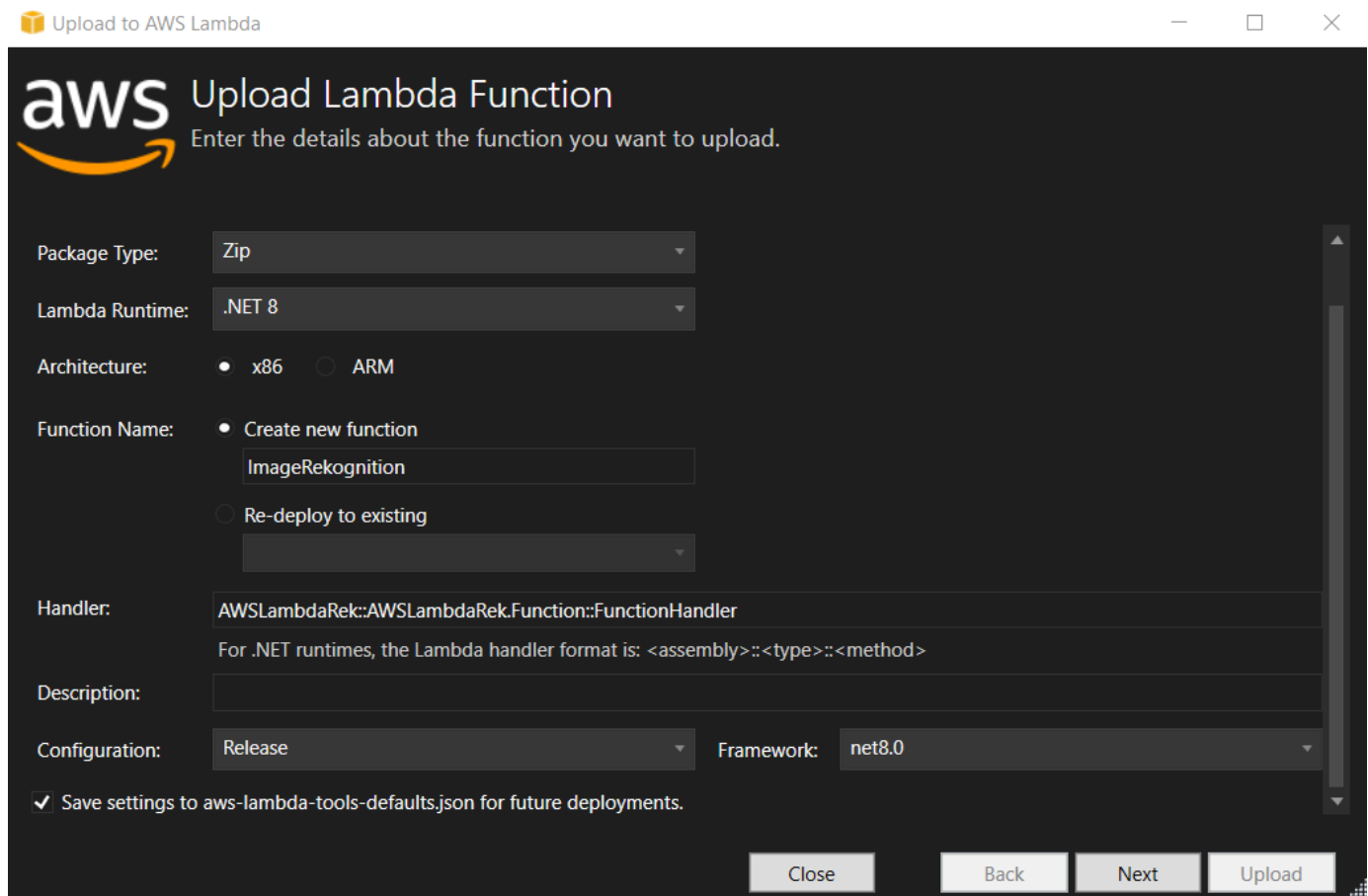
**Note**

Nilai preset diambil dari file. `aws-lambda-tools-defaults.json`

2. Dari AWS Lambda jendela Upload to, masukkan nama ke bidang Function Name, lalu pilih tombol Next untuk maju ke jendela Advanced Function Details.

**Note**

Contoh ini, menggunakan Nama Fungsi **ImageRekognition**.

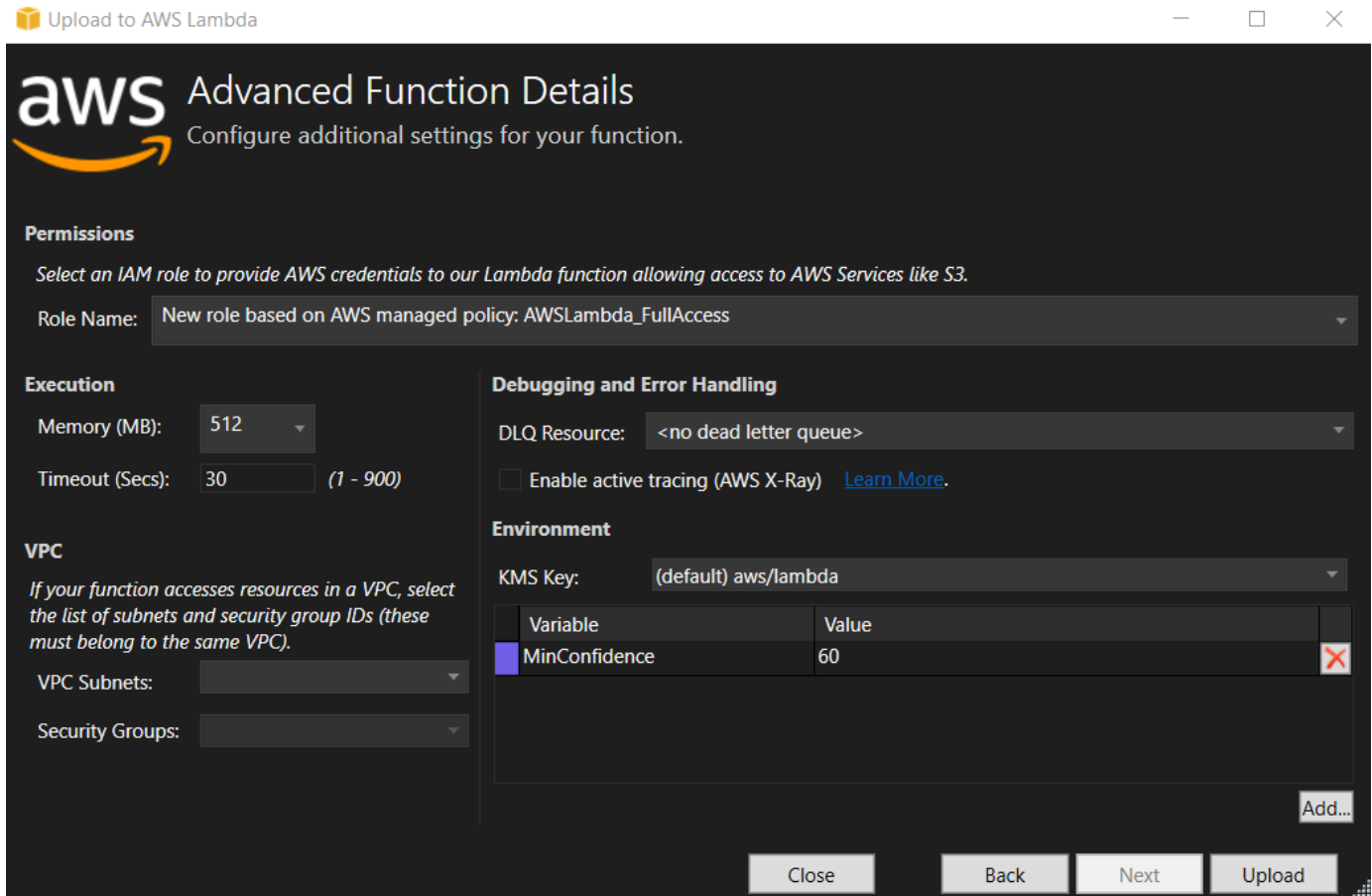


3. Dari jendela Detail Fungsi Lanjutan, pilih peran IAM yang memberikan izin bagi kode Anda untuk mengakses sumber daya Amazon S3 dan Amazon Rekognition Anda.

**Note**

Jika Anda mengikuti contoh ini, pilih `AWSLambda_FullAccess` peran.

4. Setel variabel lingkungan `MinConfidence` ke 60, lalu pilih Unggah untuk meluncurkan proses penerapan. Proses penerbitan selesai ketika tampilan Fungsi ditampilkan di AWS Explorer.



5. Setelah penerapan berhasil, konfigurasi Amazon S3 untuk mengirim peristiwanya ke fungsi baru Anda dengan menavigasi ke tab Sumber Peristiwa.
6. Dari tab Sumber Acara, pilih tombol Tambah, lalu pilih bucket Amazon S3 untuk terhubung dengan fungsi Lambda Anda.

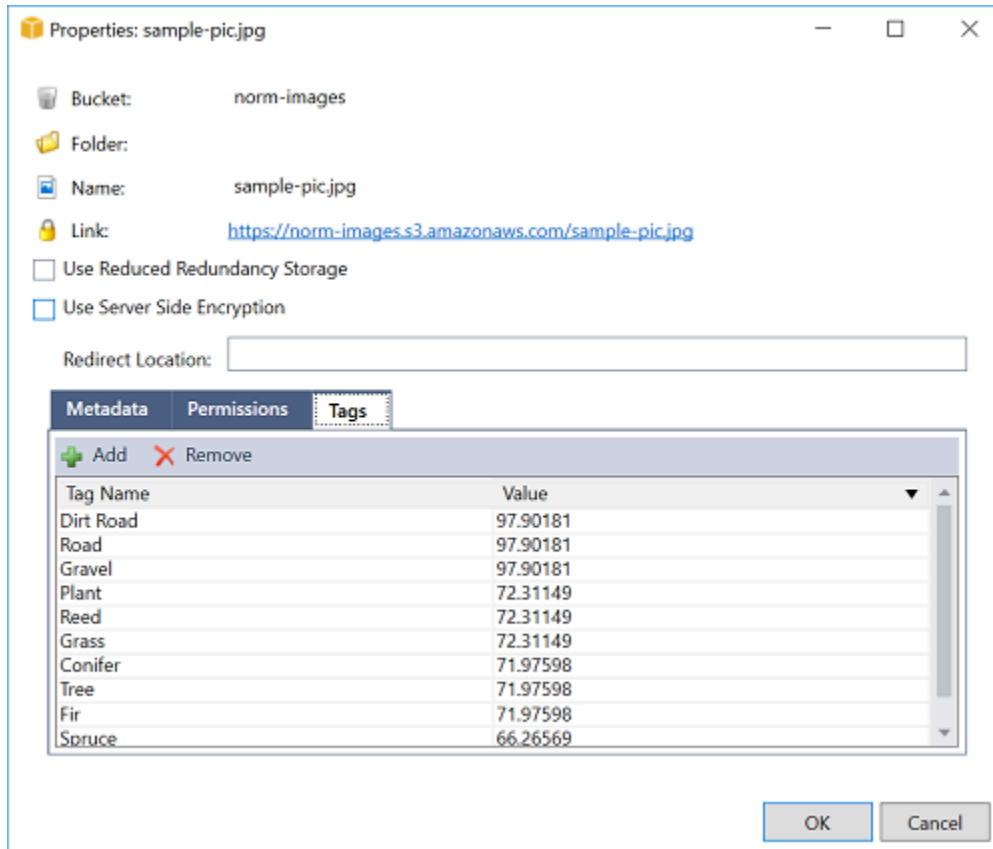
**Note**

Bucket harus berada di AWS wilayah yang sama dengan fungsi Lambda Anda.

## Uji Fungsinya

Sekarang setelah fungsi tersebut diterapkan dan bucket S3 dikonfigurasi sebagai sumber acara untuknya, buka browser bucket S3 dari AWS Explorer untuk bucket yang Anda pilih. Kemudian unggah beberapa gambar.

Ketika unggahan selesai, Anda dapat mengonfirmasi bahwa fungsi Anda berjalan dengan melihat log dari tampilan fungsi Anda. Atau, klik kanan gambar di browser bucket dan pilih Properties. Pada tab Tag, Anda dapat melihat tag yang diterapkan ke objek Anda.



## Tutorial: Menggunakan Amazon Logging Frameworks dengan AWS Lambda untuk Membuat Log Aplikasi

Anda dapat menggunakan Amazon CloudWatch Logs untuk memantau, menyimpan, dan mengakses log aplikasi Anda. Untuk mendapatkan data CloudWatch log ke Log, gunakan AWS SDK atau instal agen CloudWatch Log untuk memantau folder log tertentu. CloudWatch Log terintegrasi dengan beberapa framework logging .NET yang populer, menyederhanakan alur kerja.

Untuk mulai bekerja dengan kerangka kerja CloudWatch logging Log dan .NET, tambahkan NuGet paket yang sesuai dan sumber keluaran CloudWatch Log ke aplikasi Anda, lalu gunakan

pustaka logging Anda seperti biasa. Ini memungkinkan aplikasi Anda untuk mencatat pesan dengan framework .NET Anda, mengirimkannya ke CloudWatch Log, menampilkan pesan log aplikasi Anda di konsol CloudWatch Log. Anda juga dapat mengatur metrik dan alarm dari konsol CloudWatch Log, berdasarkan pesan log aplikasi Anda.

Kerangka kerja logging.NET yang didukung meliputi:

- nLog: Untuk melihat, lihat paket nLog [nuget.org](http://nuget.org).
- Log4net: Untuk melihat, lihat paket Log4net [nuget.org](http://nuget.org).
- ASP.NET Core logging Framework: Untuk melihat, lihat paket [nuget.org](http://nuget.org) [ASP.NET Core logging Framework](https://www.nuget.org/packages/Microsoft.Extensions.Logging).

Berikut ini adalah contoh NLog.config file yang memungkinkan CloudWatch Log dan konsol sebagai output untuk pesan log dengan menambahkan AWS.Logger.NLog NuGet paket, dan AWS target ke dalam NLog.config.

```
<?xml version="1.0" encoding="utf-8" ?>
<nlog xmlns="http://www.nlog-project.org/schemas/NLog.xsd"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      throwExceptions="true">
  <targets>
    <target name="aws" type="AWSTarget" logGroup="NLog.ConfigExample" region="us-east-1"/>
    <target name="logfile" xsi:type="Console" layout="${callsite} ${message}" />
  </targets>
  <rules>
    <logger name="*" minlevel="Info" writeTo="logfile,aws" />
  </rules>
</nlog>
```

Plugin logging semuanya dibangun di atas AWS SDK for .NET dan mengautentikasi AWS kredensial Anda dalam proses yang mirip dengan SDK. Contoh berikut merinci izin yang diperlukan oleh kredensi plugin logging untuk mengakses Log: CloudWatch

#### Note

Plugin AWS logging.NET adalah proyek open source. Untuk informasi tambahan, sampel, dan instruksi, lihat topik [sampel](#) dan [instruksi](#) di [GitHub repositori.NET AWS Logging](#).

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "logs:CreateLogGroup",
        "logs:CreateLogStream",
        "logs:PutLogEvents",
        "logs:DescribeLogGroups"
      ],
      "Resource": [
        "arn:aws:logs:*:*:*"
      ]
    }
  ]
}
```

# Menyebarkan keAWS

Toolkit for Visual Studio mendukung Deployment aplikasi untukAWS Elastic Beanstalk kontainer atauAWS CloudFormation tumpukan.

## Note

Jika Anda menggunakan Visual Studio Express Edition:

- Anda dapat menggunakan [Docker CLI](#) untuk menerapkan aplikasi ke kontainer Amazon ECS.
- Anda dapat menggunakan [AWSManagement Console](#) untuk menyebarkan aplikasi ke container Elastic Beanstalk.

Untuk penyebaran Elastic Beanstalk, Anda harus terlebih dahulu membuat paket penyebaran web. Untuk informasi lebih lanjut, kunjungi [Cara: Membuat Package Deployment Web dalam Visual Studio](#). Untuk penyebaran Amazon ECS, Anda harus memiliki image Docker. Untuk informasi selengkapnya, lihat [Alat Visual Studio untuk Docker](#).

## Topik

- [Bekerja dengan PublikasikanAWSdi Visual](#)
- [Men-deployAWS LambdaProyek dengan .NET Core CLI](#)
- [Menerapkan ke Elastic Beanstalk](#)
- [Menyebarkan ke Amazon EC2 Container Service](#)

## Bekerja dengan PublikasikanAWSdi Visual

PublikasikanAWSadalah pengalaman penyebaran interaktif yang membantu Anda menerbitkan aplikasi .NET Anda keAWSTarget deployment, aplikasi pendukung yang menargetkan .NET Core 3.1 dan yang lebih baru. Bekerja dengan PublikasikanAWSmenjaga alur kerja Anda di dalam Visual Studio dengan membuat fitur penyebaran ini tersedia, langsung dari IDE Anda:

- Kemampuan untuk menyebarkan aplikasi Anda dengan satu klik.
- Rekomendasi penerapan berdasarkan aplikasi Anda.

- Pembuatan Dockerfile otomatis, sebagaimana relevan dan diperlukan oleh lingkungan tujuan penyebaran Anda (target penyebaran).
- Pengaturan yang dioptimalkan untuk membangun dan mengemas aplikasi Anda, seperti yang dipersyaratkan oleh target penyebaran Anda.

#### Note

Untuk informasi tambahan tentang penerbitan aplikasi .NET Framework, lihat panduan [Membuat dan men-deploy aplikasi .NET di Elastic Beanstalk](#). Anda juga dapat mengakses Publikasikan keAWS dari CLI .NET. Untuk informasi lebih lanjut, lihat [Menyebarkan aplikasi .NET padaAWS](#) Panduan

#### Topik

- [Prasyarat](#)
- [Tipe aplikasi yang didukung](#)
- [Publikasikan aplikasiAWSsasaran](#)

## Prasyarat

Untuk berhasil mempublikasikan aplikasi .NET keAWS layanan, instal yang berikut ini ke perangkat lokal Anda:

- .NET Core 3.1+ (yang meliputi .NET5 dan .NET6): Untuk informasi tambahan tentang produk ini dan mengunduh informasi, kunjungi [Situs unduhan Microsoft](#).
- Node.js 14.x atau versi yang lebih baru: Node.js diperlukan untuk menjalankan AWS Cloud Development Kit (AWS CDK). Untuk mengunduh atau mendapatkan informasi lebih lanjut tentang Node.js, kunjungi [Situs unduhan Node.js](#).

#### Note

PublikasikanAWS memanfaatkanAWS CDK untuk menyebarkan aplikasi Anda dan semua infrastruktur penyebarannya sebagai satu proyek. Untuk informasi lebih lanjut tentangAWS CDK lihat [Cloud Development Kit](#) Panduan

- (Opsional) Docker digunakan saat menerapkan ke layanan berbasis kontainer seperti Amazon ECS. Untuk informasi lebih lanjut dan untuk mengunduh Docker, lihat [Unduh Docker](#) Situs

## Tipe aplikasi yang didukung

Sebelum menerbitkan ke target baru atau keluar, mulailah dengan membuat atau membuka salah satu jenis proyek berikut di Visual Studio:

- Aplikasi ASP.NET
- Aplikasi Konsol .NET
- Blazor WebAssembly penerapan

## Publikasikan aplikasi AWS sasaran

Saat menerbitkan ke target baru, Publikasikan ke AWS akan memandu Anda melalui proses dengan membuat rekomendasi dan menggunakan pengaturan umum. Jika Anda perlu mempublikasikan ke target yang telah diatur sebelumnya, preferensi Anda disimpan dan dapat disesuaikan, atau segera tersedia untuk penyebaran sekali klik.

### Publikasikan ke target baru

Berikut ini menjelaskan cara mengkonfigurasi Publikasikan ke AWS preferensi penyebaran, saat Anda memublikasikan ke target baru.

1. Dari AWS Penjelajah, memperluas Kredensial menu drop-down, lalu pilih AWS profil yang sesuai dengan wilayah dan AWS layanan yang diperlukan untuk penyebaran Anda.
2. Perluas Wilayah menu drop-down, lalu pilih AWS wilayah yang berisi AWS layanan yang diperlukan untuk penyebaran Anda.
3. Dari Studio Solusi panel, membuka menu konteks untuk (klik kanan) nama proyek, dan pilih Publikasikan AWS. Ini akan membuka Publikasikan AWS.
4. From Publikasikan AWS, pilih Publikasikan ke Target Baru untuk mengkonfigurasi penyebaran baru.

#### Note

Untuk mengubah kredensi penerapan default Anda, pilih atau klik edit link yang terletak di sebelah Kredensial Bagian Publikasikan AWS.



Untuk melewati proses konfigurasi target, pilih **Publikasikan ke Target yang Ada**, lalu pilih konfigurasi pilihan Anda dari daftar target penyebaran Anda sebelumnya.

5. Dari **Publikasikan panel**, pilih **AWS Layanan** untuk mengelola penyebaran aplikasi Anda.
6. Saat Anda puas dengan konfigurasi Anda, pilih **Publikasikan** untuk memulai proses deployment

#### Note

Setelah memulai penyebaran, **Publikasikan AWS** menampilkan pembaruan status berikut:

- Selama proses deployment **Publikasikan AWS** menampilkan informasi tentang kemajuan penyebaran.
- Mengikuti proses deployment **Publikasikan AWS** menunjukkan apakah penyebaran berhasil atau gagal.
- Setelah penyebaran berhasil, **Sumber daya panel** menawarkan informasi tambahan tentang sumber daya yang dibuat. Informasi ini akan bervariasi tergantung pada jenis aplikasi dan konfigurasi penyebaran.

## Publikasikan ke target yang ada

Berikut ini menjelaskan cara mempublikasikan ulang aplikasi .NET Anda ke yang sudah ada **AWS Target**

1. Dari **AWS Penjelajah**, memperluas **Kredensial** menu drop-down, lalu pilih **AWS profil** yang sesuai dengan wilayah dan **AWS Layanan** yang diperlukan untuk penyebaran Anda.
2. Perluas **Wilayah** menu drop-down, lalu pilih **AWS wilayah** yang berisi **AWS Layanan** yang diperlukan untuk penyebaran Anda.
3. Dari **Studio Solusi panel**, klik kanan nama proyek dan pilih **Publikasikan AWS** membuka **Publikasikan AWS**.
4. From **Publikasikan AWS**, pilih **Publikasikan ke Target yang Ada** untuk memilih lingkungan deployment Anda dari daftar target yang sudah ada.

**Note**

Jika Anda baru saja menerbitkan aplikasi apa pun keAWS Cloud, aplikasi tersebut ditampilkan di Publikasikan keAWS.

5. Pilih target publikasi tempat Anda ingin men-deploy aplikasi Anda, lalu klik **Publikasikan** untuk memulai proses deployment

## Men-deploy AWS Lambda Proyek dengan .NET Core CLI

Parameter AWS Toolkit for Visual Studio termasuk AWS Lambda .NET Core template for Visual Studio. Anda dapat men-deploy fungsi Lambda dibangun di Visual Studio menggunakan antarmuka .NET Core baris perintah (CLI).

### Topik

- [Prasyarat](#)
- [Topik terkait](#)
- [Daftar Perintah Lambda Tersedia melalui .NET Core CLI](#)
- [Menerapkan .NET Core Lambda Project dari .NET Core CLI](#)

## Prasyarat

Sebelum bekerja dengan .NET Core CLI untuk men-deploy fungsi Lambda, Anda harus memenuhi prasyarat berikut:

- Pastikan Visual Studio 2015 Update 3 diinstal.
- Pasang [.NET Core untuk Windows](#).
- Mengatur .NET Core CLI untuk bekerja dengan Lambda. Untuk informasi selengkapnya, lihat [.NET Core CLI](#) di dalam [AWS Lambda Panduan Pengembang](#).
- Menginstal Toolkit for Visual Studio. Untuk informasi selengkapnya, lihat [Memasang AWS Toolkit for Visual Studio](#).

## Topik terkait

Topik terkait berikut dapat membantu saat Anda menggunakan CLI Inti .NET untuk menerapkan fungsi Lambda:

- Untuk informasi selengkapnya tentang fungsi Lambda, lihat [ApaAWSLambda?](#) di dalam [AWS Lambda Panduan Pengembang](#).
- Untuk informasi selengkapnya tentang cara membuat fungsi Lambda di Visual Studio, lihat [AWS Lambda](#).
- Untuk informasi selengkapnya tentang Microsoft .NET Core, lihat [.NET Core](#) dalam dokumentasi online Microsoft.

## Daftar Perintah Lambda Tersedia melalui .NET Core CLI

Untuk mencantumkan perintah Lambda yang tersedia melalui CLI inti .NET, lakukan hal berikut.

1. Buka jendela prompt perintah, dan arahkan ke folder yang berisi proyek Visual Studio .NET Core Lambda.
2. Masukkan `dotnet lambda --help`.

```
C:\Lambda\AWSLambda1\AWSLambda1>dotnet lambda --help
AWS Lambda Tools for .NET Core functions
Project Home: https://github.com/aws/aws-lambda-dotnet
.
Commands to deploy and manage Lambda functions:
.
    deploy-function      Deploy the project to Lambda
    invoke-function      Invoke the function in Lambda with an optional
input
    list-functions       List all of your Lambda functions
    delete-function      Delete a Lambda function
    get-function-config  Get the current runtime configuration for a Lambda
function
    update-function-config Update the runtime configuration for a Lambda
function
.
Commands to deploy and manage AWS serverless applications using AWS CloudFormation:
.
    deploy-serverless    Deploy an AWS serverless application
```

```
list-serverless      List all of your AWS serverless applications
delete-serverless   Delete an AWS serverless application
.
Other Commands:
.
package             Package a Lambda project into a .zip file ready for
deployment
.
To get help on individual commands, run the following:

dotnet lambda help <command>
```

## Menerapkan .NET Core Lambda Project dari .NET Core CLI

Petunjuk berikut menganggap Anda telah membuat AWS Lambda Fungsi .NET Core di Visual Studio.

1. Buka jendela prompt perintah, dan arahkan ke folder yang berisi proyek Visual Studio .NET Core Lambda Anda.
2. Masukkan `dotnet lambda deploy-function`.
3. Saat diminta, masukkan nama fungsi untuk men-deploy. Ini bisa menjadi nama baru atau nama fungsi yang ada.
4. Saat diminta, masukkan AWS Wilayah (Wilayah tempat fungsi Lambda Anda akan digunakan).
5. Saat diminta, pilih atau buat peran IAM yang akan diasumsikan Lambda saat menjalankan fungsi.

Pada penyelesaian yang berhasil, pesan Fungsi Lambda baru dibuat ditampilkan.

```
C:\Lambda\AWSLambda1\AWSLambda1>dotnet lambda deploy-function
Executing publish command
... invoking 'dotnet publish', working folder 'C:\Lambda\AWSLambda1\AWSLambda1\bin
\Release\netcoreapp1.0\publish'
... publish: Publishing AWSLambda1 for .NETCoreApp,Version=v1.0
... publish: Project AWSLambda1 (.NETCoreApp,Version=v1.0) will be compiled because
expected outputs are missing
... publish: Compiling AWSLambda1 for .NETCoreApp,Version=v1.0
... publish: Compilation succeeded.
... publish:      0 Warning(s)
... publish:      0 Error(s)
... publish: Time elapsed 00:00:01.2479713
... publish:
```

```
... publish: publish: Published to C:\Lambda\AWSLambda1\AWSLambda1\bin\Release\netcoreapp1.0\publish
... publish: Published 1/1 projects successfully
Zipping publish folder C:\Lambda\AWSLambda1\AWSLambda1\bin\Release\netcoreapp1.0\publish to C:\Lambda\AWSLambda1\AWSLambda1\bin\Release\netcoreapp1.0\AWSLambda1.zip
Enter Function Name: (AWS Lambda function name)
DotNetCoreLambdaTest
Enter AWS Region: (The region to connect to AWS services)
us-west-2
Creating new Lambda function
Select IAM Role that Lambda will assume when executing function:
    1) lambda_exec_LambdaCoreFunction
    2) *** Create new IAM Role ***
1
New Lambda function created
```

Jika Anda menerapkan fungsi yang ada, fungsi deploy hanya meminta AWS Wilayah.

```
C:\Lambda\AWSLambda1\AWSLambda1>dotnet lambda deploy-function
Executing publish command
Deleted previous publish folder
... invoking 'dotnet publish', working folder 'C:\Lambda\AWSLambda1\AWSLambda1\bin\Release\netcoreapp1.0\publish'
... publish: Publishing AWSLambda1 for .NETCoreApp,Version=v1.0
... publish: Project AWSLambda1 (.NETCoreApp,Version=v1.0) was previously compiled.
Skipping compilation.
... publish: publish: Published to C:\Lambda\AWSLambda1\AWSLambda1\bin\Release\netcoreapp1.0\publish
... publish: Published 1/1 projects successfully
Zipping publish folder C:\Lambda\AWSLambda1\AWSLambda1\bin\Release\netcoreapp1.0\publish to C:\Lambda\AWSLambda1\AWSLambda1\bin\Release\netcoreapp1.0\AWSLambda1.zip
Enter Function Name: (AWS Lambda function name)
DotNetCoreLambdaTest
Enter AWS Region: (The region to connect to AWS services)
us-west-2
Updating code for existing function
```

Setelah fungsi Lambda Anda diterapkan, fungsi tersebut siap digunakan. Untuk informasi selengkapnya, lihat [Contoh Cara Menggunakan AWS Lambda](#).

Lambda secara otomatis memonitor fungsi Lambda untuk Anda dan melaporkan metrik melalui Amazon CloudWatch. Untuk memantau dan memecahkan masalah fungsi Lambda Anda, lihat [Pemecahan Masalah dan Pemantauan AWS Fungsi Lambda dengan Amazon CloudWatch](#).

## Menerapkan ke Elastic Beanstalk

AWS Elastic Beanstalk adalah layanan yang menyederhanakan proses penyediaan AWS sumber daya untuk aplikasi Anda. Elastic Beanstalk menyediakan semua AWS infrastruktur yang diperlukan untuk men-deploy aplikasi Anda. Infrastruktur ini meliputi:

- Instans Amazon EC2 yang menghosting executable dan konten untuk aplikasi Anda.
- Grup Auto Scaling untuk mempertahankan jumlah instans Amazon EC2 yang sesuai untuk mendukung aplikasi Anda.
- Penyeimbang beban Elastic Load Balancing yang merutekan lalu lintas masuk ke instans Amazon EC2 dengan bandwidth terbanyak.

Toolkit for Visual Studio menyediakan wizard yang menyederhanakan penerbitan aplikasi melalui Elastic Beanstalk. Wizard ini dijelaskan di bagian berikut.

Untuk informasi selengkapnya tentang Elastic Beanstalk, buka [Dokumentasi Elastic Beanstalk](#).


### Topik

- [Menyebarkan Aplikasi ASP.NET Tradisional ke Elastic Beanstalk](#)
- [Men-deploy aplikasi ASP.NET Core ke Elastic Beanstalk \(Legacy\)](#)
- [Cara Menentukan AWS Kredensial Keamanan untuk Aplikasi Anda](#)
- [Cara Menerbitkan Ulang Aplikasi Anda ke Lingkungan Elastic Beanstalk \(Legacy\)](#)
- [Penyebaran Aplikasi Elastic Beanstalk Kustom](#)
- [Khusus ASP.NET Core Elastic Beanstalk deployment](#)
- [Support Beberapa Aplikasi untuk NET dan Elastic Beanstalk](#)


## Menyebarkan Aplikasi ASP.NET Tradisional ke Elastic Beanstalk

Bagian ini menjelaskan cara menggunakan wizard Publish to Elastic Beanstalk, yang disediakan sebagai bagian dari Toolkit for Visual Studio, untuk menyebarkan aplikasi melalui Elastic Beanstalk.

Untuk berlatih, Anda dapat menggunakan instance dari proyek starter aplikasi web yang dibangun ke Visual Studio atau Anda dapat menggunakan proyek Anda sendiri.

 Note

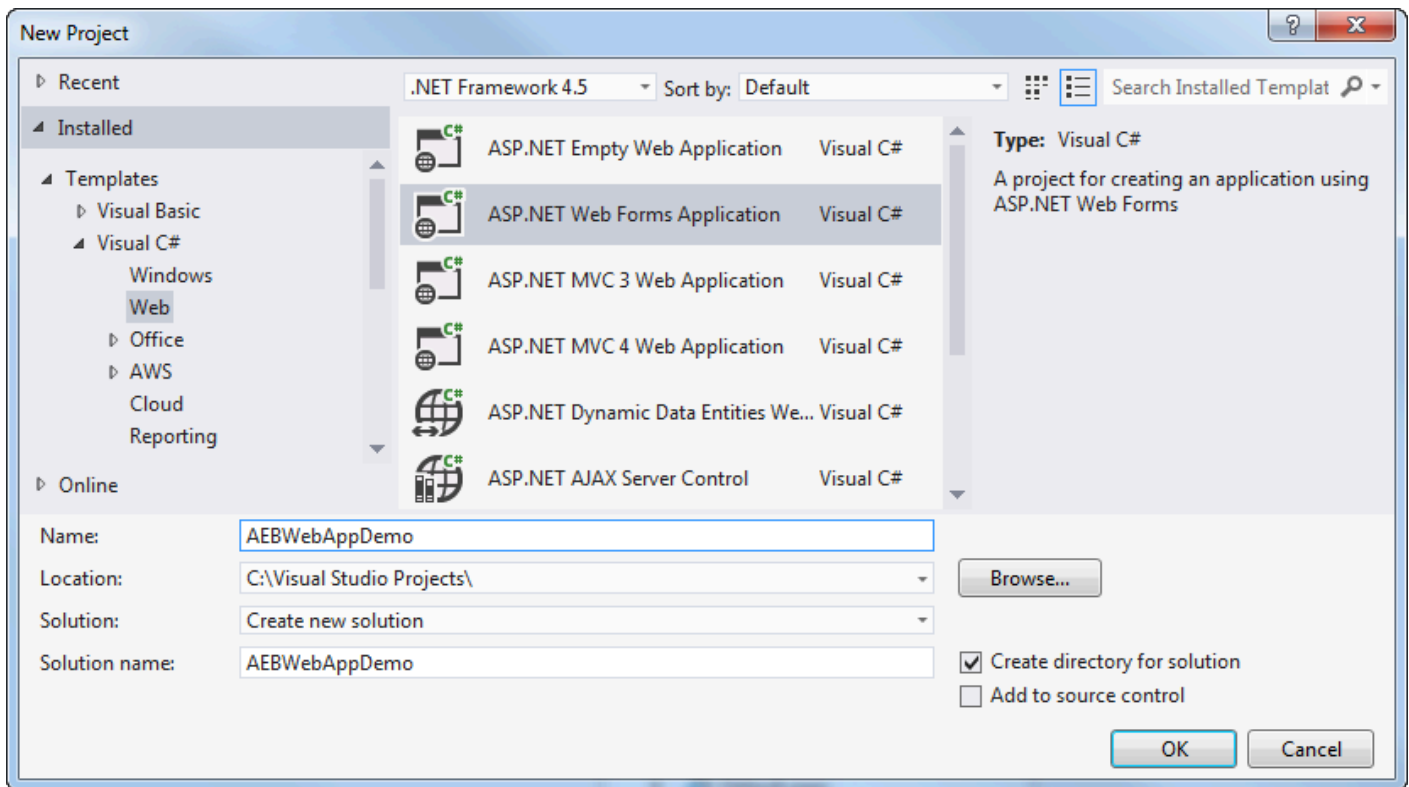
Wizard juga mendukung penerapan aplikasi ASP.NET Core. Untuk informasi tentang ASP.NET Core, lihat panduan [alat penyebaranAWS .NET](#) dan [Deploying yang diperbarui keAWS](#) daftar isi.

 Note

Sebelum Anda dapat menggunakan wizard Publish to Elastic Beanstalk, Anda harus mengunduh dan menginstal [Web Deploy](#). Wizard bergantung pada Web Deploy untuk menyebarkan aplikasi web dan situs web ke server web Internet Information Services (IIS).

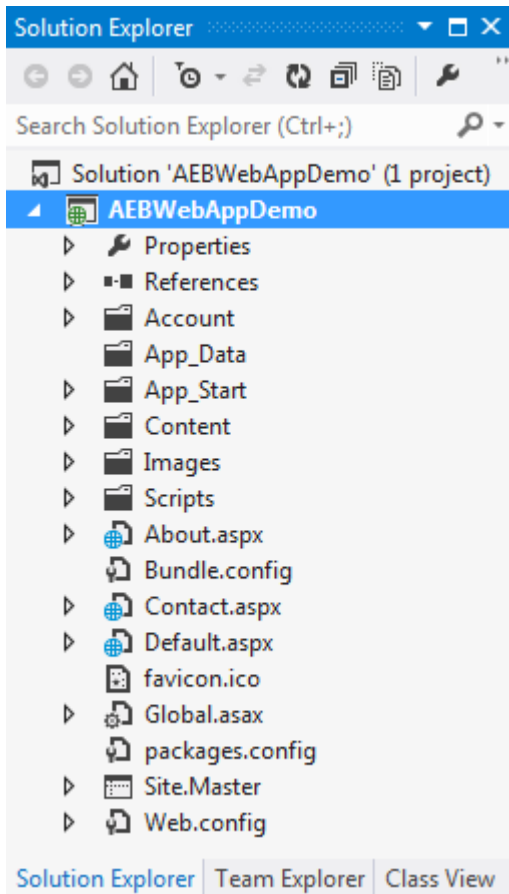
Untuk membuat contoh proyek starter aplikasi web

1. Di Visual Studio, dari menu File, pilih New, dan kemudian pilih Project.
2. Di panel navigasi kotak dialog New Project, luaskan Installed, expand Templates, expand Visual C#, dan kemudian pilih Web.
3. Dalam daftar template proyek web, pilih template apa pun yang berisi kata-kataWeb danApplication dalam deskripsinya. Untuk contoh ini, pilih Aplikasi Formulir Web ASP.NET.



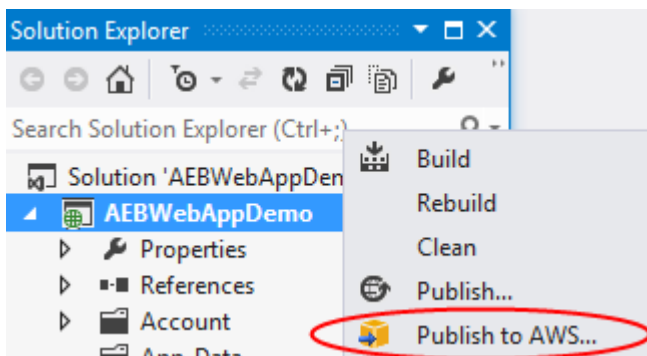
4. Di kotak Nama, ketik AEBWebAppDemo.
5. Di kotak Lokasi, ketik jalur ke folder solusi di mesin pengembangan Anda atau pilih Jelajahi, lalu telusuri dan pilih folder solusi, lalu pilih Pilih Folder.
6. Konfirmasikan kotak Buat direktori untuk solusi dipilih. Dalam daftar drop-down Solusi, konfirmasi Buat solusi baru dipilih, dan kemudian pilih OK. Visual Studio akan membuat solusi dan proyek berdasarkan template proyek ASP.NET Formulir Web Aplikasi. Visual Studio kemudian akan menampilkan Solution Explorer di mana solusi baru dan proyek muncul.



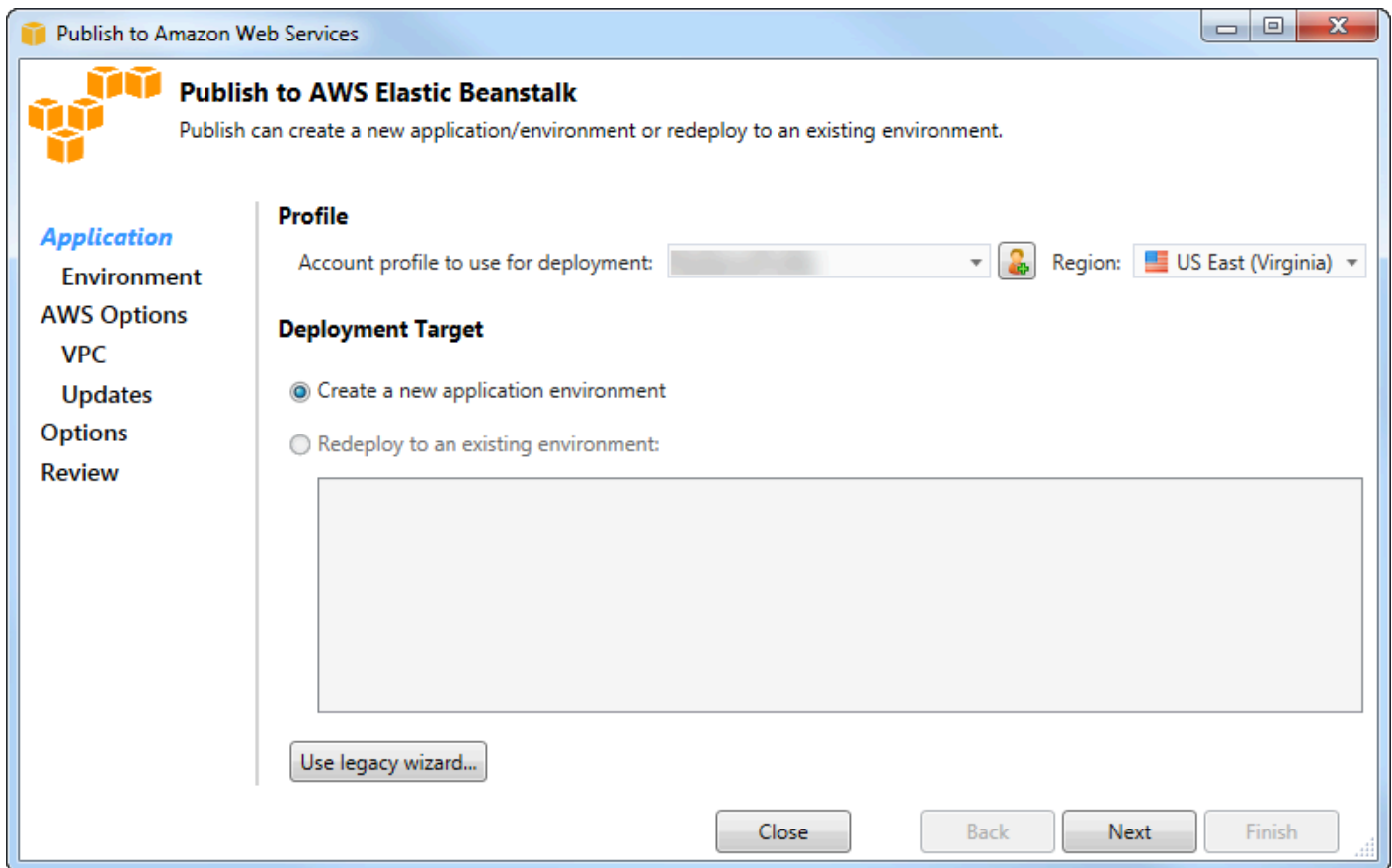


Untuk menyebarkan aplikasi dengan menggunakan wizard Publish to Elastic Beanstalk

1. Di Solution Explorer, buka menu konteks (klik kanan) untuk folder WebAppDemo proyek AEB untuk proyek yang Anda buat di bagian sebelumnya, atau buka menu konteks untuk folder proyek untuk aplikasi Anda sendiri, dan pilih Publish to AWS Elastic Beanstalk.



Wisaya Publish to Elastic Beanstalk Kacang muncul.



2. Di Profil, dari profil Akun yang akan digunakan untuk daftar drop-down penyebaran, pilih profil AWS akun yang ingin Anda gunakan untuk penyebaran.

Secara opsional, jika Anda memiliki AWS akun yang ingin Anda gunakan, tetapi Anda belum membuat profil AWS akun untuk itu, Anda dapat memilih tombol dengan simbol plus (+) untuk menambahkan profil AWS akun.

3. Dari daftar drop-down Region, pilih wilayah yang Anda inginkan Elastic Beanstalk untuk menyebarkan aplikasi.
4. Dalam Deployment Target, Anda dapat memilih salah satu Buat lingkungan aplikasi baru untuk melakukan penyebaran awal aplikasi atau Redeploy ke lingkungan yang ada untuk men-deploy aplikasi yang sebelumnya diterapkan. (Penyebaran sebelumnya mungkin telah dilakukan dengan wizard atau Standalone Deployment Tool yang tidak digunakan lagi.) Jika Anda memilih Redeploy ke lingkungan yang ada, mungkin ada penundaan saat wizard mengambil informasi dari penyebaran sebelumnya yang sedang berjalan.

**Note**

Jika Anda memilih Redeploy ke lingkungan yang ada, pilih lingkungan dalam daftar, dan kemudian pilih Berikutnya, wizard akan membawa Anda langsung ke halaman Opsi Aplikasi. Jika Anda menempuh rute ini, lewati petunjuk nanti di bagian ini yang menjelaskan cara menggunakan halaman Opsi Aplikasi.

**5. Pilih Selanjutnya.**

The screenshot shows the 'Publish to Amazon Web Services' wizard window. The title bar reads 'Publish to Amazon Web Services'. The main content area is titled 'Application Environment' and includes the instruction: 'Enter the details for your new application environment. To create a new new environment for an existing application, select the appropriate application.' On the left, a navigation pane lists 'Application', 'Environment' (highlighted), 'AWS Options', 'VPC', 'Updates', 'Options', and 'Review'. The main form has three sections: 'Application' with a 'Name' dropdown set to 'AEBWebAppDemo'; 'Environment' with a 'Name' dropdown; and 'URL' with a text input field containing 'http:' followed by a blurred domain, '.elasticbeanstalk.com', and a 'Check availability...' button. A green checkmark message below the URL field states 'The requested URL is available'. At the bottom, there are four buttons: 'Close', 'Back', 'Next', and 'Finish'.

6. Pada halaman Application Environment, di area Application, daftar drop-down Nama mengusulkan nama default untuk aplikasi. Anda dapat mengubah nama default dengan memilih nama yang berbeda dari daftar pilihan menurun.
7. Di area Lingkungan, dalam daftar drop-down Nama, ketik nama untuk lingkungan Elastic Beanstalk Anda. Dalam konteks ini, istilah lingkungan mengacu pada infrastruktur ketentuan Elastic Beanstalk untuk aplikasi Anda. Nama default mungkin sudah diusulkan dalam daftar drop-down ini. Jika nama default belum diusulkan, Anda dapat mengetikkan satu atau memilih salah satu dari daftar drop-down, jika ada nama tambahan yang tersedia. Nama lingkungan tidak boleh lebih dari 23 karakter.

8. Di area URL, kotak mengusulkan subdomain default. `elasticbeanstalk.com` yang akan menjadi URL untuk aplikasi web Anda. Anda dapat mengubah subdomain default dengan mengetikkan nama subdomain baru.
9. Pilih Periksa ketersediaan untuk memastikan URL untuk aplikasi web Anda belum digunakan.
10. Jika URL untuk aplikasi web Anda boleh digunakan, pilih Berikutnya.

**Publish to Amazon Web Services**

**AWS**  
Set Amazon EC2 and other AWS-related options for the deployed application.

**Application**  
Environment  
**AWS Options**  
VPC  
Updates  
Options  
Review

**Amazon EC2 Launch Configuration**

Container type \*: 64bit Windows Server 2012 R2 running IIS 8.5

Instance type \*: Micro Key pair \*: MyKeyPair

Use custom AMI:

Use a VPC  Single instance environment  Enable Rolling Deployments

**Deployed Application Permissions**

Role: aws-elasticbeanstalk-ec2-role

*The permissions for the Identity and Access Management role can be updated after the environment is created.*

**Relational Database Access**

*Select the Amazon RDS security groups to be modified to permit access from the EC2 instance(s) hosting your application.*

default

Close Back Next Finish

1. Pada halaman AWSOps, dalam Konfigurasi Peluncuran Amazon EC2, dari daftar drop-down Jenis Kontainer, pilih jenis Amazon Machine Image (AMI) yang akan digunakan untuk aplikasi Anda.
2. Dalam daftar drop-down Jenis Instans, tentukan jenis instans Amazon EC2 yang akan digunakan. Untuk contoh ini, kami sarankan Anda menggunakan Micro. Ini akan meminimalkan biaya yang terkait dengan menjalankan instans. Untuk informasi selengkapnya tentang biaya Amazon EC2, kunjungi halaman [Harga EC2](#).
3. Dalam daftar drop-down key pair, pilih pasangan kunci instans Amazon EC2 yang akan digunakan untuk masuk ke instans yang akan digunakan untuk aplikasi Anda.

4. Secara opsional, dalam kotak Use custom AMI, Anda dapat menentukan AMI kustom yang akan menimpa AMI yang ditentukan dalam daftar drop-down Container type. Untuk informasi selengkapnya tentang cara membuat AMI kustom, buka [Menggunakan AMI Kustom](#) di [Panduan Pengembang AWS Elastic Beanstalk](#) dan [Buat AMI dari Instans Amazon EC2](#).
5. Secara opsional, jika Anda ingin meluncurkan instance Anda di VPC, pilih kotak Use a VPC.
6. Secara opsional, jika Anda ingin meluncurkan instans Amazon EC2 tunggal dan kemudian menerapkan aplikasi Anda ke dalamnya, pilih kotak lingkungan Instans tunggal.

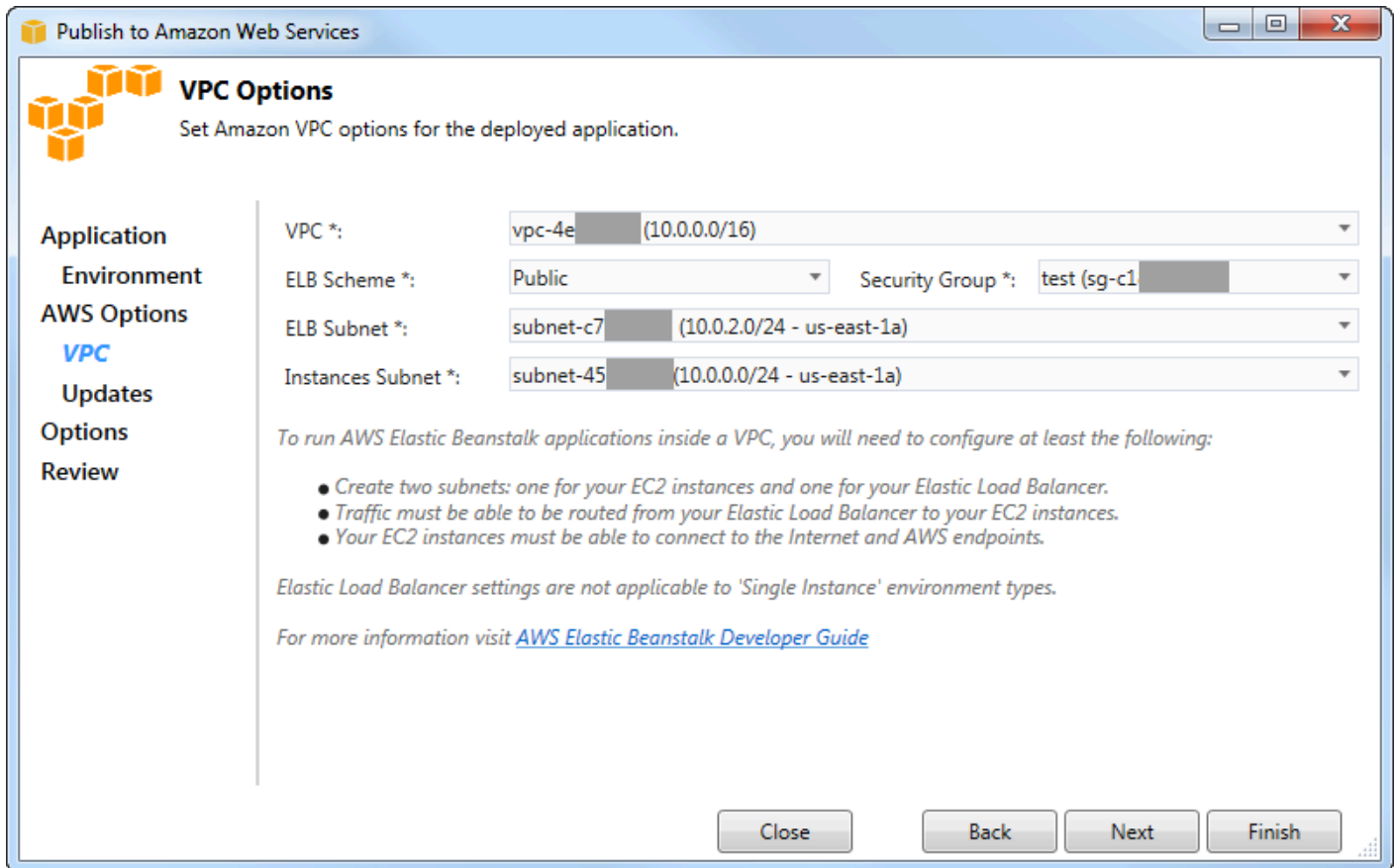
Jika Anda memilih kotak ini, Elastic Beanstalk akan tetap membuat grup Auto Scaling, tetapi tidak akan mengkonfigurasinya. Jika Anda ingin mengkonfigurasi grup Auto Scaling nanti, Anda dapat menggunakan file AWS Management Console.

7. Secara opsional, jika Anda ingin mengontrol kondisi di mana aplikasi Anda diterapkan ke instance, pilih kotak Enable Rolling Deployments. Anda dapat memilih kotak ini hanya jika Anda belum memilih kotak lingkungan contoh tunggal.
8. Jika aplikasi Anda menggunakan AWS layanan seperti Amazon S3 dan DynamoDB, cara terbaik untuk memberikan kredensi adalah dengan menggunakan peran IAM. Di area Izin Aplikasi yang Diterahkan, Anda dapat memilih peran IAM yang ada atau membuatnya yang akan digunakan wizard untuk meluncurkan lingkungan Anda. Aplikasi yang menggunakan AWS SDK for .NET akan secara otomatis menggunakan kredensi yang disediakan oleh peran IAM ini ketika membuat permintaan ke AWS layanan.
9. Jika aplikasi Anda mengakses database Amazon RDS, dalam daftar drop-down di area Akses Database Relasional, pilih kotak di samping grup keamanan Amazon RDS mana pun yang akan diperbarui oleh wizard sehingga instans Amazon EC2 Anda dapat mengakses database tersebut.

#### 10 Pilih Selanjutnya.

- Jika Anda memilih Gunakan VPC, halaman Opsi VPC akan muncul.
- Jika Anda memilih Aktifkan Penyebaran Bergulir, tetapi tidak memilih Gunakan VPC, halaman Rolling Deployment akan muncul. Lewati petunjuk nanti di bagian ini yang menjelaskan cara menggunakan halaman Rolling Deployment.
- Jika Anda tidak memilih Gunakan VPC atau Aktifkan Penyebaran Bergulir, halaman Opsi Aplikasi akan muncul. Lewati petunjuk nanti di bagian ini yang menjelaskan cara menggunakan halaman Opsi Aplikasi.

- 11 Jika Anda memilih Gunakan VPC, tentukan informasi di halaman Opsi VPC untuk meluncurkan aplikasi Anda ke VPC.



VPC harus telah dibuat. Jika Anda membuat VPC di Toolkit for Visual Studio, Toolkit for Visual Studio akan mengisi halaman ini untuk Anda. Jika Anda membuat VPC di [AWSManagement Console](#), ketik informasi tentang VPC Anda ke halaman ini.

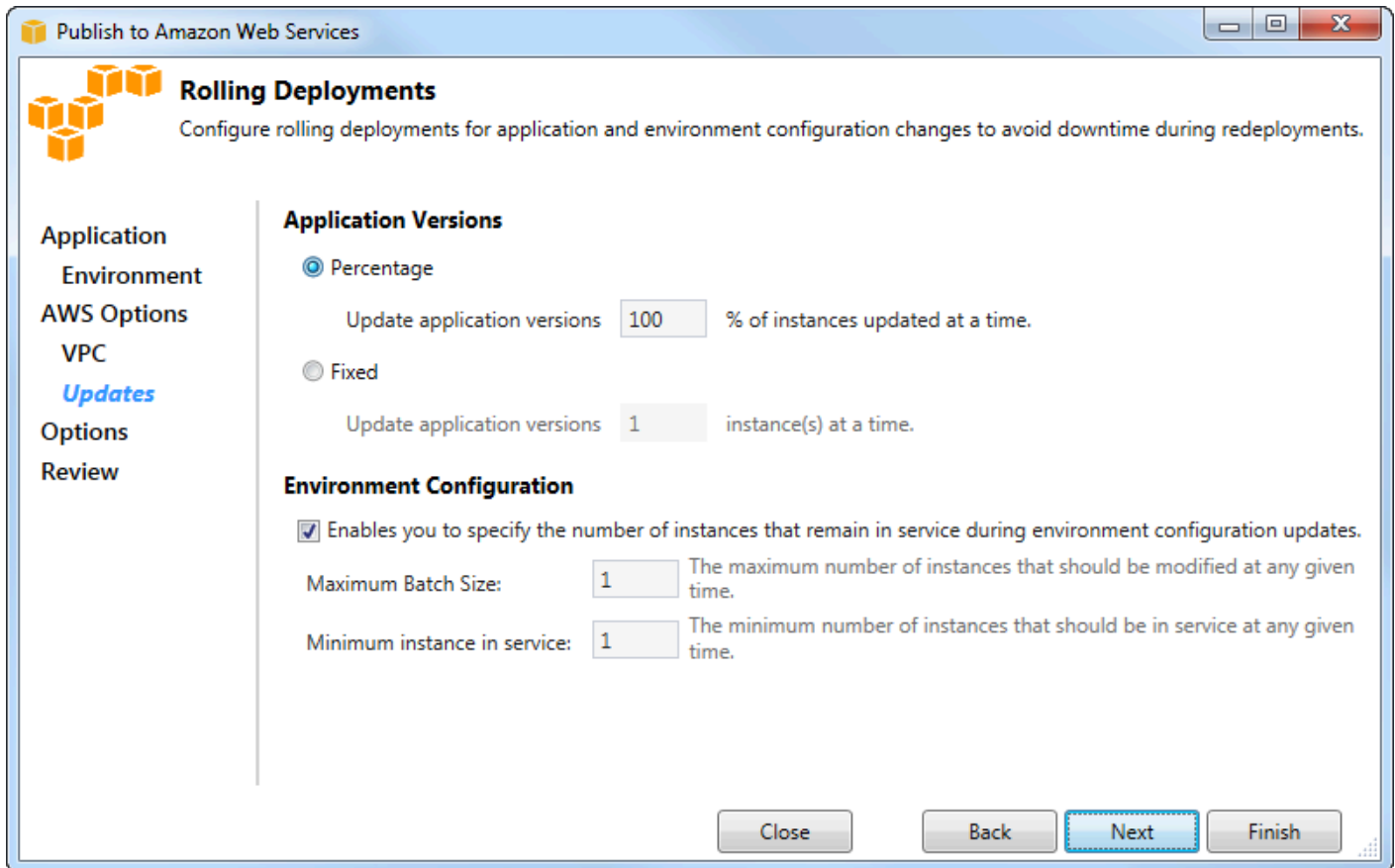
## Pertimbangan utama untuk penyebaran ke VPC

- VPC Anda membutuhkan setidaknya satu subnet publik dan satu subnet privat.
- Dalam daftar drop-down ELB Subnet, tentukan subnet publik. Toolkit for Visual Studio menyebarkan load balancer Elastic Load Balancing untuk aplikasi Anda ke subnet publik. Subnet publik dikaitkan dengan tabel routing yang memiliki entri yang menunjuk ke gateway Internet. Anda dapat mengenali gateway Internet karena memiliki ID yang dimulai dengan `igw-` (misalnya, `igw-83cddaex`). Subnet publik yang Anda buat menggunakan Toolkit for Visual Studio memiliki nilai tag yang mengidentifikasi mereka sebagai publik.
- Dalam daftar drop-down Instances Subnet, tentukan subnet pribadi. Toolkit for Visual Studio menerapkan instans Amazon EC2 untuk aplikasi Anda ke subnet pribadi.

- Instans Amazon EC2 untuk aplikasi Anda berkomunikasi dari subnet pribadi ke Internet melalui instans Amazon EC2 di subnet publik yang melakukan terjemahan alamat jaringan (NAT). Untuk mengaktifkan komunikasi ini, Anda memerlukan [grup keamanan VPC](#) yang memungkinkan lalu lintas mengalir dari subnet pribadi ke instance NAT. Tentukan grup keamanan VPC ini di Grup Keamanan daftar drop-down.

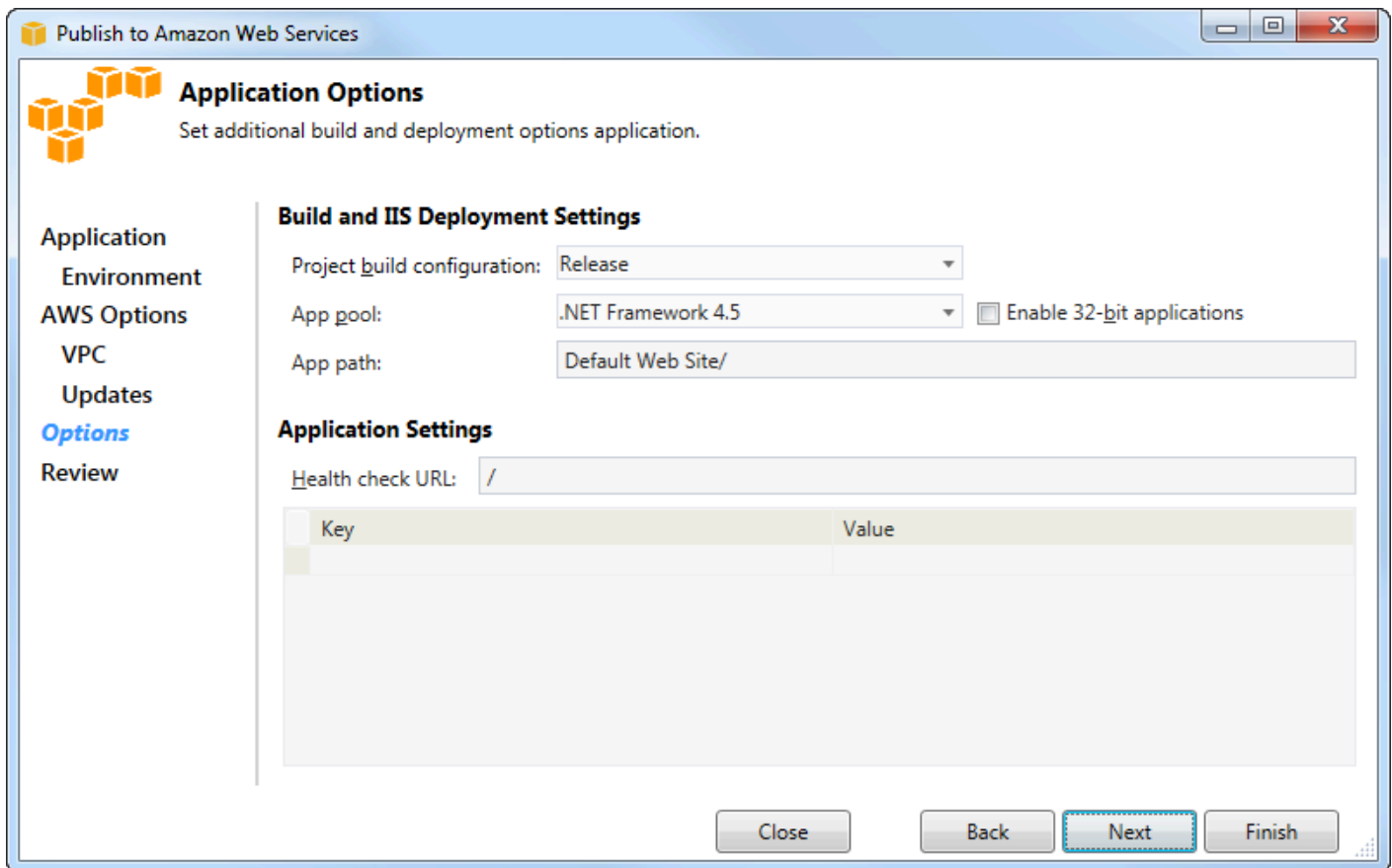
Untuk informasi lebih lanjut tentang cara menyebarkan aplikasi Elastic Beanstalk ke VPC, buka [Panduan Pengembang AWS Elastic Beanstalk](#).

1. Setelah Anda mengisi semua informasi di halaman Opsi VPC, pilih Berikutnya.
  - Jika Anda memilih Aktifkan Rolling Deployment, halaman Rolling Deployment akan muncul.
  - Jika Anda tidak memilih Aktifkan Penyebaran Bergulir, halaman Opsi Aplikasi akan muncul. Lewati petunjuk nanti di bagian ini yang menjelaskan cara menggunakan halaman Opsi Aplikasi.
2. Jika Anda memilih Aktifkan Rolling Deployment, Anda menentukan informasi di halaman Rolling Deployment untuk mengonfigurasi bagaimana versi baru aplikasi Anda diterapkan ke instance di lingkungan load-balanced. Misalnya, jika Anda memiliki empat instance di lingkungan Anda dan ingin mengubah jenis instans, Anda dapat mengonfigurasi lingkungan untuk mengubah dua instance sekaligus. Ini membantu memastikan aplikasi Anda masih berjalan saat perubahan sedang dilakukan.



3. Di area Versi Aplikasi, pilih opsi untuk mengontrol penerapan ke persentase atau jumlah instance dalam satu waktu. Tentukan persentase atau angka yang diinginkan.
4. Secara opsional, di area Konfigurasi Lingkungan, pilih kotak jika Anda ingin menentukan jumlah instance yang tetap dalam layanan selama penerapan. Jika Anda memilih kotak ini, tentukan jumlah maksimum instans yang harus diubah dalam % satu waktu, jumlah minimum instance yang harus tetap dalam layanan pada satu waktu, atau keduanya.
5. Pilih Selanjutnya.
6. Pada halaman Opsi Aplikasi, Anda menentukan informasi tentang build, Internet Information Services (IIS), dan pengaturan aplikasi.

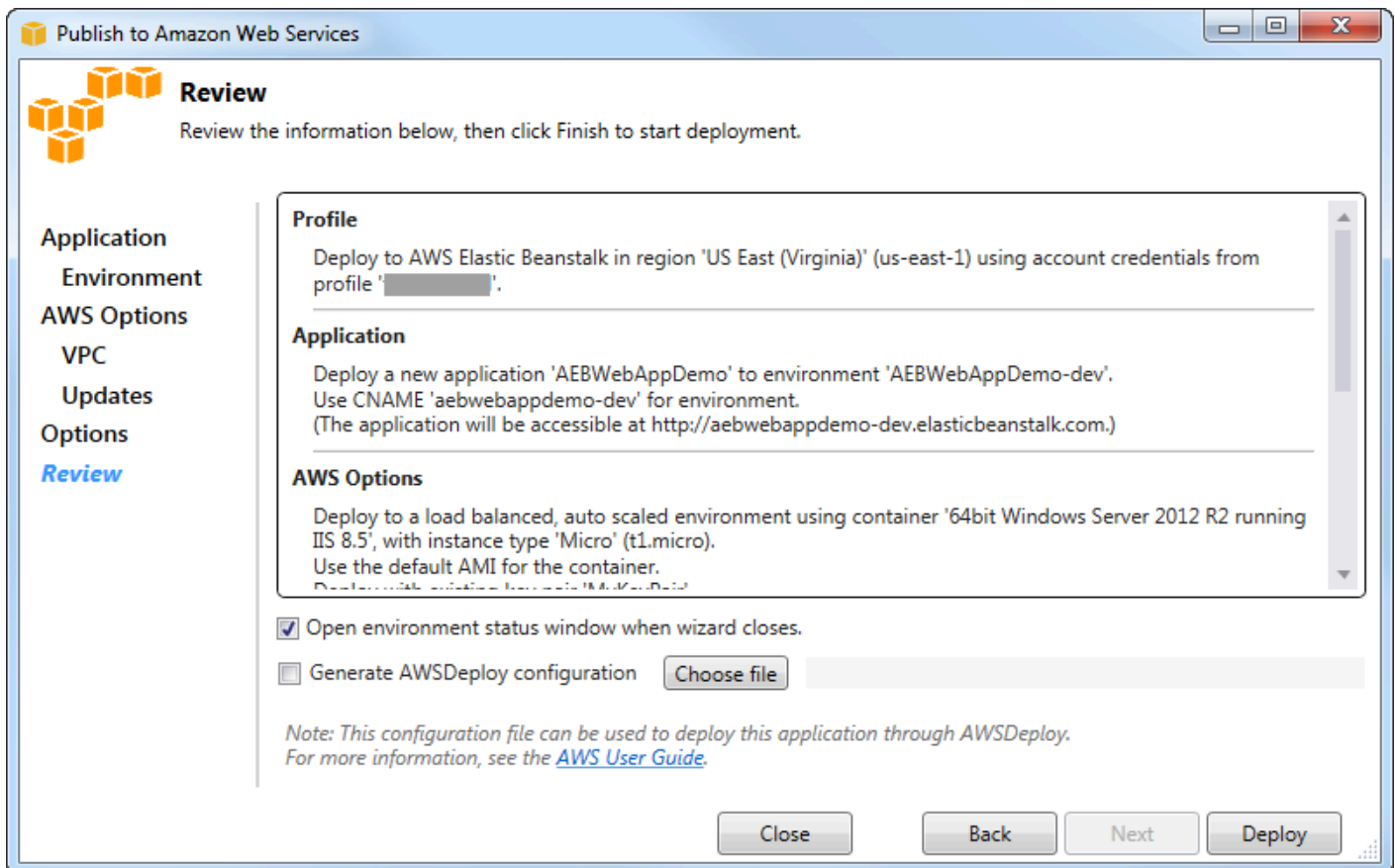




7. Di area Pengaturan Penyebaran Build dan IIS, dalam daftar drop-down Project build configuration, pilih konfigurasi build target. Jika wizard dapat menemukannya, Rilis muncul sebaliknya, konfigurasi aktif ditampilkan di kotak ini.
8. Dalam daftar drop-down App pool, pilih versi .NET Framework yang diperlukan oleh aplikasi Anda. Versi .NET Framework yang benar seharusnya sudah ditampilkan.
9. Jika aplikasi Anda 32-bit, pilih kotak Aktifkan aplikasi 32-bit.
- 10 Di kotak App path, tentukan path yang akan digunakan IIS untuk menyebarkan aplikasi. Secara default, Default Web Site/ ditentukan, yang biasanya diterjemahkan ke jalur `:\inetpub\wwwroot`. Jika Anda menentukan jalur selain Default Web Site/, wizard akan menempatkan redirect di Default Web Site/ path yang menunjuk ke jalur yang Anda tentukan.
- 11 Di area Pengaturan Aplikasi, di kotak URL centang Health, ketik URL untuk Elastic Beanstalk untuk memeriksa untuk menentukan apakah aplikasi web Anda masih responsif. URL ini relatif terhadap URL server root. URL server root ditentukan secara default. Misalnya, jika URL lengkapnya `example.com/site-is-up.html`, Anda akan mengetik `/site-is-up.html`.
- 12 Di area untuk Kunci dan Nilai, Anda dapat menentukan pasangan kunci dan nilai apa pun yang ingin Anda tambahkan ke `Web.config` file aplikasi Anda.

**Note**

Meskipun tidak disarankan, Anda dapat menggunakan area untuk Key dan Value, untuk menentukan AWS kredensial di mana aplikasi Anda harus dijalankan. Pendekatan yang disukai adalah menentukan peran IAM dalam daftar drop-down Identity and Access Management Role pada halaman AWSOptions. Namun, jika Anda harus menggunakan AWS kredensial alih-alih peran IAM untuk menjalankan aplikasi Anda, di baris Key, pilih AWSAccessKey. Di baris Nilai, ketik tombol akses. Ulangi langkah-langkah ini untuk AWSSecretKey.

**13Pilih Selanjutnya.**

14 Pada halaman Tinjau, tinjau opsi yang Anda konfigurasi, lalu pilih jendela Buka status lingkungan saat wizard ditutup.

15 Jika semuanya terlihat benar, pilih Deploy.

**Note**

Ketika Anda menerapkan aplikasi, akun aktif akan dikenakan biaya untuk AWS sumber daya yang digunakan oleh aplikasi.

Informasi tentang penyebaran akan muncul di bilah status Visual Studio dan jendela Output. Ini mungkin memakan waktu beberapa menit. Saat penyebaran selesai, pesan konfirmasi akan muncul di jendela Output.

16. Untuk menghapus deployment, dalam AWS Explorer, perluas node Elastic Beanstalk, buka menu konteks (klik kanan) untuk subnode untuk deployment, lalu pilih Mengelola. Proses penghapusan mungkin memakan waktu beberapa menit.

## Men-deploy aplikasi ASP.NET Core ke Elastic Beanstalk (Legacy)

**Important**

Dokumentasi ini mengacu pada layanan dan fitur lama. Untuk panduan dan konten yang diperbarui, lihat panduan [alat penyebaran AWS .NET](#) dan [Deploying to AWS](#) table of contents yang diperbarui.

AWS Elastic Beanstalk adalah layanan yang menyederhanakan proses penyediaan AWS sumber daya untuk aplikasi Anda. AWS Elastic Beanstalk menyediakan semua AWS infrastruktur yang diperlukan untuk menyebarkan aplikasi Anda.

Toolkit for Visual Studio mendukung penerapan aplikasi ASP.NET Core untuk AWS menggunakan Elastic Beanstalk. ASP.NET Core adalah desain ulang ASP.NET dengan arsitektur termodulasi yang meminimalkan ketergantungan overhead dan merampingkan aplikasi Anda untuk berjalan di awan.

AWS Elastic Beanstalk membuatnya mudah untuk menyebarkan aplikasi dalam berbagai bahasa yang berbeda untuk AWS. Elastic Beanstalk mendukung aplikasi ASP.NET tradisional dan aplikasi ASP.NET Core. Topik ini menjelaskan penerapan aplikasi ASP.NET Core.

## Menggunakan Wizard Deployment

Cara termudah untuk menyebarkan aplikasi ASP.NET Core ke Elastic Beanstalk adalah dengan Toolkit for Visual Studio.

Jika Anda telah menggunakan toolkit sebelumnya untuk menyebarkan ASP tradisional. NET, Anda akan menemukan pengalaman untuk ASP.NET Core menjadi sangat mirip. Pada langkah-langkah di bawah ini, kita akan berjalan melalui pengalaman penyebaran.

Jika Anda belum pernah menggunakan kit alat sebelumnya, hal pertama yang harus Anda lakukan setelah menginstal kit alat adalah mendaftarkan AWS kredensial Anda dengan kit alat. Lihat [Cara Menentukan Kredensial AWS Keamanan untuk Aplikasi Anda](#) untuk dokumentasi Visual Studio untuk detail tentang cara melakukannya.

Untuk menyebarkan aplikasi web ASP.NET Core, klik kanan proyek di Solution Explorer dan pilih Publish to AWS...

Pada halaman pertama Wisaya Publish to AWS Elastic Beanstalk deployment, pilih untuk membuat aplikasi Elastic Beanstalk baru. Aplikasi Elastic Beanstalk adalah sebuah koleksi logis komponen Elastic Beanstalk, termasuk Lingkungan, versi, dan Konfigurasi lingkungan. Wizard penyebaran menghasilkan aplikasi yang pada gilirannya berisi kumpulan versi aplikasi dan lingkungan. Lingkungan berisi AWS sumber daya aktual yang menjalankan versi aplikasi. Setiap kali Anda men-deploy aplikasi, versi aplikasi baru dibuat dan wizard mengarahkan lingkungan ke versi itu. Anda dapat mempelajari lebih lanjut tentang konsep-konsep ini di [Komponen Elastic Beanstalk](#).

Selanjutnya, tetapkan nama untuk aplikasi dan lingkungan pertamanya. Setiap lingkungan memiliki CNAME unik yang terkait dengannya yang dapat Anda gunakan untuk mengakses aplikasi saat penyebaran selesai.

Halaman berikutnya, AWS Pilihan, memungkinkan Anda untuk mengkonfigurasi jenis AWS sumber daya yang akan digunakan. Untuk contoh ini, tinggalkan nilai default, kecuali untuk bagian Key pair. Pasangan kunci memungkinkan Anda mengambil kata sandi administrator Windows sehingga Anda dapat masuk ke mesin. Jika Anda belum membuat key pair, Anda mungkin ingin memilih Buat key pair baru.

### Izin

Halaman Izin digunakan untuk menetapkan AWS kredensial ke instans EC2 yang menjalankan aplikasi Anda. Hal ini penting jika aplikasi Anda menggunakan AWS SDK for .NET untuk mengakses AWS

layanan lain. Jika Anda tidak menggunakan layanan lain dari aplikasi Anda maka Anda dapat meninggalkan halaman ini secara default.

## Opsi Aplikasi

Rincian pada halaman Opsi Aplikasi berbeda dari yang ditentukan saat menyebarkan aplikasi ASP.NET tradisional. Di sini, Anda menentukan konfigurasi build dan kerangka kerja yang digunakan untuk mengemas aplikasi, dan juga menentukan jalur sumber daya IIS untuk aplikasi.

Setelah menyelesaikan halaman Opsi Aplikasi, klik Berikutnya untuk meninjau pengaturan, lalu klik Deploy untuk memulai proses penyebaran.

## Memeriksa status lingkungan

Setelah aplikasi dikemas dan diunggah keAWS, Anda dapat memeriksa status lingkungan Elastic Beanstalk dengan membuka tampilan status lingkungan dariAWS Explorer di Visual Studio.

Acara ditampilkan di bilah status saat lingkungan online. Setelah semuanya selesai, status lingkungan akan beralih ke keadaan sehat. Anda dapat mengklik URL untuk melihat situs. Dari sini, Anda juga dapat menarik log dari lingkungan atau desktop jarak jauh ke instans Amazon EC2 yang merupakan bagian dari lingkungan Elastic Beanstalk Anda.

Penyebaran pertama aplikasi apa pun akan memakan waktu sedikit lebih lama daripada penyebaran ulang berikutnya, karena menciptakanAWS sumber daya baru. Saat Anda melakukan iterasi pada aplikasi Anda selama pengembangan, Anda dapat dengan cepat menyebarkan ulang dengan kembali melalui wizard, atau memilih opsi Publikasikan ulang saat Anda mengklik kanan proyek.

Publikasikan ulang paket aplikasi Anda menggunakan pengaturan dari proses sebelumnya melalui wizard penyebaran dan upload bundel aplikasi ke lingkungan Elastic Beanstalk yang ada.

## Cara MenentukanAWSKredensial Keamanan untuk Aplikasi Anda

ParameterAWSakun yang Anda tentukan diMenerbitkan ke Elastic BeanstalkwizardAWSakun wizard akan digunakan untuk deployment ke Elastic Beanstalk.

Meskipun tidak disarankan, Anda mungkin perlu menentukanAWSkredensi akun yang akan digunakan aplikasi Anda untuk mengaksesAWSlayanan setelah itu telah dikerahkan. Pendekatan yang disukai adalah menentukan peran IAM. DiMenerbitkan ke Elastic Beanstalkpenyihir, Anda melakukan ini melaluiPeran Identity and Access Managementdaftar drop-down

pada AWS Ops konsol. Dalam warisan Menerbitkan ke Amazon Web Services, Anda melakukan ini melalui IAM Role daftar drop-down pada AWS Ops konsol.

Jika Anda harus menggunakan AWS kredensial akun alih-alih peran IAM, Anda dapat menentukan AWS kredensial akun untuk aplikasi Anda dengan salah satu cara berikut:

- Referensi profil yang sesuai dengan AWS kredensial akun di `appSettings` elemen proyek `Web.config` berkas. (Untuk membuat profil, lihat [konfigurasi AWS kredensial](#).) Contoh berikut menentukan kredensial yang nama profilnya `myProfile`.

```
<appSettings>
  <!-- AWS CREDENTIALS -->
  <add key="AWSProfileName" value="myProfile"/>
</appSettings>
```

- Jika Anda menggunakan Menerbitkan ke Elastic Beanstalk wizard, pada Opsi Aplikasi konsol, di Kunci retan Kunci dan Nilai area, pilih `AWSAccessKey`. Di Nilai baris, ketik tombol akses. Ulangi langkah-langkah ini untuk `AWSecretKey`.
- Jika Anda menggunakan warisan Menerbitkan ke Amazon Web Services wizard, pada Opsi Aplikasi konsol, di Kredensial Aplikasi area, pilih `Gunakan kredensial ini`, dan kemudian ketik kunci akses dan kunci akses rahasia ke `Access key` dan `Kunci rahasia` kotak.

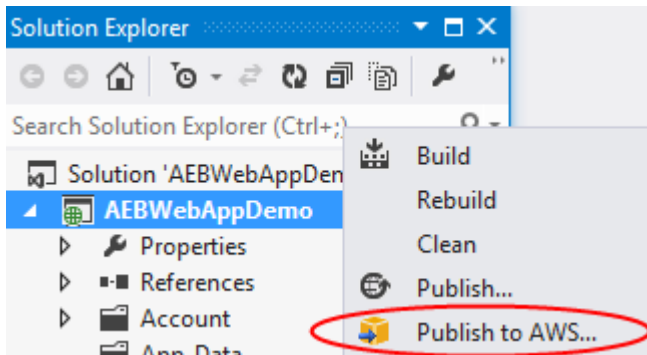
## Cara Menerbitkan Ulang Aplikasi Anda ke Lingkungan Elastic Beanstalk (Legacy)

### Important

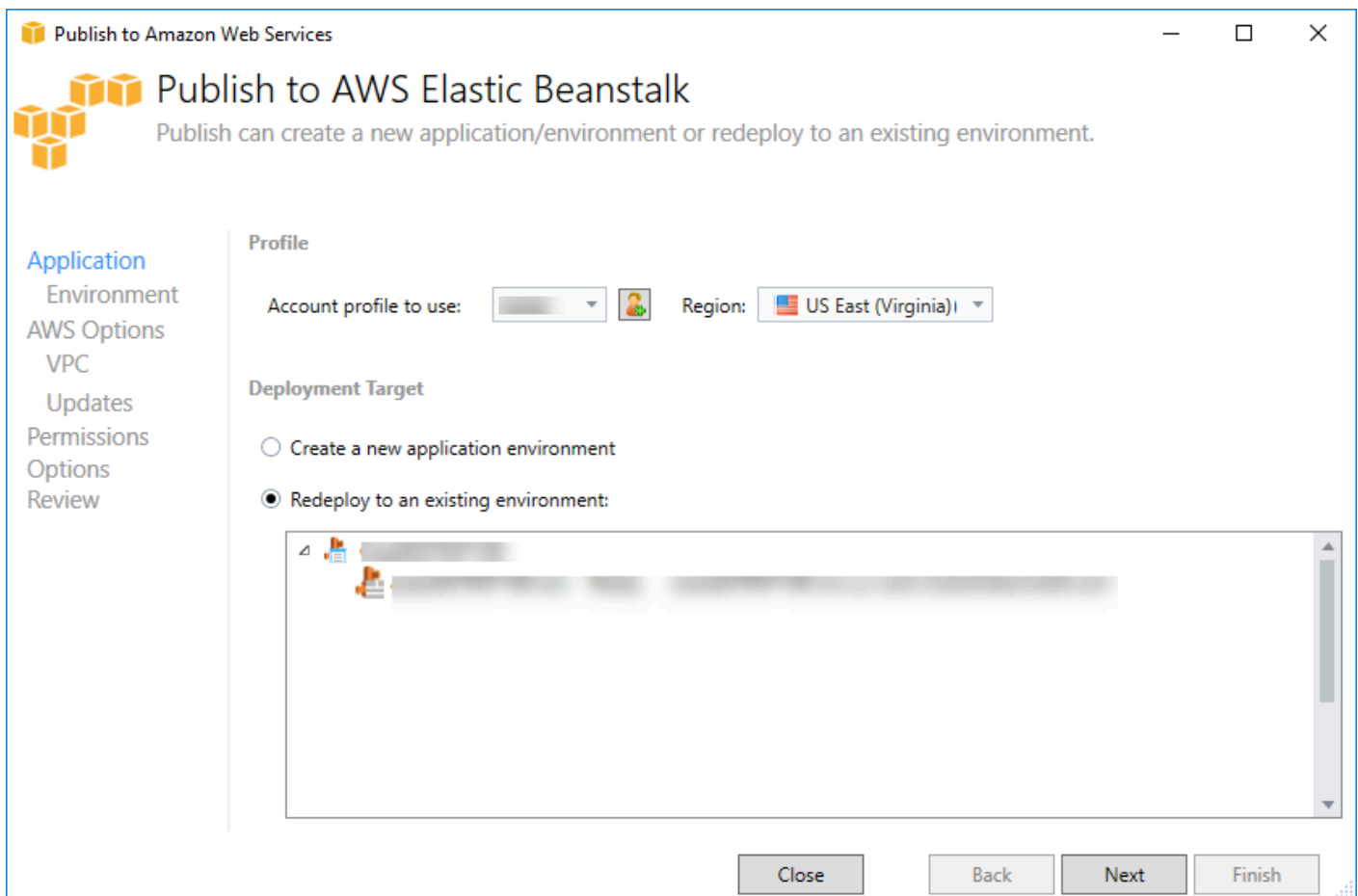
Dokumentasi ini mengacu pada layanan dan fitur lama. Untuk panduan dan konten yang diperbarui, lihat panduan [alat penyebaran AWS .NET](#) dan [Deploying to AWS](#) table of contents yang diperbarui.

Anda dapat melakukan iterasi pada aplikasi Anda dengan membuat perubahan diskrit dan kemudian menerbitkan ulang versi baru ke lingkungan Elastic Beanstalk yang sudah diluncurkan.

1. Dalam Solution Explorer Explorer, buka menu konteks (klik kanan) untuk folder `WebAppDemo` proyek AEB, buka menu konteks (klik kanan) untuk folder proyek AEB, dan pilih `Mengelola ke AWS Elastic Beanstalk`.

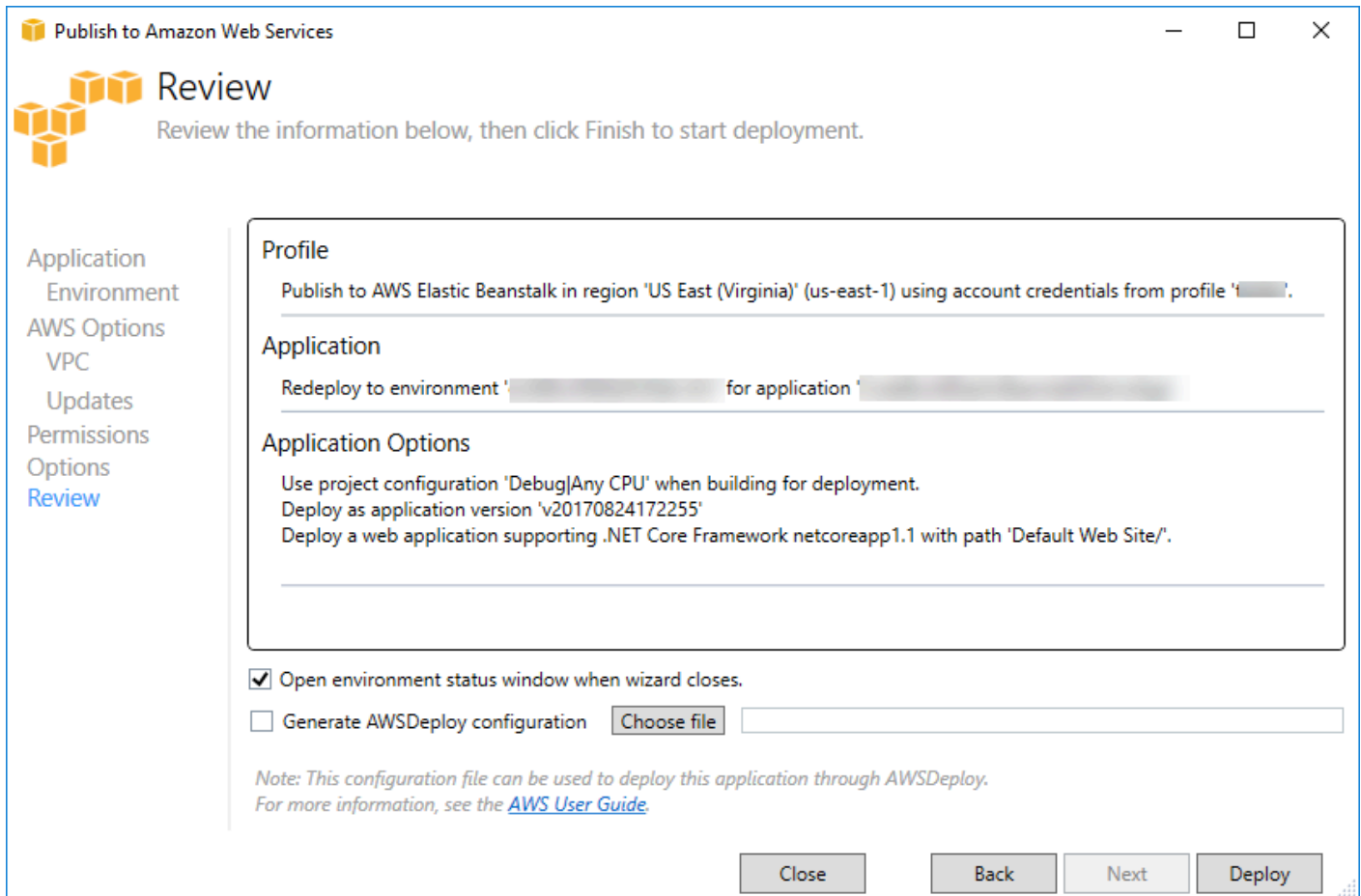


Wisaya Publish to Elastic Beanstalk Kacang muncul.



2. Pilih Redeploy ke lingkungan yang ada dan pilih lingkungan yang sebelumnya Anda publikasikan. Klik Selanjutnya.

Wisaya Tinjauan muncul.



3. Klik Deploy. Aplikasi akan redeploy ke lingkungan yang sama.

Anda tidak dapat mempublikasikan ulang jika aplikasi Anda sedang dalam proses peluncuran atau penghentian.

## Penyebaran Aplikasi Elastic Beanstalk Kustom

Topik ini menjelaskan bagaimana manifes penyebaran untuk wadah Microsoft Windows Elastic Beanstalk mendukung penerapan aplikasi kustom.

Penerapan aplikasi khusus adalah fitur canggih bagi pengguna tingkat lanjut yang ingin memanfaatkan kekuatan Elastic Beanstalk untuk membuat dan mengelola merekaAWS sumber daya, tetapi ingin kontrol penuh tentang bagaimana aplikasi mereka dikerahkan. Untuk penerapan aplikasi kustom, Anda membuat skrip Windows PowerShell untuk tiga tindakan berbeda Elastic Beanstalk melakukan. Tindakan install digunakan ketika penyebaran dimulai, restart digunakan ketika `RestartAppServerAPI` dipanggil baik dari toolkit atau konsol web, dan uninstall yang dipanggil pada setiap penyebaran sebelumnya setiap kali penyebaran baru terjadi.



Misalnya, Anda mungkin memiliki aplikasi ASP.NET yang ingin Anda gunakan sementara tim dokumentasi Anda telah menulis situs web statis yang mereka inginkan disertakan dengan penyebaran. Anda dapat melakukannya dengan menulis manifes penyebaran Anda seperti ini:

```
{
  "manifestVersion": 1,
  "deployments": {
    "msDeploy": [
      {
        "name": "app",
        "parameters": {
          "appBundle": "CoolApp.zip",
          "iisPath": "/"
        }
      }
    ],
    "custom": [
      {
        "name": "PowerShellDocs",
        "scripts": {
          "install": {
            "file": "install.ps1"
          },
          "restart": {
            "file": "restart.ps1"
          },
          "uninstall": {
            "file": "uninstall.ps1"
          }
        }
      }
    ]
  }
}
```

Skrip yang terdaftar untuk setiap tindakan harus dalam bundel aplikasi relatif terhadap file manifes penyebaran. Untuk contoh ini, bundel aplikasi juga akan berisi file `documentation.zip` yang berisi situs web statis yang dibuat oleh tim dokumentasi Anda.

Parameter `install.ps1` ekstrak file zip dan set up IIS Path.

```
Add-Type -assembly "system.io.compression.filesystem"  
[io.compression.zipfile]::ExtractToDirectory('./documentation.zip', 'c:\inetpub\wwwroot\documentation')  
  
powershell.exe -Command {New-WebApplication -Name documentation -PhysicalPath c:\inetpub\wwwroot\documentation -Force}
```

Karena aplikasi Anda berjalan di IIS, tindakan restart akan memanggil IIS reset.

```
iisreset /timeout:1
```

Untuk menghapus skrip, penting untuk membersihkan semua pengaturan dan file yang digunakan selama tahap instalasi. Dengan cara itu selama fase instalasi untuk versi baru, Anda dapat menghindari tabrakan dengan penyebaran sebelumnya. Untuk contoh ini, Anda perlu menghapus aplikasi IIS untuk situs web statis dan menghapus file situs web.

```
powershell.exe -Command {Remove-WebApplication -Name documentation}  
Remove-Item -Recurse -Force 'c:\inetpub\wwwroot\documentation'
```

Dengan file script ini dan file `documentation.zip` termasuk dalam bundel aplikasi Anda, penyebaran menciptakan aplikasi ASP.NET dan kemudian menyebarkan situs dokumentasi.

Untuk contoh ini, kita memilih contoh sederhana yang menyebarkan situs web statis sederhana, tetapi dengan penerapan aplikasi khusus Anda dapat menyebarkan semua jenis aplikasi dan membiarkan Elastic Beanstalk mengelola AWS sumber daya untuk itu.

## Khusus ASP.NET Core Elastic Beanstalk deployment

Topik ini menjelaskan bagaimana penyebaran bekerja dan apa yang dapat Anda lakukan menyesuaikan penyebaran saat membuat aplikasi ASP.NET Core dengan Elastic Beanstalk dan Toolkit for Visual Studio.

Setelah Anda menyelesaikan wizard penyebaran di Toolkit for Visual Studio, toolkit bundel aplikasi dan mengirimkannya ke Elastic Beanstalk. Langkah pertama Anda dalam membuat bundel aplikasi adalah dengan menggunakan CLI dotnet baru untuk mempersiapkan aplikasi untuk penerbitan dengan menggunakan `menerbitkanperintah`. Kerangka kerja dan konfigurasi diturunkan dari pengaturan di wizard `kemenerbitkanperintah`. Jadi jika Anda memilih `Rilis untuk konfigurasi dan netcoreapp1.0 untuk framework`, toolkit akan menjalankan perintah berikut:

```
dotnet publish --configuration Release --framework netcoreapp1.0
```

Saat menerbitkan perintah selesai, toolkit menulis manifes penyebaran baru ke dalam folder penerbitan. Manifes penyebaran adalah file JSON bernama `aws-windows-deployment-manifest.json`, yang Elastic Beanstalk Windows container (versi 1.2 atau yang lebih baru) dibaca untuk menentukan cara menyebarkan aplikasi. Misalnya, untuk aplikasi ASP.NET Core Anda ingin menyebarkan pada akar IIS, toolkit menghasilkan file manifes yang terlihat seperti ini:

```
{
  "manifestVersion": 1,
  "deployments": {
    "aspNetCoreWeb": [
      {
        "name": "app",
        "parameters": {
          "appBundle": ".",
          "iisPath": "/",
          "iisWebSite": "Default Web Site"
        }
      }
    ]
  }
}
```

Parameter `appBundle` properti menunjukkan di mana bit aplikasi dalam kaitannya dengan file manifes. Properti ini dapat menunjuk ke direktori atau arsip ZIP.

Parameter `iisPath` dan `iisWebSite` properti menunjukkan di mana di IIS untuk meng-host aplikasi.

## Menyesuaikan manifes

Toolkit hanya menulis file manifes jika salah satu belum ada di folder penerbitan. Jika file tidak ada, toolkit akan memperbarui `appBundle`, `iisPath` dan `iisWebSite` properti dalam aplikasi pertama yang tercantum di bawah `aspNetCoreWeb` bagian manifes. Hal ini memungkinkan Anda untuk menambahkan `aws-windows-deployment-manifest.json` untuk proyek Anda dan menyesuaikan manifes. Untuk melakukan hal ini untuk aplikasi ASP.NET Core Web di Visual Studio menambahkan file JSON baru ke akar proyek dan nama itu `aws-windows-deployment-manifest.json`.

Manifes harus diberi nama `aws-windows-deployment-manifest.json` dan itu harus menjadi akar proyek. Kontainer Elastic Beanstalk mencari manifes di root dan jika menemukan itu akan memanggil

perkakas penyebaran. Jika file tidak ada, kontainer Elastic Beanstalk jatuh kembali ke perkakas penyebaran lama, yang mengasumsikan arsip adalah `msdeployarsip`.

Untuk memastikan `dotnet CLI publish` perintah termasuk manifes, memperbarui `project.json` file untuk menyertakan file manifes di bagian `include` di bawah `includePublishOptions`.

```
{
  "publishOptions": {
    "include": [
      "wwwroot",
      "Views",
      "Areas/**/Views",
      "appsettings.json",
      "web.config",
      "aws-windows-deployment-manifest.json"
    ]
  }
}
```

Sekarang setelah Anda mendeklarasikan manifes sehingga disertakan dalam app bundle, Anda dapat mengkonfigurasi lebih lanjut bagaimana Anda ingin menyebarkan aplikasi. Anda dapat menyesuaikan penyebaran di luar apa yang didukung oleh wizard penyebaran. AWS telah mendefinisikan skema JSON untuk `aws-windows-deployment-manifest.json`, dan ketika Anda menginstal Toolkit for Visual Studio, setup mendaftarkan URL untuk skema.

Ketika Anda membuk `aws-windows-deployment-manifest.json`, Anda akan melihat URL skema yang dipilih di kotak drop down Schema. Anda dapat menavigasi ke URL untuk mendapatkan deskripsi lengkap tentang apa yang dapat diatur dalam manifes. Dengan skema yang dipilih, Visual Studio akan menyediakan IntelliSense saat Anda mengedit manifes.

Satu kustomisasi yang dapat Anda lakukan adalah untuk mengkonfigurasi kolam aplikasi IIS tempat aplikasi akan berjalan. Contoh berikut menunjukkan bagaimana Anda dapat menentukan kolam Aplikasi IIS ("CustomPool") yang mendaur ulang proses setiap 60 menit, dan menugaskan ke aplikasi menggunakan `"appPool": "customPool"`.

```
{
  "manifestVersion": 1,
  "iisConfig": {
    "appPools": [
      {
```

```
        "name": "customPool",
        "recycling": {
            "regularTimeInterval": 60
        }
    }
],
},
"deployments": {
    "aspNetCoreWeb": [
        {
            "name": "app",
            "parameters": {
                "appPool": "customPool"
            }
        }
    ]
}
}
```

Selain itu, manifes dapat mendeklarasikan skrip Windows PowerShell untuk dijalankan sebelum dan sesudah tindakan penginstalan, restart, dan uninstall. Misalnya, manifes berikut menjalankan skrip Windows PowerShell `PostInstallSetup.ps1` untuk melakukan pekerjaan setup lebih lanjut setelah aplikasi ASP.NET Core dikerahkan ke IIS. Saat menambahkan skrip seperti ini, pastikan skrip ditambahkan ke bagian `include` di bawah `publishOptions` di `project.json` file, seperti yang Anda lakukan dengan `aws-windows-deployment-manifest.json` berkas. Jika tidak, skrip tidak akan disertakan sebagai bagian dari CLI `dotnetmenerbitkan` perintah.

```
{
  "manifestVersion": 1,
  "deployments": {
    "aspNetCoreWeb": [
      {
        "name": "app",
        "scripts": {
          "postInstall": {
            "file": "SetupScripts/PostInstallSetup.ps1"
          }
        }
      }
    ]
  }
}
```

## Bagaimana dengan .ebextensions?

Pohon Kacang Elastic.ebextensionsfile konfigurasi didukung sebagai dengan semua kontainer Elastic Beanstalk lainnya. Untuk menyertakan .ebextensions dalam aplikasi ASP.NET Core, tambahkan .ebextensionsdirektori keincludebagian bawahpublishOptionsdi dalamproject.jsonberkas. Untuk informasi lebih lanjut tentang .ebextensions checkout[Panduan Pengembang Elastic Beanstalk](#).

## Support Beberapa Aplikasi untuk NET dan Elastic Beanstalk

Dengan menggunakan manifes penyebaran, Anda memiliki kemampuan untuk menyebarkan beberapa aplikasi ke lingkungan Elastic Beanstalk yang sama.

Manifes penyebaran mendukung[ASP.NET](#)aplikasi web serta arsip msdeploy untuk aplikasi ASP.NET tradisional. Bayangkan skenario di mana Anda telah menulis aplikasi baru yang menakjubkan menggunakan ASP.NET Core untuk frontend dan proyek API Web untuk API ekstensi. Anda juga memiliki aplikasi admin yang Anda tulis menggunakan ASP.NET tradisional.

Wizard penyebaran toolkit berfokus pada penerapan satu proyek. Untuk memanfaatkan beberapa penerapan aplikasi, Anda harus membangun bundel aplikasi dengan tangan. Untuk memulai, tulis manifes. Untuk contoh ini, Anda akan menulis manifes di akar solusi Anda.

Bagian penyebaran dalam manifes memiliki dua anak: array aplikasi web ASP.NET Core untuk menyebarkan, dan array arsip msdeploy untuk menyebarkan. Untuk setiap aplikasi, Anda mengatur jalur IIS dan lokasi bit aplikasi relatif terhadap manifes.

```
{
  "manifestVersion": 1,
  "deployments": {

    "aspNetCoreWeb": [
      {
        "name": "frontend",
        "parameters": {
          "appBundle": "./frontend",
          "iisPath": "/frontend"
        }
      },
      {
        "name": "ext-api",
        "parameters": {
          "appBundle": "./ext-api",
```

```
        "iisPath": "/ext-api"
    }
}
],
"msDeploy": [
{
    "name": "admin",
    "parameters": {
        "appBundle": "AmazingAdmin.zip",
        "iisPath": "/admin"
    }
}
]
}
}
```

Dengan manifes yang ditulis, Anda akan menggunakan Windows PowerShell untuk membuat bundel aplikasi dan memperbarui lingkungan Elastic Beanstalk yang ada untuk menjalankannya. Script ditulis dengan asumsi bahwa itu akan dijalankan dari folder yang berisi solusi Visual Studio Anda.

Hal pertama yang perlu Anda lakukan dalam skrip adalah menyiapkan folder ruang kerja untuk membuat bundel aplikasi.

```
$publishFolder = "c:\temp\publish"

$publishWorkspace = [System.IO.Path]::Combine($publishFolder, "workspace")
$appBundle = [System.IO.Path]::Combine($publishFolder, "app-bundle.zip")

If (Test-Path $publishWorkspace){
    Remove-Item $publishWorkspace -Confirm:$false -Force
}
If (Test-Path $appBundle){
    Remove-Item $appBundle -Confirm:$false -Force
}
```

Setelah Anda membuat folder, sekarang saatnya untuk mendapatkan frontend siap. Seperti halnya wizard penyebaran, gunakan CLI dotnet untuk mempublikasikan aplikasi.

```
Write-Host 'Publish the ASP.NET Core frontend'
$publishFrontendFolder = [System.IO.Path]::Combine($publishWorkspace, "frontend")
dotnet publish .\src\AmazingFrontend\project.json -o $publishFrontendFolder -c Release
-f netcoreapp1.0
```

Perhatikan bahwa subfolder “frontend” digunakan untuk folder output, cocok dengan folder yang Anda tetapkan dalam manifes. Sekarang Anda perlu melakukan hal yang sama untuk proyek Web API.

```
Write-Host 'Publish the ASP.NET Core extensibility API'
$publishExtAPIFolder = [System.IO.Path]::Combine($publishWorkspace, "ext-api")
dotnet publish .\src\AmazingExtensibleAPI\project.json -o $publishExtAPIFolder -c
Release -f netcoreapp1.0
```

Situs admin adalah aplikasi ASP.NET tradisional, sehingga Anda tidak dapat menggunakan CLI dotnet. Untuk aplikasi admin, Anda harus menggunakan msbuild, meneruskan paket target build untuk membuat arsip msdeploy. Secara default target paket menciptakan arsip msdeploy di bawahobj\Release\Packagefolder, sehingga Anda akan perlu untuk menyalin arsip ke ruang kerja publish.

```
Write-Host 'Create msdeploy archive for admin site'
msbuild .\src\AmazingAdmin\AmazingAdmin.csproj /t:package /p:Configuration=Release
Copy-Item .\src\AmazingAdmin\obj\Release\Package\AmazingAdmin.zip $publishWorkspace
```

Untuk memberi tahu lingkungan Elastic Beanstalk apa yang harus dilakukan dengan semua aplikasi ini, salin manifes dari solusi Anda ke ruang kerja publikasi dan kemudian zip folder.

```
Write-Host 'Copy deployment manifest'
Copy-Item .\aws-windows-deployment-manifest.json $publishWorkspace

Write-Host 'Zipping up publish workspace to create app bundle'
Add-Type -assembly "system.io.compression.filesystem"
[io.compression.zipfile]::CreateFromDirectory( $publishWorkspace, $appBundle)
```

Sekarang setelah Anda memiliki bundel aplikasi, Anda bisa pergi ke konsol web dan mengunggah arsip ke lingkungan Elastic Beanstalk. Sebagai alternatif, Anda dapat terus menggunakanAWSCmdlet PowerShell untuk memperbarui lingkungan Elastic Beanstalk dengan bundel aplikasi. Pastikan Anda telah menetapkan profil dan wilayah saat ini ke profil dan wilayah yang berisi lingkungan Elastic Beanstalk Anda dengan menggunakanSet-AWSCredentialsdanSet-DefaultAWSRegioncmdlet.

```
Write-Host 'Write application bundle to S3'
# Determine S3 bucket to store application bundle
$s3Bucket = New-EBStorageLocation
Write-S3Object -BucketName $s3Bucket -File $appBundle
```



```
$applicationName = "ASPNETCoreOnAWS"  
$environmentName = "ASPNETCoreOnAWS-dev"  
$versionLabel = [System.DateTime]::Now.Ticks.ToString()  
  
Write-Host 'Update Beanstalk environment for new application bundle'  
New-EBApplicationVersion -ApplicationName $applicationName -VersionLabel $versionLabel  
-SourceBundle_S3Bucket $s3Bucket -SourceBundle_S3Key app-bundle.zip  
Update-EBEnvironment -ApplicationName $applicationName -EnvironmentName  
$environmentName -VersionLabel $versionLabel
```

Sekarang, periksa status pembaruan menggunakan halaman status lingkungan Elastic Beanstalk baik di toolkit atau konsol web. Setelah selesai, Anda akan dapat menavigasi ke masing-masing aplikasi yang Anda gunakan di jalur IIS yang ditetapkan dalam manifes penyebaran.

## Menyebarkan ke Amazon EC2 Container Service

### Important

Yang baruPublikasikan keAWSfitur ini dirancang untuk menyederhanakan bagaimana Anda mempublikasikan aplikasi NET untukAWS. Anda mungkin ditanya apakah Anda ingin beralih ke pengalaman penerbitan ini setelah Anda memilihMemublikasikan Kontainer keAWS. Untuk informasi selengkapnya, lihat [Bekerja dengan PublikasikanAWSdi Visual](#).

Layanan Wadah Amazon Elastic adalah layanan manajemen container dengan performa tinggi yang sangat dapat diskalakan dan memungkinkan Anda untuk dengan mudah menjalankan aplikasi pada kluster instans Amazon EC2 yang dikelola.

Untuk men-deploy aplikasi di Amazon Elastic Container Service, komponen aplikasi Anda harus dikembangkan untuk dijalankan di Kontainer Docker. Kontainer Docker adalah unit standar pengembangan perangkat lunak, yang berisi segala sesuatu pada aplikasi perangkat lunak Anda: kode, runtime, alat sistem, perpustakaan sistem, dll.

Toolkit for Visual Studio menyediakan wizard yang menyederhanakan aplikasi penerbitan melalui Amazon ECS. Wizard ini dijelaskan di bagian berikut.

Untuk informasi selengkapnya tentang Amazon ECS, kunjungi[Dokumentasi Layanan Wadah Elastic](#). Ini termasuk ikhtisar[Dasar docker](#)dan[membuat sebuah kluster](#).

## Topik

- [MenentukanAWSKredenal untuk ASP.NET Core 2 Aplikasi](#)
- [Menerapkan Aplikasi ASP.NET Core 2.0 ke Amazon ECS \(Fargate\) \(Legacy\)](#)
- [Menyebarkan Aplikasi ASP.NET Core 2.0 ke Amazon ECS \(EC2\)](#)

## MenentukanAWSKredenal untuk ASP.NET Core 2 Aplikasi

Ada dua jenis kredensyal yang diputar saat Anda menyebarkan aplikasi ke kontainer Docker: kredensyal penyebaran dan kredensyal instans.

Kredensi penyebaran digunakan oleh Publish Container untukAWSwizad untuk menciptakan lingkungan di Amazon ECS. Ini termasuk hal-hal seperti tugas, layanan, peran IAM, repositori kontainer Docker, dan jika Anda memilih, load balancer.

Kredensi instans digunakan oleh instance (termasuk aplikasi Anda) untuk mengakses yang berbedaAWSlayanan. Misalnya, jika aplikasi ASP.NET Core 2.0 Anda membaca dan menulis ke objek Amazon S3, aplikasi tersebut akan memerlukan izin yang sesuai. Anda dapat memberikan kredensyal yang berbeda menggunakan metode yang berbeda berdasarkan lingkungan. Misalnya, aplikasi ASP.NET Core 2 Anda mungkin menargetkanPengembangandanProduksilingkungan. Anda dapat menggunakan instans Docker lokal dan kredensyal untuk pengembangan dan peran yang ditetapkan dalam produksi.

### Menentukan kredenal penyebaran

ParameterAWSakun yang Anda tentukan diPublikasikan ContainerAWSwizad adalahAWSakun wizard akan digunakan untuk penyebaran ke Amazon ECS. Profil akun harus memiliki izin untuk Amazon Elastic Compute Cloud, Amazon Elastic Container Service, danAWS Identity and Access Management.

Jika Anda melihat opsi yang hilang dari daftar drop-down, mungkin karena Anda tidak memiliki izin. Misalnya, jika Anda membuat klaster untuk aplikasi Anda tetapi tidak melihatnya diPublikasikan ContainerAWSHalaman penyihir Cluster. Jika ini terjadi, tambahkan izin yang hilang dan coba wizard lagi.

### Menentukan kredensi instans pengembangan

Untuk lingkungan non-produksi, Anda dapat mengonfigurasi kredensyal di setelan aplikasi. <environment>berkas.json. Misalnya, untuk mengonfigurasi kredensyal Anda di file AppSettings.development.json di Visual Studio 2017:

1. Tambahkan paket `AWSSDK.Extensions.netcore.setup` NuGet ke proyek Anda.
2. Tambahkan `AWSpengaturan` ke `AppSettings.development.json`. Konfigurasi di bawah `setProfile` dan `Region`.

```
{
  "AWS": {
    "Profile": "local-test-profile",
    "Region": "us-west-2"
  }
}
```

## Menentukan kredensi instans produksi

Untuk instans produksi, kami sarankan Anda menggunakan peran IAM untuk mengontrol apa yang dapat diakses aplikasi Anda (dan layanan). Misalnya, untuk mengkonfigurasi peran IAM dengan Amazon ECS sebagai prinsip layanan dengan izin ke Amazon Simple Storage Service dan Amazon DynamoDB dari AWS Management Console:

1. Masuk ke AWS Management Console dan buka konsol IAM di <https://console.aws.amazon.com/iam/>.
2. Di panel navigasi konsol IAM, pilih Peran, dan lalu pilih Buat peran.
3. Pilih AWS Layanan jenis peran, dan kemudian pilih Layanan EC2 Wadah.
4. Pilih Tugas Layanan Kontainer EC2 kasus penggunaan. Kasus penggunaan ditentukan oleh layanan untuk menyertakan kebijakan kepercayaan yang diperlukan layanan. Lalu, pilih Selanjutnya: Izin.
5. Pilih Amazon S3 Full Access dan Amazon DynamoDB Full Access kebijakan izin. Beri tanda pada kotak centang di samping setiap kebijakan, dan kemudian pilih Selanjutnya: Tinjau,
6. Untuk Nama peran, ketik nama peran atau akhiran nama peran untuk membantu Anda mengidentifikasi tujuan peran ini. Nama peran harus unik di akun AWS Anda. Grup tidak dibedakan berdasarkan huruf besar-kecil. Misalnya, Anda tidak dapat membuat peran dengan nama `PRODRole` dan `prodrole`. Anda tidak dapat mengubah nama peran setelah dibuat karena berbagai entitas mungkin mereferensikan peran tersebut.
7. (Opsional) Untuk Deskripsi peran, ketikkan deskripsi untuk peran baru tersebut.
8. Tinjau peran dan kemudian pilih Buat peran.

Anda dapat menggunakan peran ini sebagai peran tugas pada Definisi Tugas ECShalamanPublikasikan ContainerAWSpenyihir.

Untuk informasi selengkapnya, lihat [Menggunakan Peran Berbasis Layanan](#).

## Menerapkan Aplikasi ASP.NET Core 2.0 ke Amazon ECS (Fargate) (Legacy)

### Important

Dokumentasi ini mengacu pada layanan dan fitur lama. Untuk panduan dan konten yang diperbarui, lihat panduan [alat penyebaranAWS .NET](#) dan [Deploying to list](#) ofAWS contents yang diperbarui.

Bagian ini menjelaskan cara menggunakanAWS Wisaya Publish Container to, yang disediakan sebagai bagian dari Toolkit for Visual Studio, untuk menerapkan aplikasi ASP.NET Core 2.0 dalam kontainer yang menargetkan Linux melalui Amazon ECS menggunakan jenis peluncuran Fargate. Karena aplikasi web dimaksudkan untuk berjalan terus menerus, itu akan digunakan sebagai layanan.

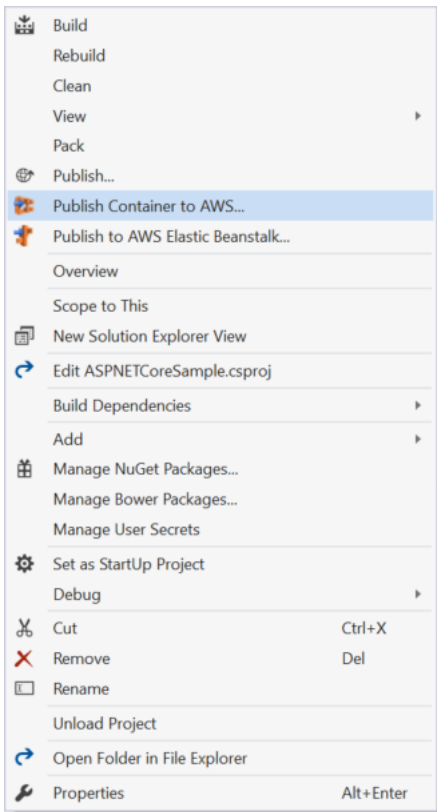
### Sebelum Anda memublikasikan kontainer

Sebelum menggunakan Publish Container untukAWS wizard untuk menyebarkan aplikasi ASP.NET Core 2.0 Anda:

- [TentukanAWS kredensi Anda](#) dan [dapatkan penyiapan dengan Amazon ECS](#).
- [Instal Docker](#). Anda memiliki beberapa opsi instalasi yang berbeda termasuk [Docker untuk Windows](#).
- Di Visual Studio, buat (atau buka) proyek untuk aplikasi kontainer ASP.NET Core 2.0 yang menargetkan Linux.

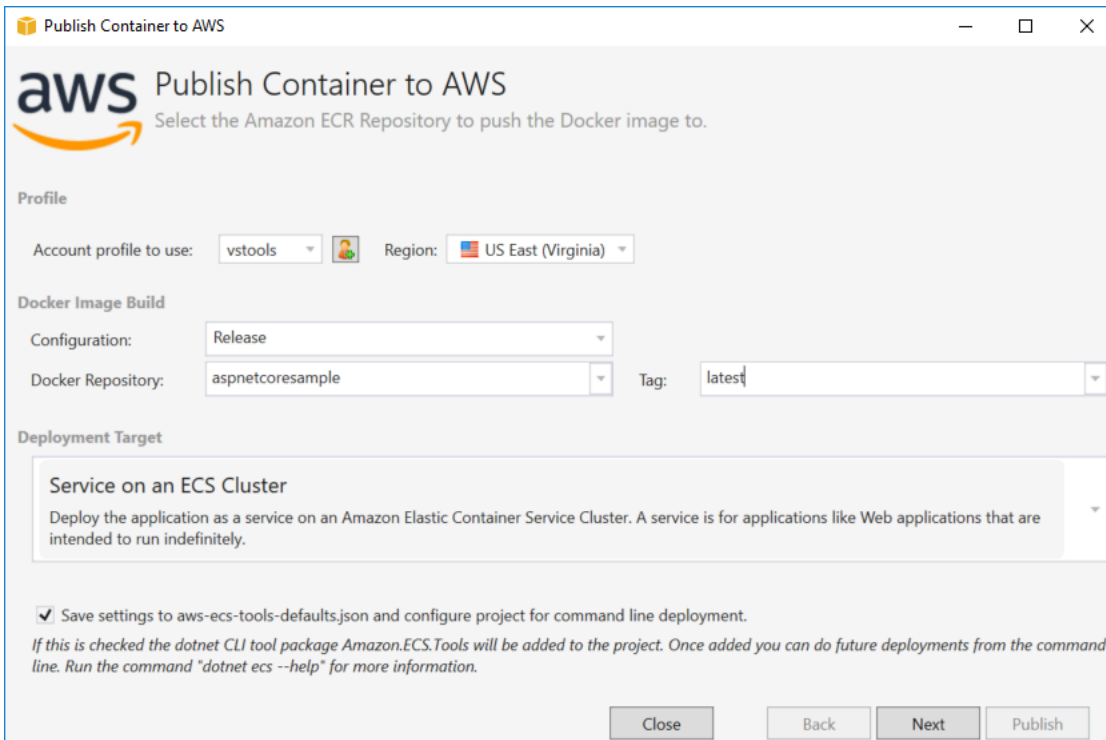
### Mengakses Publish Container keAWS wizard

Untuk menyebarkan aplikasi kontainer ASP.NET Core 2.0 yang menargetkan Linux, klik kanan proyek di Solution Explorer dan pilih Publish Container untukAWS.



Anda juga dapat memilih Publish Container keAWS menu Visual Studio Build.

## Publikasikan Kontainer keAWS Wisaya



Profil akun yang akan digunakan - Pilih profil akun yang akan digunakan.

Wilayah - Pilih wilayah penyebaran. Profil dan wilayah digunakan untuk menyiapkan sumber daya lingkungan penyebaran Anda dan memilih registri Docker default.

Konfigurasi - Pilih konfigurasi build image Docker.

Docker Repository - Pilih repositori Docker yang ada atau ketik nama repositori baru dan itu akan dibuat. Ini adalah repositori kontainer build didorong ke.

Tag - Pilih tag yang ada atau ketik nama tag baru. Tag dapat melacak detail penting seperti versi, opsi, atau elemen konfigurasi unik lainnya dari wadah Docker.

Deployment Target - Pilih Layanan pada Cluster ECS. Gunakan opsi penyebaran ini ketika aplikasi Anda dimaksudkan untuk berjalan lama (seperti aplikasi web ASP.NET).

Simpan pengaturan ke **aws-docker-tools-defaults.json** dan konfigurasi proyek untuk penyebaran baris perintah - Periksa opsi ini jika Anda menginginkan fleksibilitas penerapan dari baris perintah. Gunakan `dotnet ecs deploy` dari direktori proyek Anda untuk menyebarkan `dotnet ecs publish` wadah.

## Halaman Konfigurasi Peluncuran

**Publish Container to AWS**

**Launch Configuration**  
Choose how to provide compute capacity to your application.

ECS Cluster:

*This wizard supports creating an empty cluster which is suitable for running Fargate based services and tasks. It will not have any EC2 instances registered to it so services and tasks with the EC2 launch type will not run. The easiest way to create a cluster with EC2 instances registered is to use the AWS web console.*

Launch Type:

*FARGATE will automatically provision the necessary compute capacity needed to run the application based on the CPU and Memory settings. This removes the need to add any EC2 instances to your cluster.*

**Allocated Compute Capacity**

CPU Maximum (vCPU):  Memory Maximum (GB):

**Network Configuration**

VPC Subnets:  Security Groups:

Assign Public IP Address

**ECS Cluster** - Pilih cluster yang akan menjalankan image Docker Anda. Jika Anda memilih untuk membuat klaster kosong, berikan nama untuk klaster baru Anda.

**Jenis Peluncuran** - Pilih FARGATE.

**CPU Maximum (vCPU)** - Pilih jumlah maksimum kapasitas komputasi yang diperlukan untuk aplikasi Anda. Untuk melihat rentang nilai CPU dan Memori yang diizinkan, lihat [ukuran tugas](#).

**Memory Maximum (GB)** - Pilih jumlah maksimum memori yang tersedia untuk aplikasi Anda.

**Subnet VPC** - Pilih satu atau lebih subnet di bawah satu VPC. Jika Anda memilih lebih dari satu subnet, tugas Anda akan didistribusikan ke seluruh subnet. Hal ini dapat meningkatkan ketersediaan. Untuk informasi selengkapnya, lihat [VPC default dan subnet default](#).

**Grup Keamanan** - Pilih grup keamanan.

Sebuah grup keamanan bertindak sebagai firewall untuk instans Amazon EC2 yang terkait, yang mengontrol lalu lintas masuk maupun keluar di tingkat instans.

[Grup keamanan default](#) dikonfigurasi untuk memungkinkan lalu lintas masuk dari instans yang ditetapkan ke grup keamanan yang sama dan semua lalu lintas IPv4 keluar. Anda perlu outbound diperbolehkan sehingga layanan dapat mencapai repositori kontainer.

**Tetapkan Alamat IP Publik** - Periksa ini untuk membuat tugas Anda dapat diakses dari internet.

## Halaman Konfigurasi Layanan

Publish Container to AWS

**aws** Service Configuration  
Choose the number of instances of the service and how the instances should be deployed.

**Service Parameters**  
*Deploying an application as a service is good for web applications or long lived services. If any of your tasks should fail or stop for any reason, the Amazon ECS service scheduler will launch another instance of your application to replace the failed instance.*

Service:

Number of Tasks:

Minimum Healthy Percent:

Maximum Percent:

Close Back Next Publish

Layanan - Pilih salah satu layanan di drop-down untuk menyebarkan wadah Anda ke layanan yang ada. Atau pilih Buat Baru untuk membuat layanan baru. Nama layanan harus unik dalam sebuah klaster, tetapi Anda dapat memiliki layanan yang bernama sama di beberapa klaster dalam satu Wilayah atau lebih.

Jumlah Tugas - Jumlah tugas untuk diterapkan dan terus berjalan di klaster Anda. Setiap tugas adalah salah satu instance dari wadah Anda.

Persen Sehat Minimum - Persentase tugas yang harus tetap dalam RUNNING keadaan selama penyebaran dibulatkan ke bilangan bulat terdekat.

Persen Maksimum - Persentase tugas yang diizinkan dalam RUNNING atau PENDING negara selama penyebaran dibulatkan ke bilangan bulat terdekat.



## Halaman Application Load Balancer

**Publish Container to AWS**

**aws** Application Load Balancer Configuration

Using an Application Load Balancer allows multiple instances of the application be accessible through a single URL endpoint.

Configure Application Load Balancer

*It is recommended for web applications to use an Application Load Balancer which allows containers to use dynamic host port mapping. This will give the ability to run multiple instances of the web applications on the same container host without contention for port 80.*

Load Balancer:

Listener Port:

**Load Balancer Target Group**

*The Application Load Balancer will send requests to the Target Group if the request matches the specified URL path pattern. Amazon ECS will register all instances of the container with their dynamic port to the Target Group using the provided IAM role for the service.*

Target Group:

Path Pattern:

Health Check Path:

Konfigurasi Application Load Balancer - Periksa untuk mengkonfigurasi penyeimbang beban aplikasi.

Load Balancer - Pilih load balancer yang ada atau pilih Create New dan ketik nama untuk load balancer baru.

Port Pendengar - Pilih port pendengar yang ada atau pilih Buat Baru dan ketik nomor port. Default, port80, sesuai untuk sebagian besar aplikasi web.

Grup Target - Pilih grup target Amazon ECS akan mendaftarkan tugas ke layanan.

Path Pattern - Load balancer akan menggunakan routing berbasis jalur. Terima default/ atau berikan pola yang berbeda. Pola jalur peka huruf besar-kecil, dapat memiliki panjang hingga 128 karakter, dan berisi [satu set karakter tertentu](#).

Health Check Path - Jalur Ping yang merupakan tujuan pada target untuk pemeriksaan kesehatan. Secara default, itu adalah /. Masukkan jalur yang berbeda jika diperlukan. Jika jalur yang Anda masukkan tidak valid, pemeriksaan kesehatan akan gagal dan akan dianggap tidak sehat.

Jika Anda menerapkan beberapa layanan, dan setiap layanan akan diterapkan ke jalur atau lokasi yang berbeda, Anda akan memerlukan jalur pemeriksaan khusus.

## Halaman Definisi Tugas

**Task Definition**  
Task Definition defines the parameters for how the application will run within its Docker container.

Task Definition:

Container:

Permissions

Task Role:

Select an IAM role to provide AWS credentials to your application to access AWS Services.

Task Execution Role:

Fargate requires a role to pull private images and publish logs on your behalf.

Port Mapping

Container Port
80

Environment Variables

Variable	Value
ASPNETCORE_ENVIRONMENT	Production

Buttons: Close, Back, Next, Publish

Definisi Tugas - Pilih definisi tugas yang ada atau pilih Buat Baru dan ketik nama definisi tugas baru.

Container - Pilih wadah yang ada atau pilih Create New dan ketik nama kontainer baru.

Peran Tugas - Pilih peran IAM yang memiliki kredensi yang dibutuhkan aplikasi Anda untuk mengakses AWS Layanan. Ini adalah bagaimana kredensial diteruskan ke aplikasi Anda. Lihat [cara menentukan kredensi AWS keamanan untuk aplikasi Anda](#).

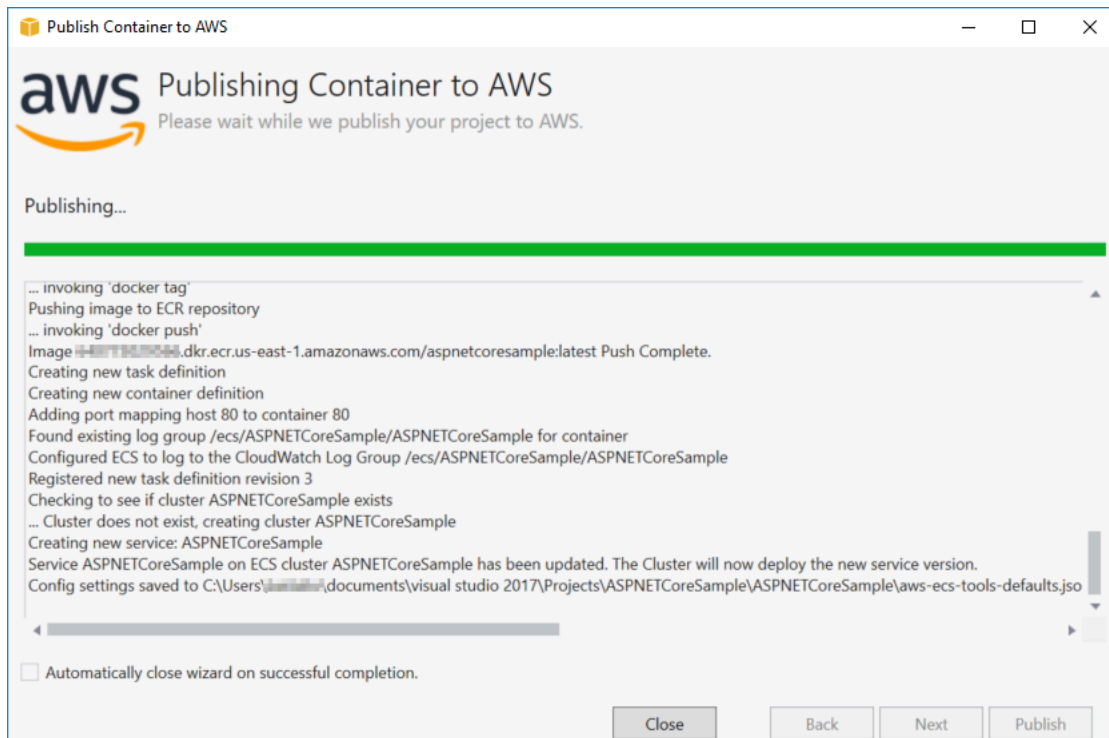
Peran Eksekusi Tugas - Pilih peran dengan izin untuk menarik gambar pribadi dan mempublikasikan log. AWS Fargate akan menggunakannya atas nama Anda.

Pemetaan Port - Pilih nomor port pada kontainer yang terikat ke port host yang ditugaskan secara otomatis.

Variabel Lingkungan - Menambahkan, memodifikasi, atau menghapus variabel lingkungan untuk wadah. Anda dapat memodifikasinya agar sesuai dengan penyebaran Anda.

Ketika Anda puas dengan konfigurasi, klik Publikasikan untuk memulai proses penyebaran.

## Publishing Container keAWS



Acara ditampilkan selama deployment. Wizard secara otomatis ditutup pada penyelesaian yang berhasil. Anda dapat mengganti ini dengan menghapus centang kotak di bagian bawah halaman.

Anda dapat menemukan URL instance baru Anda diAWS Explorer. Perluas Amazon ECS dan Cluster, lalu klik klaster Anda.

## Menyebarkan Aplikasi ASP.NET Core 2.0 ke Amazon ECS (EC2)

Bagian ini menjelaskan cara menggunakanPublikasikan KontainerAWSwizard, disediakan sebagai bagian dari Toolkit for Visual Studio, untuk menyebarkan aplikasi ASP.NET Core 2.0 berisi yang menargetkan Linux melalui Amazon ECS menggunakan jenis peluncuran EC2. Karena aplikasi web dimaksudkan berjalan terus menerus, itu akan digunakan sebagai layanan.

### Sebelum memublikasikan kontainer

Sebelum menggunakanPublikasikan KontainerAWSuntuk menyebarkan aplikasi ASP.NET Core 2.0 Anda:

- [TentukanAWSkredensial](#)dan[dapatkan pengaturan dengan Amazon ECS](#).
- [Instal Docker](#). Anda memiliki beberapa opsi instalasi yang berbeda termasuk[Docker untuk Windows](#).

- [Membuat kluster Amazon ECS](#) berdasarkan kebutuhan aplikasi web Anda. Hanya butuh beberapa langkah.
- Di Visual Studio, buat (atau buka) proyek untuk aplikasi kontainer ASP.NET Core 2.0 yang menargetkan Linux.

## Mengakses Container Publish keAWSpenyihir

Untuk menyebarkan aplikasi kontainer ASP.NET Core 2.0 yang menargetkan Linux, klik kanan proyek di Solution Explorer dan pilih **Publikasikan KontainerAWS**.

Anda juga dapat memilih **Publikasikan KontainerAWS** pada menu Visual Studio Build.

## Publikasikan KontainerAWSPenyihir

Profil akun yang akan digunakan- Pilih profil akun yang akan digunakan.

Wilayah- Pilih wilayah penyebaran. Profil dan wilayah digunakan untuk mengatur sumber daya lingkungan penyebaran Anda dan memilih registri Docker default.

Konfigurasi- Pilih konfigurasi build gambar Docker.

Repositori Docker- Pilih repositori Docker yang ada atau ketik nama repositori baru dan akan dibuat. Ini adalah repositori gambar kontainer dibangun didorong ke.

Tag- Pilih tag yang ada atau ketik nama tag baru. Tag dapat melacak rincian penting seperti versi, opsi atau elemen konfigurasi unik lainnya dari wadah Docker.

Penerapan- Pilih Layanan di Cluster ECS. Gunakan opsi penyebaran ini saat aplikasi Anda dimaksudkan untuk berjalan lama (seperti aplikasi web ASP.NET Core 2.0).

Menyimpan pengaturan **aws-docker-tools-defaults.json** dan konfigurasi proyek untuk penyebaran baris perintah- Periksa opsi ini jika Anda ingin fleksibilitas penyebaran dari baris perintah. Gunakan `dotnet ecs deploy` dari direktori proyek Anda untuk menyebarkan `dotnet ecs publish` wadah.

## Halaman Konfigurasi peluncuran

Kluster ECS- Pilih cluster yang akan menjalankan gambar Docker Anda. Anda dapat [membuat kluster ECS](#) menggunakan AWSKonsol Manajemen.

Tipe peluncuran- Pilih EC2. Untuk menggunakan tipe peluncuran Fargate, lihat [Menyebarkan Aplikasi ASP.NET Core 2.0 ke Amazon ECS \(Fargate\)](#).

## Halaman Konfigurasi Layanan

Layanan- Pilih salah satu layanan di drop-down untuk menyebarkan kontainer Anda ke layanan yang ada. Atau pilih **Membuat Baru** untuk membuat layanan baru. Nama layanan harus unik dalam kluster, tetapi Anda dapat memiliki layanan yang sama di beberapa kluster dalam satu wilayah atau di beberapa wilayah.

Jumlah Tugas- Jumlah tugas untuk digunakan dan terus berjalan di kluster Anda. Setiap tugas adalah salah satu contoh dari wadah Anda.

Persen Sehat Minimum- Persentase tugas yang harus tetap di **RUNNING** negara selama penyebaran dibulatkan ke atas ke atas ke bilangan bulat terdekat.

Persen maksimum- Persentase tugas yang diperbolehkan dalam **RUNNING** atau **PENDING** negara selama penyebaran dibulatkan ke integer terdekat.

Template penempatan- Pilih template penempatan tugas.

Ketika Anda meluncurkan tugas ke dalam kluster, Amazon ECS harus menentukan tempat menempatkan tugas berdasarkan persyaratan yang ditentukan dalam ketentuan tugas, seperti CPU dan memori. Demikian pula, saat Anda menurunkan skala jumlah tugas, Amazon ECS harus menentukan tugas mana yang harus diakhiri.

Template penempatan mengontrol bagaimana tugas diluncurkan ke dalam kluster:

- Penyebaran Seimbang AZ - mendistribusikan tugas di seluruh Availability Zone dan di seluruh instans kontainer.
- BinPack AZ - mendistribusikan tugas di seluruh Availability Zone dan di seluruh instans kontainer dengan memori paling sedikit.
- BinPack - mendistribusikan tugas berdasarkan jumlah CPU atau memori yang paling sedikit.
- Satu Tugas Per Host - tempat, paling banyak, satu tugas dari layanan pada setiap instans kontainer.

Untuk informasi selengkapnya, lihat [Penempatan Tugas Amazon ECS](#).

## Halaman Application Load Balancer

Konfigurasi Application Load Balancer- Periksa untuk mengkonfigurasi penyeimbang beban aplikasi.

Pilih peran IAM untuk layanan- Pilih peran yang ada atau pilihMembuat Barudan peran baru akan dibuat.

Penyeimbang Beban- Pilih penyeimbang beban yang ada atau pilihMembuat Barudan ketik nama untuk penyeimbang beban baru.

Port listener- Pilih port pendengar yang ada atau pilihMembuat Barudan ketik nomor port. Default, port80, cocok untuk sebagian besar aplikasi web.

Grup target- Secara default, penyeimbang beban mengirimkan permintaan ke target yang terdaftar menggunakan port dan protokol yang Anda tentukan untuk grup target. Anda dapat mengganti port ini ketika Anda mendaftarkan setiap target dengan kelompok target.

Pola Jalan- Penyeimbang beban akan menggunakan routing berbasis jalur. Menerima default/atau memberikan pola yang berbeda. Pola jalur peka huruf besar/kecil, panjangnya bisa hingga 128 karakter, dan berisi[pilih set karakter](#).

Jalur Health- Jalur ping yang merupakan tujuan pada target pemeriksaan kesehatan. Secara default, itu adalah/dan sesuai untuk aplikasi web. Masukkan jalur yang berbeda jika diperlukan. Jika jalur yang Anda masukkan tidak valid, pemeriksaan kesehatan akan gagal dan akan dianggap tidak sehat.

Jika Anda menerapkan beberapa layanan, dan setiap layanan akan digunakan ke jalur atau lokasi yang berbeda, Anda mungkin memerlukan jalur pemeriksaan khusus.

## Halaman Definisi Tugas ECS

Definisi tugas- Pilih ketentuan tugas yang ada atau pilihMembuat Barudan ketik nama definisi tugas baru.

Kontainer- Pilih wadah yang ada atau pilihMembuat Barudan ketik nama kontainer baru.

Memori (MiB)- Memberikan nilai untukBatas lunakatauBatas kerasatau keduanya.

Parameterbatas lunak(dalam MiB) memori untuk cadangan kontainer. Docker mencoba untuk menjaga memori kontainer di bawah batas lunak. Kontainer dapat menggunakan lebih banyak memori, hingga batas keras ditentukan dengan parameter memory (jika berlaku), atau semua memori yang tersedia pada instans kontainer, mana yang lebih dulu.

Parameterbatas keras(dalam MiB) memori yang akan ditampilkan ke kontainer. Jika kontainer Anda mencoba untuk melebihi memori yang ditentukan di sini, kontainer akan dimatikan.

Peran tugas- Pilih peran tugas untuk peran IAM yang memungkinkan izin kontainer untuk memanggilAWSAPI yang ditentukan dalam kebijakan yang terkait atas nama Anda. Ini adalah bagaimana kredensi diteruskan ke aplikasi Anda. Lihat[bagaimana menentukanAWSkredensi keamanan untuk aplikasi Anda](#).

Pemetaan Port- Menambahkan, mengubah atau menghapus pemetaan port untuk kontainer. Jika load balancer aktif, port host akan default ke 0 dan port assignment akan dinamis.

Variabel Lingkungan- Menambahkan, mengubah, atau menghapus variabel lingkungan untuk kontainer.

Jika Anda puas dengan konfigurasinya, klikPublikasikanuntuk memulai proses deployment.

## Publikasikan KontainerAWS

Peristiwa ditampilkan selama deployment. Wizard secara otomatis ditutup pada penyelesaian yang berhasil. Anda dapat mengganti ini dengan menghapus centang pada kotak di bagian bawah halaman.

Anda dapat menemukan URL instans baru Anda diAWSExplorer. Perluas Amazon ECS dan Cluster, lalu klik klaster Anda.

# Memecahkan masalah AWS Toolkit for Visual Studio

Bagian berikut berisi informasi pemecahan masalah umum tentang AWS Toolkit for Visual Studio dan bekerja dengan AWS layanan dari toolkit.

## Note

Informasi set-up-specific penginstalan dan pemecahan masalah tersedia di topik [Masalah penginstalan Pemecahan Masalah](#), yang terdapat di Panduan Pengguna ini.

## Topik

- [Memecahkan masalah praktik terbaik](#)
- [Amazon CodeWhisperer Masuk dan Keluar dinonaktifkan](#)

## Memecahkan masalah praktik terbaik

Berikut ini adalah praktik terbaik yang disarankan saat memecahkan masalah AWS Toolkit for Visual Studio .

- 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 Toolkit. Untuk penjelasan rinci tentang cara menemukan log AWS Toolkit Anda, lihat prosedur [Cara menemukan AWS log Anda](#), yang terletak di topik panduan ini.
- Periksa permintaan terbuka, solusi yang diketahui, atau laporkan masalah Anda yang belum terselesaikan di bagian [AWS Toolkit for Visual Studio Masalah](#) di AWS Toolkit for Visual Studio GitHub repositori.

### Cara menemukan log AWS Toolkit Anda

1. Dari menu utama Visual Studio, perluas Ekstensi.
2. Pilih AWS Toolkit untuk memperluas menu AWS Toolkit, lalu pilih View Toolkit Logs.
3. Saat folder log AWS Toolkit terbuka di Sistem Operasi Anda, urutkan file berdasarkan tanggal dan temukan file log apa pun yang berisi informasi yang relevan dengan masalah Anda saat ini.



## Amazon CodeWhisperer Masuk dan Keluar dinonaktifkan

Jika Anda mengalami masalah dengan CodeWhisperer layanan di mana kedua item menu Masuk dan Keluar dinonaktifkan, selesaikan masalah dengan menyelesaikan langkah-langkah berikut.

1. Dari Windows File Explorer, arahkan ke folder cache AWS Toolkit yang terletak di:`%LOCALAPPDATA%/aws/toolkits/language-servers/CodeWhisperer`.
2. Kosongkan isi folder cache.
3. Tutup dan buka kembali solusi saat ini.

# Keamanan untuk AWS Toolkit for Visual Studio

Keamanan cloud di Amazon Web Services (AWS) merupakan prioritas tertinggi. Sebagai seorang pelanggan AWS, Anda mendapatkan manfaat dari pusat data dan arsitektur jaringan yang dibangun untuk memenuhi persyaratan dari organisasi yang paling sensitif terhadap keamanan. Keamanan adalah tanggung jawab bersama antara Anda AWS dan Anda. [Model Tanggung Jawab Bersama](#) menggambarkan ini sebagai Keamanan dari Cloud dan Keamanan dalam Cloud.

Security of the Cloud - AWS bertanggung jawab untuk melindungi infrastruktur yang menjalankan semua layanan yang ditawarkan di AWS Cloud dan memberi Anda layanan yang dapat Anda gunakan dengan aman. Tanggung jawab keamanan kami adalah prioritas tertinggi di AWS, dan efektivitas keamanan kami secara teratur diuji dan diverifikasi oleh auditor pihak ketiga sebagai bagian dari [Program AWS Kepatuhan](#).

Keamanan di Cloud — Tanggung jawab Anda ditentukan oleh AWS layanan yang Anda gunakan, dan faktor-faktor lain termasuk sensitivitas data Anda, persyaratan organisasi Anda, serta undang-undang dan peraturan yang berlaku.

AWS Produk atau layanan ini mengikuti [model tanggung jawab bersama](#) melalui layanan Amazon Web Services (AWS) tertentu yang didukungnya. Untuk informasi keamanan AWS layanan, lihat [halaman dokumentasi keamanan AWS layanan](#) dan [AWS layanan yang berada dalam lingkup upaya AWS kepatuhan oleh program kepatuhan](#).

## Topik

- [Perlindungan Data di AWS Toolkit for Visual Studio](#)
- [Identity and Access Management](#)
- [Validasi Kepatuhan untuk AWS Produk atau Layanan ini](#)
- [Ketahanan untuk AWS Produk atau Layanan ini](#)
- [Keamanan Infrastruktur untuk AWS Produk atau Layanan ini](#)
- [Konfigurasi dan Analisis Kerentanan di AWS Toolkit for Visual Studio](#)

## Perlindungan Data di AWS Toolkit for Visual Studio

[Model tanggung jawab AWS bersama model](#) berlaku untuk perlindungan data di AWS Toolkit for Visual Studio. 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 kredensi dan mengatur pengguna individu dengan AWS IAM Identity Center atau AWS Identity and Access Management (IAM). Dengan cara itu, setiap pengguna hanya diberi izin yang diperlukan untuk memenuhi tanggung jawab tugasnya. Kami juga menyarankan supaya Anda mengamankan data dengan cara-cara berikut:

- Gunakan 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.
- Gunakan solusi AWS enkripsi, bersama dengan semua kontrol keamanan default di dalamnya Layanan AWS.
- Gunakan layanan keamanan terkelola lanjut seperti Amazon Macie, yang membantu menemukan dan mengamankan data sensitif yang disimpan di Amazon S3.
- Jika Anda memerlukan modul kriptografi tervalidasi FIPS 140-2 saat mengakses AWS melalui antarmuka baris perintah atau API, gunakan titik akhir FIPS. Lihat informasi yang lebih lengkap tentang titik akhir FIPS yang tersedia di [Standar Pemrosesan Informasi Federal \(FIPS\) 140-2](#).

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 Toolkit for Visual Studio atau Layanan AWS lainnya menggunakan konsol, API AWS CLI, AWS atau SDK. 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.

# Identity and Access Management

AWS Identity and Access Management (IAM) adalah Layanan AWS yang membantu administrator mengontrol akses ke AWS sumber daya dengan aman. Administrator IAM mengontrol siapa yang dapat diautentikasi (masuk) dan diberi wewenang (memiliki izin) untuk menggunakan sumber daya. AWS IAM adalah Layanan AWS yang dapat Anda gunakan tanpa biaya tambahan.

Topik

- [Audiens](#)
- [Mengautentikasi dengan identitas](#)
- [Mengelola akses menggunakan kebijakan](#)
- [Bagaimana Layanan AWS bekerja dengan IAM](#)
- [Memecahkan masalah AWS identitas dan akses](#)

## Audiens

Cara Anda menggunakan AWS Identity and Access Management (IAM) berbeda, tergantung pada pekerjaan yang Anda lakukan. AWS

**Pengguna layanan** — Jika Anda menggunakan Layanan AWS untuk melakukan pekerjaan Anda, maka administrator Anda memberi Anda kredensi dan izin yang Anda butuhkan. Saat Anda menggunakan lebih banyak AWS fitur untuk melakukan pekerjaan Anda, Anda mungkin memerlukan izin tambahan. Memahami cara akses dikelola dapat membantu Anda meminta izin yang tepat dari administrator Anda. Jika Anda tidak dapat mengakses fitur AWS, lihat [Memecahkan masalah AWS identitas dan akses](#) atau panduan pengguna yang Layanan AWS Anda gunakan.

**Administrator layanan** — Jika Anda bertanggung jawab atas AWS sumber daya di perusahaan Anda, Anda mungkin memiliki akses penuh ke AWS. Tugas Anda adalah menentukan AWS fitur dan sumber daya mana yang harus diakses pengguna layanan Anda. Kemudian, Anda harus mengirimkan permintaan kepada administrator IAM untuk mengubah izin pengguna layanan Anda. Tinjau informasi di halaman ini untuk memahami konsep Basic IAM. Untuk mempelajari lebih lanjut tentang bagaimana perusahaan Anda dapat menggunakan IAM AWS, lihat panduan pengguna yang Layanan AWS Anda gunakan.

**Administrator IAM** – Jika Anda adalah administrator IAM, Anda mungkin ingin belajar dengan lebih detail tentang cara Anda menulis kebijakan untuk mengelola akses ke AWS. Untuk melihat contoh

kebijakan AWS berbasis identitas yang dapat Anda gunakan di IAM, lihat panduan pengguna yang Anda gunakan. Layanan AWS

## Mengautentikasi dengan identitas

Otentikasi adalah cara Anda masuk AWS menggunakan kredensial identitas Anda. Anda harus diautentikasi (masuk ke AWS) sebagai Pengguna root akun AWS, sebagai pengguna IAM, atau dengan mengasumsikan peran IAM.

Anda dapat masuk AWS sebagai identitas federasi dengan menggunakan kredensi yang disediakan melalui sumber identitas. AWS IAM Identity Center Pengguna (IAM Identity Center), autentikasi masuk tunggal perusahaan Anda, dan kredensi Google atau Facebook Anda adalah contoh identitas federasi. Saat Anda masuk sebagai identitas terfederasi, administrator Anda sebelumnya menyiapkan federasi identitas menggunakan peran IAM. Ketika Anda mengakses AWS dengan menggunakan federasi, Anda secara tidak langsung mengambil peran.

Bergantung pada jenis pengguna Anda, Anda dapat masuk ke AWS Management Console atau portal AWS akses. Untuk informasi selengkapnya tentang masuk AWS, lihat [Cara masuk ke Panduan AWS Sign-In Pengguna Anda Akun AWS](#).

Jika Anda mengakses AWS secara terprogram, AWS sediakan kit pengembangan perangkat lunak (SDK) dan antarmuka baris perintah (CLI) untuk menandatangani permintaan Anda secara kriptografis dengan menggunakan kredensial Anda. Jika Anda tidak menggunakan AWS alat, Anda harus menandatangani permintaan sendiri. Untuk informasi selengkapnya tentang penggunaan metode yang disarankan untuk menandatangani permintaan sendiri, lihat [Menandatangani permintaan AWS API](#) di Panduan Pengguna IAM.

Apa pun metode autentikasi yang digunakan, Anda mungkin diminta untuk menyediakan informasi keamanan tambahan. Misalnya, AWS merekomendasikan agar Anda menggunakan otentikasi multi-faktor (MFA) untuk meningkatkan keamanan akun Anda. Untuk mempelajari selengkapnya, lihat [Autentikasi multi-faktor](#) dalam Panduan Pengguna AWS IAM Identity Center dan [Menggunakan autentikasi multi-faktor \(MFA\) dalam AWS](#) dalam Panduan Pengguna IAM.

## Akun AWS pengguna root

Saat Anda membuat Akun AWS, Anda mulai dengan satu identitas masuk yang memiliki akses lengkap ke semua Layanan AWS dan sumber daya di akun. Identitas ini disebut pengguna Akun AWS root dan diakses dengan masuk dengan alamat email dan kata sandi yang Anda gunakan untuk membuat akun. Kami sangat menyarankan agar Anda tidak menggunakan pengguna root untuk tugas sehari-hari. Lindungi kredensial pengguna root Anda dan gunakan kredensial tersebut

untuk melakukan tugas yang hanya dapat dilakukan pengguna root. Untuk daftar lengkap tugas yang mengharuskan Anda masuk sebagai pengguna root, lihat [Tugas yang memerlukan kredensial pengguna root](#) dalam Panduan Pengguna IAM.

## Identitas gabungan

Sebagai praktik terbaik, mewajibkan pengguna manusia, termasuk pengguna yang memerlukan akses administrator, untuk menggunakan federasi dengan penyedia identitas untuk mengakses Layanan AWS dengan menggunakan kredensi sementara.

Identitas federasi adalah pengguna dari direktori pengguna perusahaan Anda, penyedia identitas web, direktori Pusat Identitas AWS Directory Service, atau pengguna mana pun yang mengakses Layanan AWS dengan menggunakan kredensial yang disediakan melalui sumber identitas. Ketika identitas federasi mengakses Akun AWS, mereka mengambil peran, dan peran memberikan kredensi sementara.

Untuk manajemen akses terpusat, kami sarankan Anda menggunakan AWS IAM Identity Center. Anda dapat membuat pengguna dan grup di Pusat Identitas IAM, atau Anda dapat menghubungkan dan menyinkronkan ke sekumpulan pengguna dan grup di sumber identitas Anda sendiri untuk digunakan di semua aplikasi Akun AWS dan aplikasi Anda. Untuk informasi tentang Pusat Identitas IAM, lihat [Apakah itu Pusat Identitas IAM?](#) dalam Panduan Pengguna AWS IAM Identity Center .

## Pengguna dan grup IAM

[Pengguna IAM](#) adalah identitas dalam diri Anda Akun AWS yang memiliki izin khusus untuk satu orang atau aplikasi. Jika memungkinkan, kami merekomendasikan untuk mengandalkan kredensial sementara, bukan membuat pengguna IAM yang memiliki kredensial jangka panjang seperti kata sandi dan kunci akses. Namun, jika Anda memiliki kasus penggunaan tertentu yang memerlukan kredensial jangka panjang dengan pengguna IAM, kami merekomendasikan Anda merotasi kunci akses. Untuk informasi selengkapnya, lihat [Merotasi kunci akses secara teratur untuk kasus penggunaan yang memerlukan kredensial jangka panjang](#) dalam Panduan Pengguna IAM.

[Grup IAM](#) adalah identitas yang menentukan sekumpulan pengguna IAM. Anda tidak dapat masuk sebagai grup. Anda dapat menggunakan grup untuk menentukan izin bagi beberapa pengguna sekaligus. Grup mempermudah manajemen izin untuk sejumlah besar pengguna sekaligus. Misalnya, Anda dapat memiliki grup yang bernama IAMAdmins dan memberikan izin ke grup tersebut untuk mengelola sumber daya IAM.

Pengguna berbeda dari peran. Pengguna secara unik terkait dengan satu orang atau aplikasi, tetapi peran dimaksudkan untuk dapat digunakan oleh siapa pun yang membutuhkannya. Pengguna

memiliki kredensial jangka panjang permanen, tetapi peran memberikan kredensial sementara. Untuk mempelajari selengkapnya, lihat [Kapan harus membuat pengguna IAM \(bukan peran\)](#) dalam Panduan Pengguna IAM.

## Peran IAM

[Peran IAM](#) adalah identitas dalam diri Anda Akun AWS yang memiliki izin khusus. Peran ini mirip dengan pengguna IAM, tetapi tidak terkait dengan orang tertentu. Anda dapat mengambil peran IAM untuk sementara AWS Management Console dengan [beralih peran](#). Anda dapat mengambil peran dengan memanggil operasi AWS CLI atau AWS API atau dengan menggunakan URL kustom. Untuk informasi selengkapnya tentang cara menggunakan peran, lihat [Menggunakan peran IAM](#) dalam Panduan Pengguna IAM.

Peran IAM dengan kredensial sementara berguna dalam situasi berikut:

- Akses pengguna terfederasi – Untuk menetapkan izin ke identitas terfederasi, Anda membuat peran dan menentukan izin untuk peran tersebut. Ketika identitas terfederasi mengautentikasi, identitas tersebut terhubung dengan peran dan diberi izin yang ditentukan oleh peran. Untuk informasi tentang peran untuk federasi, lihat [Membuat peran untuk Penyedia Identitas pihak ketiga](#) dalam Panduan Pengguna IAM. Jika menggunakan Pusat Identitas IAM, Anda harus mengonfigurasi set izin. Untuk mengontrol apa yang dapat diakses identitas Anda setelah identitas tersebut diautentikasi, Pusat Identitas IAM akan mengorelasikan set izin ke peran dalam IAM. Untuk informasi tentang set izin, lihat [Set izin](#) dalam Panduan Pengguna AWS IAM Identity Center .
- Izin pengguna IAM sementara – Pengguna atau peran IAM dapat mengambil peran IAM guna mendapatkan berbagai izin secara sementara untuk tugas tertentu.
- Akses lintas akun – Anda dapat menggunakan peran IAM untuk mengizinkan seseorang (prinsipal tepercaya) di akun lain untuk mengakses sumber daya di akun Anda. Peran adalah cara utama untuk memberikan akses lintas akun. Namun, dengan beberapa Layanan AWS, Anda dapat melampirkan kebijakan secara langsung ke sumber daya (alih-alih menggunakan peran sebagai proxy). Untuk mempelajari perbedaan antara peran dan kebijakan berbasis sumber daya untuk akses lintas akun, lihat [Bagaimana peran IAM berbeda dari kebijakan berbasis sumber daya](#) dalam Panduan Pengguna IAM.
- Akses lintas layanan — Beberapa Layanan AWS menggunakan fitur lain Layanan AWS. Sebagai contoh, ketika Anda memanggil suatu layanan, biasanya layanan tersebut menjalankan aplikasi di Amazon EC2 atau menyimpan objek di Amazon S3. Sebuah layanan mungkin melakukannya menggunakan izin prinsipal yang memanggil, menggunakan peran layanan, atau peran terkait layanan.

- Sesi akses teruskan (FAS) — Saat Anda menggunakan pengguna atau peran IAM untuk melakukan tindakan AWS, Anda dianggap sebagai prinsipal. Ketika Anda menggunakan beberapa layanan, Anda mungkin melakukan sebuah tindakan yang kemudian menginisiasi tindakan lain di layanan yang berbeda. FAS menggunakan izin dari pemanggilan utama Layanan AWS, dikombinasikan dengan permintaan Layanan AWS untuk membuat permintaan ke layanan hilir. Permintaan FAS hanya dibuat ketika layanan menerima permintaan yang memerlukan interaksi dengan orang lain Layanan AWS atau sumber daya untuk menyelesaikannya. Dalam hal ini, Anda harus memiliki izin untuk melakukan kedua tindakan tersebut. Untuk detail kebijakan ketika mengajukan permintaan FAS, lihat [Sesi akses maju](#).
- Peran layanan – Peran layanan adalah [peran IAM](#) yang dijalankan oleh layanan untuk melakukan tindakan atas nama Anda. Administrator IAM dapat membuat, mengubah, dan menghapus peran layanan dari dalam IAM. Untuk informasi selengkapnya, lihat [Membuat sebuah peran untuk mendelegasikan izin ke Layanan AWS](#) dalam Panduan pengguna IAM.
- Peran terkait layanan — Peran terkait layanan adalah jenis peran layanan yang ditautkan ke peran layanan. Layanan AWS Layanan tersebut dapat menjalankan peran untuk melakukan tindakan atas nama Anda. Peran terkait layanan muncul di Anda Akun AWS dan dimiliki oleh layanan. Administrator IAM dapat melihat, tetapi tidak dapat mengedit izin untuk peran terkait layanan.
- Aplikasi yang berjalan di Amazon EC2 — Anda dapat menggunakan peran IAM untuk mengelola kredensi sementara untuk aplikasi yang berjalan pada instans EC2 dan membuat atau permintaan API. AWS CLI AWS Cara ini lebih dianjurkan daripada menyimpan kunci akses dalam instans EC2. Untuk menetapkan AWS peran ke instans EC2 dan membuatnya tersedia untuk semua aplikasinya, Anda membuat profil instance yang dilampirkan ke instance. Profil instans berisi peran dan memungkinkan program yang berjalan di instans EC2 mendapatkan kredensial sementara. Untuk informasi selengkapnya, lihat [Menggunakan peran IAM untuk memberikan izin ke aplikasi yang berjalan dalam instans Amazon EC2](#) dalam Panduan Pengguna IAM.

Untuk mempelajari apakah kita harus menggunakan peran IAM atau pengguna IAM, lihat [Kapan harus membuat peran IAM \(bukan pengguna\)](#) dalam Panduan Pengguna IAM.

## Mengelola akses menggunakan kebijakan

Anda mengontrol akses AWS dengan membuat kebijakan dan melampirkannya ke AWS identitas atau sumber daya. Kebijakan adalah objek AWS yang, ketika dikaitkan dengan identitas atau sumber daya, menentukan izinnya. AWS mengevaluasi kebijakan ini ketika prinsipal (pengguna, pengguna root, atau sesi peran) membuat permintaan. Izin dalam kebijakan menentukan apakah permintaan



diizinkan atau ditolak. Sebagian besar kebijakan disimpan AWS sebagai dokumen JSON. Untuk informasi selengkapnya tentang struktur dan isi dokumen kebijakan JSON, lihat [Gambaran umum kebijakan JSON](#) dalam Panduan Pengguna IAM.

Administrator dapat menggunakan kebijakan AWS JSON untuk menentukan siapa yang memiliki akses ke apa. Artinya, prinsipal manakah yang dapat melakukan tindakan pada sumber daya apa, dan dengan kondisi apa.

Secara default, pengguna dan peran tidak memiliki izin. Untuk memberikan izin kepada pengguna untuk melakukan tindakan di sumber daya yang mereka perlukan, administrator IAM dapat membuat kebijakan IAM. Administrator kemudian dapat menambahkan kebijakan IAM ke peran, dan pengguna dapat mengambil peran.

Kebijakan IAM mendefinisikan izin untuk suatu tindakan terlepas dari metode yang Anda gunakan untuk melakukan operasinya. Misalnya, anggaplah Anda memiliki kebijakan yang mengizinkan tindakan `iam:GetRole`. Pengguna dengan kebijakan tersebut bisa mendapatkan informasi peran dari AWS Management Console, API AWS CLI, atau AWS API.

## Kebijakan berbasis identitas

Kebijakan berbasis identitas adalah dokumen kebijakan izin JSON yang dapat Anda lampirkan ke sebuah identitas, seperti pengguna IAM, grup pengguna IAM, atau peran IAM. Kebijakan ini mengontrol jenis tindakan yang dapat dilakukan oleh pengguna dan peran, di sumber daya mana, dan berdasarkan kondisi seperti apa. Untuk mempelajari cara membuat kebijakan berbasis identitas, lihat [Membuat kebijakan IAM](#) dalam Panduan Pengguna IAM.

Kebijakan berbasis identitas dapat dikategorikan lebih lanjut sebagai kebijakan inline atau kebijakan yang dikelola. Kebijakan inline disematkan langsung ke satu pengguna, grup, atau peran. Kebijakan terkelola adalah kebijakan mandiri yang dapat Anda lampirkan ke beberapa pengguna, grup, dan peran dalam Akun AWS. Kebijakan AWS terkelola mencakup kebijakan terkelola dan kebijakan yang dikelola pelanggan. Untuk mempelajari cara memilih antara kebijakan yang dikelola atau kebijakan inline, lihat [Memilih antara kebijakan yang dikelola dan kebijakan inline](#) dalam Panduan Pengguna IAM.

## Kebijakan berbasis sumber daya

Kebijakan berbasis sumber daya adalah dokumen kebijakan JSON yang Anda lampirkan ke sumber daya. Contoh kebijakan berbasis sumber daya adalah kebijakan kepercayaan peran IAM dan kebijakan bucket Amazon S3. Dalam layanan yang mendukung kebijakan berbasis sumber daya,

administrator layanan dapat menggunakannya untuk mengontrol akses ke sumber daya tertentu. Untuk sumber daya tempat kebijakan dilampirkan, kebijakan menentukan tindakan apa yang dapat dilakukan oleh prinsipal tertentu pada sumber daya tersebut dan dalam kondisi apa. Anda harus [menentukan prinsipal](#) dalam kebijakan berbasis sumber daya. Prinsipal dapat mencakup akun, pengguna, peran, pengguna federasi, atau. Layanan AWS

Kebijakan berbasis sumber daya merupakan kebijakan inline yang terletak di layanan tersebut. Anda tidak dapat menggunakan kebijakan AWS terkelola dari IAM dalam kebijakan berbasis sumber daya.

## Daftar kontrol akses (ACL)

Daftar kontrol akses (ACL) mengendalikan prinsipal mana (anggota akun, pengguna, atau peran) yang memiliki izin untuk mengakses sumber daya. ACL serupa dengan kebijakan berbasis sumber daya, meskipun kebijakan tersebut tidak menggunakan format dokumen kebijakan JSON.

Amazon S3, AWS WAF, dan Amazon VPC adalah contoh layanan yang mendukung ACL. Untuk mempelajari ACL selengkapnya, lihat [Gambaran umum daftar kontrol akses \(ACL\)](#) dalam Panduan Developer Amazon Simple Storage Service.

## Jenis-jenis kebijakan lain

AWS mendukung jenis kebijakan tambahan yang kurang umum. Jenis-jenis kebijakan ini dapat mengatur izin maksimum yang diberikan kepada Anda oleh jenis kebijakan yang lebih umum.

- Batasan izin – Batasan izin adalah fitur lanjutan tempat Anda mengatur izin maksimum yang dapat diberikan oleh kebijakan berbasis identitas ke entitas IAM (pengguna IAM atau peran IAM). Anda dapat menetapkan batasan izin untuk suatu entitas. Izin yang dihasilkan adalah perpotongan antara kebijakan berbasis identitas milik entitas dan batasan izinnya. Kebijakan berbasis sumber daya yang menentukan pengguna atau peran dalam bidang `Principal` tidak dibatasi oleh batasan izin. Penolakan eksplisit dalam salah satu kebijakan ini akan menggantikan pemberian izin. Untuk informasi selengkapnya tentang batasan izin, lihat [Batasan izin untuk entitas IAM](#) dalam Panduan Pengguna IAM.
- Kebijakan kontrol layanan (SCP) — SCP adalah kebijakan JSON yang menentukan izin maksimum untuk organisasi atau unit organisasi (OU) di. AWS Organizations AWS Organizations adalah layanan untuk mengelompokkan dan mengelola secara terpusat beberapa Akun AWS yang dimiliki bisnis Anda. Jika Anda mengaktifkan semua fitur di organisasi, Anda dapat menerapkan kebijakan kontrol layanan (SCP) ke salah satu atau semua akun Anda. SCP membatasi izin untuk entitas di akun anggota, termasuk masing-masing. Pengguna root akun AWS Untuk informasi selengkapnya tentang Organisasi dan SCP, lihat [Cara kerja SCP](#) dalam Panduan Pengguna AWS Organizations .

- Kebijakan sesi – Kebijakan sesi adalah kebijakan lanjutan yang Anda berikan sebagai parameter ketika Anda membuat sesi sementara secara programatis untuk peran atau pengguna terfederasi. Izin sesi yang dihasilkan adalah perpotongan antara kebijakan berbasis identitas pengguna atau peran dan kebijakan sesi. Izin juga bisa datang dari kebijakan berbasis sumber daya. Penolakan eksplisit dalam salah satu kebijakan ini akan menggantikan pemberian izin. Untuk informasi selengkapnya, lihat [Kebijakan sesi](#) dalam Panduan Pengguna IAM.

## Berbagai jenis kebijakan

Ketika beberapa jenis kebijakan berlaku pada suatu permintaan, izin yang dihasilkan lebih rumit untuk dipahami. Untuk mempelajari cara AWS menentukan apakah akan mengizinkan permintaan saat beberapa jenis kebijakan terlibat, lihat [Logika evaluasi kebijakan](#) di Panduan Pengguna IAM.

## Bagaimana Layanan AWS bekerja dengan IAM

Untuk mendapatkan tampilan tingkat tinggi tentang cara Layanan AWS bekerja dengan sebagian besar fitur IAM, lihat [AWS layanan yang bekerja dengan IAM di Panduan Pengguna IAM](#).

Untuk mempelajari cara menggunakan yang spesifik Layanan AWS dengan IAM, lihat bagian keamanan dari Panduan Pengguna layanan yang relevan.

## Memecahkan masalah AWS identitas dan akses

Gunakan informasi berikut untuk membantu Anda mendiagnosis dan memperbaiki masalah umum yang mungkin Anda temui saat bekerja dengan AWS dan IAM.

### Topik

- [Saya tidak berwenang untuk melakukan tindakan di AWS](#)
- [Saya tidak berwenang untuk melakukan iam: PassRole](#)
- [Saya ingin mengizinkan orang di luar saya Akun AWS untuk mengakses AWS sumber daya saya](#)

## Saya tidak berwenang untuk melakukan tindakan di AWS

Jika Anda menerima pesan kesalahan bahwa Anda tidak memiliki otorisasi untuk melakukan tindakan, kebijakan Anda harus diperbarui agar Anda dapat melakukan tindakan tersebut.

Contoh kesalahan berikut terjadi ketika pengguna IAM `mateojackson` mencoba menggunakan konsol untuk melihat detail tentang suatu sumber daya `my-example-widget` rekaan, tetapi tidak memiliki izin `aws:GetWidget` rekaan.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:  
aws:GetWidget on resource: my-example-widget
```

Dalam hal ini, kebijakan untuk pengguna `mateojackson` harus diperbarui untuk mengizinkan akses ke sumber daya `my-example-widget` dengan menggunakan tindakan `aws:GetWidget`.

Jika Anda memerlukan bantuan, hubungi AWS administrator Anda. Administrator Anda adalah orang yang memberi Anda kredensial masuk.

## Saya tidak berwenang untuk melakukan `iam:PassRole`

Jika Anda menerima kesalahan yang tidak diizinkan untuk melakukan `iam:PassRole` tindakan, kebijakan Anda harus diperbarui agar Anda dapat meneruskan peran AWS.

Beberapa Layanan AWS memungkinkan Anda untuk meneruskan peran yang ada ke layanan tersebut alih-alih membuat peran layanan baru atau peran terkait layanan. Untuk melakukannya, Anda harus memiliki izin untuk meneruskan peran ke layanan.

Contoh kesalahan berikut terjadi ketika pengguna IAM bernama `marymajor` mencoba menggunakan konsol tersebut untuk melakukan tindakan di AWS. Namun, tindakan tersebut memerlukan layanan untuk mendapatkan izin yang diberikan oleh peran layanan. Mary tidak memiliki izin untuk meneruskan peran tersebut pada layanan.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:  
iam:PassRole
```

Dalam kasus ini, kebijakan Mary harus diperbarui agar dia mendapatkan izin untuk melakukan tindakan `iam:PassRole` tersebut.

Jika Anda memerlukan bantuan, hubungi AWS administrator Anda. Administrator Anda adalah orang yang memberi Anda kredensial masuk.

## Saya ingin mengizinkan orang di luar saya Akun AWS untuk mengakses AWS sumber daya saya

Anda dapat membuat peran yang dapat digunakan pengguna di akun lain atau orang-orang di luar organisasi Anda untuk mengakses sumber daya Anda. Anda dapat menentukan siapa saja yang

dipercaya untuk mengambil peran tersebut. Untuk layanan yang mendukung kebijakan berbasis sumber daya atau daftar kontrol akses (ACL), Anda dapat menggunakan kebijakan tersebut untuk memberi orang akses ke sumber daya Anda.

Untuk mempelajari selengkapnya, periksa referensi berikut:

- Untuk mempelajari apakah AWS mendukung fitur ini, lihat [Bagaimana Layanan AWS bekerja dengan IAM](#).
- Untuk mempelajari cara menyediakan akses ke sumber daya Anda di seluruh sumber daya Akun AWS yang Anda miliki, lihat [Menyediakan akses ke pengguna IAM di pengguna lain Akun AWS yang Anda miliki](#) di Panduan Pengguna IAM.
- Untuk mempelajari cara menyediakan akses ke sumber daya Anda kepada pihak ketiga Akun AWS, lihat [Menyediakan akses yang Akun AWS dimiliki oleh pihak ketiga](#) dalam Panduan Pengguna IAM.
- Untuk mempelajari cara memberikan akses melalui federasi identitas, lihat [Menyediakan akses ke pengguna terautentikasi eksternal \(federasi identitas\)](#) dalam Panduan Pengguna IAM.
- Untuk mempelajari perbedaan antara penggunaan kebijakan peran dan kebijakan berbasis sumber daya untuk akses lintas akun, lihat [Bagaimana peran IAM berbeda dari kebijakan berbasis sumber daya](#) dalam Panduan Pengguna IAM.

## Validasi Kepatuhan untuk AWS Produk atau Layanan ini


Untuk mempelajari apakah an Layanan AWS berada dalam lingkup program kepatuhan tertentu, lihat [Layanan AWS di Lingkup oleh Program Kepatuhan Layanan AWS](#) dan pilih program kepatuhan yang Anda minati. Untuk informasi umum, lihat [Program AWS Kepatuhan Program AWS](#) .

Anda dapat mengunduh laporan audit pihak ketiga menggunakan AWS Artifact. Untuk informasi selengkapnya, lihat [Mengunduh Laporan di AWS Artifact](#) .

Tanggung jawab kepatuhan Anda saat menggunakan Layanan AWS ditentukan oleh sensitivitas data Anda, tujuan kepatuhan perusahaan Anda, dan hukum dan peraturan yang berlaku. AWS menyediakan sumber daya berikut untuk membantu kepatuhan:

- [Panduan Memulai Cepat Keamanan dan Kepatuhan — Panduan](#) penerapan ini membahas pertimbangan arsitektur dan memberikan langkah-langkah untuk menerapkan lingkungan dasar AWS yang berfokus pada keamanan dan kepatuhan.

- [Arsitektur untuk Keamanan dan Kepatuhan HIPAA di Amazon Web Services](#) — Whitepaper ini menjelaskan bagaimana perusahaan dapat menggunakan AWS untuk membuat aplikasi yang memenuhi syarat HIPAA.

 Note

Tidak semua memenuhi Layanan AWS syarat HIPAA. Untuk informasi selengkapnya, lihat [Referensi Layanan yang Memenuhi Syarat HIPAA](#).

- [AWS Sumber Daya AWS](#) — Kumpulan buku kerja dan panduan ini mungkin berlaku untuk industri dan lokasi Anda.
- [AWS Panduan Kepatuhan Pelanggan](#) - Memahami model tanggung jawab bersama melalui lensa kepatuhan. Panduan ini merangkum praktik terbaik untuk mengamankan Layanan AWS dan memetakan panduan untuk kontrol keamanan di berbagai kerangka kerja (termasuk Institut Standar dan Teknologi Nasional (NIST), Dewan Standar Keamanan Industri Kartu Pembayaran (PCI), dan Organisasi Internasional untuk Standardisasi (ISO)).
- [Mengevaluasi Sumber Daya dengan Aturan](#) dalam Panduan AWS Config Pengembang — AWS Config Layanan menilai seberapa baik konfigurasi sumber daya Anda mematuhi praktik internal, pedoman industri, dan peraturan.
- [AWS Security Hub](#)— Ini Layanan AWS memberikan pandangan komprehensif tentang keadaan keamanan Anda di dalamnya AWS. Security Hub menggunakan kontrol keamanan untuk sumber daya AWS Anda serta untuk memeriksa kepatuhan Anda terhadap standar industri keamanan dan praktik terbaik. Untuk daftar layanan dan kontrol yang didukung, lihat [Referensi kontrol Security Hub](#).
- [Amazon GuardDuty](#) — Ini Layanan AWS mendeteksi potensi ancaman terhadap beban kerja Akun AWS, kontainer, dan data Anda dengan memantau lingkungan Anda untuk aktivitas yang mencurigakan dan berbahaya. GuardDuty dapat membantu Anda mengatasi berbagai persyaratan kepatuhan, seperti PCI DSS, dengan memenuhi persyaratan deteksi intrusi yang diamanatkan oleh kerangka kerja kepatuhan tertentu.
- [AWS Audit Manager](#)Ini Layanan AWS membantu Anda terus mengaudit AWS penggunaan Anda untuk menyederhanakan cara Anda mengelola risiko dan kepatuhan terhadap peraturan dan standar industri.

AWS Produk atau layanan ini mengikuti [model tanggung jawab bersama](#) melalui layanan Amazon Web Services (AWS) tertentu yang didukungnya. Untuk informasi keamanan AWS layanan, lihat

[halaman dokumentasi keamanan AWS layanan](#) dan [AWS layanan yang berada dalam lingkup upaya AWS kepatuhan oleh program kepatuhan](#).

## Ketahanan untuk AWS Produk atau Layanan ini

Infrastruktur AWS global dibangun di sekitar Wilayah AWS dan Availability Zones.

Wilayah AWS menyediakan beberapa Availability Zone yang terpisah secara fisik dan terisolasi, yang terhubung dengan latensi rendah, throughput tinggi, dan jaringan yang sangat redundan.

Dengan Zona Ketersediaan, Anda dapat merancang serta mengoperasikan aplikasi dan basis data yang secara otomatis melakukan fail over di antara zona tanpa gangguan. Zona Ketersediaan memiliki ketersediaan dan toleransi kesalahan yang lebih baik, dan dapat diskalakan dibandingkan infrastruktur pusat data tunggal atau multi tradisional.

Untuk informasi selengkapnya tentang AWS Wilayah dan Availability Zone, lihat [Infrastruktur AWS Global](#).

AWS Produk atau layanan ini mengikuti [model tanggung jawab bersama](#) melalui layanan Amazon Web Services (AWS) tertentu yang didukungnya. Untuk informasi keamanan AWS layanan, lihat [halaman dokumentasi keamanan AWS layanan](#) dan [AWS layanan yang berada dalam lingkup upaya AWS kepatuhan oleh program kepatuhan](#).

## Keamanan Infrastruktur untuk AWS Produk atau Layanan ini

AWS Produk atau layanan ini menggunakan layanan terkelola, dan karenanya dilindungi oleh keamanan jaringan AWS global. Untuk informasi tentang layanan AWS keamanan dan cara AWS melindungi infrastruktur, lihat [Keamanan AWS Cloud](#). Untuk mendesain AWS lingkungan Anda menggunakan praktik terbaik untuk keamanan infrastruktur, lihat [Perlindungan Infrastruktur dalam Kerangka Kerja](#) yang AWS Diarsiteksikan dengan Baik Pilar Keamanan.

Anda menggunakan panggilan API yang AWS dipublikasikan untuk mengakses AWS Produk atau Layanan ini melalui jaringan. Klien harus mendukung hal-hal berikut:

- Keamanan Lapisan Pengangkutan (TLS). Kami mensyaratkan TLS 1.2 dan menganjurkan TLS 1.3.
- Sandi cocok dengan sistem kerahasiaan maju sempurna (perfect forward secrecy, PFS) seperti DHE (Ephemeral Diffie-Hellman) atau ECDHE (Elliptic Curve Ephemeral Diffie-Hellman). Sebagian besar sistem modern seperti Java 7 dan versi lebih baru mendukung mode-mode ini.

Selain itu, permintaan harus ditandatangani menggunakan ID kunci akses dan kunci akses rahasia yang terkait dengan prinsipal IAM. Atau Anda dapat menggunakan [AWS Security Token Service](#) (AWS STS) untuk menghasilkan kredensial keamanan sementara untuk menandatangani permintaan.

AWS Produk atau layanan ini mengikuti [model tanggung jawab bersama](#) melalui layanan Amazon Web Services (AWS) tertentu yang didukungnya. Untuk informasi keamanan AWS layanan, lihat [halaman dokumentasi keamanan AWS layanan](#) dan [AWS layanan yang berada dalam lingkup upaya AWS kepatuhan oleh program kepatuhan](#).

## Konfigurasi dan Analisis Kerentanan di AWS Toolkit for Visual Studio

Toolkit for Visual Studio dirilis ke [Marketplace Visual Studio](#) saat fitur atau perbaikan baru dikembangkan. Pembaruan ini terkadang menyertakan pembaruan keamanan, jadi penting untuk selalu memperbarui Toolkit for Visual Studio.

Untuk memverifikasi bahwa pembaruan otomatis untuk ekstensi diaktifkan

1. Buka pengelola ekstensi dengan memilih Alat, Ekstensi, dan Pembaruan (Visual Studio 2017), atau Ekstensi, Kelola Ekstensi (Visual Studio 2019).
2. Pilih Ubah pengaturan Ekstensi dan Pembaruan Anda (Visual Studio 2017), atau Ubah pengaturan Anda untuk Ekstensi (Visual Studio 2019).
3. Sesuaikan pengaturan untuk lingkungan Anda.

Jika Anda memilih untuk menonaktifkan pembaruan otomatis untuk ekstensi, pastikan untuk memeriksa pembaruan Toolkit for Visual Studio pada interval yang sesuai untuk lingkungan Anda.



# Riwayat dokumen Panduan AWS Toolkit for Visual Studio Pengguna

Pembaruan dokumentasi terakhir: April 21, 2021

## Riwayat dokumen

Tabel berikut menjelaskan perubahan penting terbaru dari Panduan AWS Toolkit for Visual Studio Pengguna. Untuk notifikasi tentang pembaruan dokumentasi ini, Anda dapat berlangganan ke [umpan RSS](#).

Perubahan	Deskripsi	Tanggal
<a href="#">Pembaruan dan pemeliharaan konten</a>	Memperbarui konten untuk perubahan pada UI dan pedoman AWS gaya.	Maret 6, 2024
<a href="#">Pembaruan dan pemeliharaan konten</a>	Memperbarui konten untuk perubahan pada UI dan pedoman AWS gaya.	Maret 6, 2024
<a href="#">Pembaruan dan pemeliharaan konten</a>	Memperbarui konten untuk perubahan pada UI dan pedoman AWS gaya.	Maret 6, 2024
<a href="#">Pembaruan dan pemeliharaan konten</a>	Memperbarui konten untuk perubahan pada UI dan pedoman AWS gaya.	Maret 6, 2024
<a href="#">Pembaruan dan pemeliharaan konten</a>	Memperbarui konten untuk perubahan pada UI dan pedoman AWS gaya.	Maret 6, 2024
<a href="#">Pembaruan untuk mengatur dan otentikasi</a>	Topik penyiapan dan otentikasi telah diperbarui untuk meningkatkan keamanan dan pengalaman orientasi toolkit.	22 Juni 2023

	Lihat ToC topik <a href="#">Memulai</a> dan <a href="#">Otentikasi dan akses</a> untuk melihat perubahan.	
<a href="#">Otentikasi dan akses</a>	Memberikan AWS kredensi sekarang Otentikasi dan akses. Refactoring TOC dan subtopik untuk memenuhi persyaratan AWS gaya dan keamanan.	4 Mei 2023
<a href="#">Topik pemecahan masalah umum baru</a>	Topik <a href="#">Pemecahan Masalah</a> berisi informasi pemecahan masalah umum untuk dan layanan terkait. AWS Toolkit for Visual Studio	April 30, 2023
<a href="#">Pembaruan pada bagian dan topik Menyiapkan</a>	<a href="#">Menyiapkan AWS Toolkit for Visual Studio bagian</a> dan topik dalam Panduan Pengguna ini telah diperbarui untuk meningkatkan pengalaman naik pesawat untuk AWS Toolkit for Visual Studio.	30 Januari 2023
<a href="#">Pembaruan pada bagian dan topik Menyiapkan</a>	<a href="#">Menyiapkan AWS Toolkit for Visual Studio bagian</a> dan topik dalam Panduan Pengguna ini telah diperbarui untuk meningkatkan pengalaman naik pesawat untuk AWS Toolkit for Visual Studio.	30 Januari 2023
<a href="#">Menambahkan AWS Toolkit for Visual Studio informasi 2022</a>	Support untuk Visual Studio 2022 telah ditambahkan ke AWS Toolkit for Visual Studio.	Desember 20, 2022

<a href="#">Pembaruan untuk Publikasi ke AWS panduan</a>	Pembaruan dokumentasi untuk mencerminkan perubahan yang dibuat pada layanan untuk peluncuran GA.	6 Juli 2022
<a href="#">Pembaruan judul dan relokasi</a>	Perubahan judul kecil dilakukan untuk mencerminkan konten dengan lebih baik. Panduan sekarang terletak di Publishing to AWS guide.	6 Juli 2022
<a href="#">Menyebarkan ke AWS: pembaruan judul dan konten</a>	Bagian panduan secara resmi berjudul: Deployment Using the AWS Toolkit, memiliki daftar isi (TOC) yang diperbarui dan sekarang berjudul: Deploying to AWS Panduan berikut telah menyelesaikan penghentian dan tidak lagi dapat diakses: Deploying to Elastic Beanstalk (Legacy) dan Deploying to (Legacy). AWS CloudFormation Konten yang diperbarui mengenai penyebaran ke Elastic Beanstalk dan Cloudformation dapat ditemukan dari TOC yang diperbarui dalam panduan ini.	6 Juli 2022
<a href="#">Menerapkan Aplikasi ASP.NET Core 2.0 (Fargate) sekarang menjadi panduan lama</a>	Dokumentasi ini mengacu pada layanan dan fitur lama. Untuk panduan dan konten yang diperbarui, lihat panduan <a href="#">AWS alat.NET Deployment</a> dan AWS daftar isi <a href="#">Deploying to</a> yang diperbarui.	6 Juli 2022

[Menerapkan Aplikasi ASP.NET sekarang menjadi panduan lama](#)

Dokumentasi ini mengacu pada layanan dan fitur lama. Untuk panduan dan konten yang diperbarui, lihat panduan [alat penyebaran AWS .NET](#) dan AWS daftar isi [Deploying to](#) yang diperbarui.

6 Juli 2022

[Menerapkan Aplikasi ASP.NET sekarang menjadi panduan lama](#)

Dokumentasi ini mengacu pada layanan dan fitur lama. Untuk panduan dan konten yang diperbarui, lihat panduan [alat penyebaran AWS .NET](#) dan AWS daftar isi [Deploying to](#) yang diperbarui.

6 Juli 2022

[Topik panduan baru: Bekerja dengan CloudWatch Log di Visual Studio](#)

Membuat topik ikhtisar baru untuk [integrasi Amazon CloudWatch Logs dalam panduan Visual Studio](#).

Juni 29, 2022

[Topik panduan baru: Menyiapkan integrasi CloudWatch Log untuk Visual Studio](#)

Membuat bagian pengaturan baru untuk [integrasi Amazon CloudWatch Logs dalam panduan Visual Studio](#).

Juni 29, 2022

[CloudWatch Integrasi log untuk Visual Studio](#)

Membuat panduan baru untuk integrasi Amazon CloudWatch Log di Visual Studio, termasuk topik panduan: [Menyiapkan CloudWatch Log untuk Visual Studio](#) dan [Bekerja dengan CloudWatch Log di Visual Studio](#).

Juni 29, 2022

<a href="#">Publikasikan ke AWS</a>	Publikasikan ke AWS tidak lagi dalam pratinjau. Pembaruan untuk mencerminkan perubahan pada UI dan peningkatan saran penerbitan.	1 Juni 2022
<a href="#">Publikasikan baru untuk AWS tersedia untuk pratinjau</a>	Pengalaman penerapan yang disempurnakan yang memberikan panduan tentang AWS layanan mana yang tepat untuk aplikasi Anda.	21 Oktober 2021
<a href="#">Dukungan SSO dan MFA untuk kredensi AWS</a>	Diperbarui untuk mendokumentasikan dukungan baru untuk AWS Single Sign-On (IAM Identity Center) dan otentikasi multi-faktor dalam kredensi AWS.	21 April 2021
<a href="#">AWS Lambda Proyek Dasar Membuat Gambar Docker</a>	Menambahkan dukungan untuk citra kontainer Lambda.	1 Desember 2020
<a href="#">Konten Keamanan</a>	Menambahkan konten keamanan.	6 Februari 2020
<a href="#">Memberikan AWS kredensi</a>	Diperbarui dengan informasi tentang membuat profil kredensi di file AWS kredensial bersama.	20 Juni 2019
<a href="#">Menggunakan Proyek AWS Lambda di AWS Toolkit for Visual Studio</a>	Support untuk Visual Studio 2019 telah ditambahkan ke AWS Toolkit for Visual Studio.	28 Maret 2019
<a href="#">Tutorial: Membuat Aplikasi Amazon Rekognition Lambda</a>	Support untuk Visual Studio 2019 telah ditambahkan ke AWS Toolkit for Visual Studio.	28 Maret 2019

<a href="#">Tutorial: Membangun dan Menguji Aplikasi Tanpa Server dengan Lambda AWS</a>	Support untuk Visual Studio 2019 telah ditambahkan ke AWS Toolkit for Visual Studio.	28 Maret 2019
<a href="#">Menyiapkan AWS Toolkit for Visual Studio</a>	Support untuk Visual Studio 2019 telah ditambahkan ke AWS Toolkit for Visual Studio.	28 Maret 2019
<a href="#">Menerapkan Aplikasi ASP.NET Core 2.0 (Fargate)</a>	Support untuk Visual Studio 2019 telah ditambahkan ke AWS Toolkit for Visual Studio.	28 Maret 2019
<a href="#">Menerapkan Aplikasi ASP.NET Core 2.0 (EC2)</a>	Support untuk Visual Studio 2019 telah ditambahkan ke AWS Toolkit for Visual Studio.	28 Maret 2019
<a href="#">Membuat Proyek AWS CloudFormation Template di Visual Studio</a>	Support untuk Visual Studio 2019 telah ditambahkan ke AWS Toolkit for Visual Studio.	28 Maret 2019
<a href="#">Tampilan Detil dari Layanan Kontainer</a>	Menambahkan informasi tentang tampilan mendetail klaster Amazon Elastic Container Service dan repositori kontainer yang disediakan oleh Explorer. AWS	16 Februari 2018
<a href="#">Menyebarkan ke Amazon EC2 Container Service</a>	Menambahkan informasi tentang penerapan ke layanan kontainer Amazon EC2.	16 Februari 2018

[Menyebarkan Layanan Kontainer menggunakan Fargate](#)

Menambahkan informasi tentang cara menerapkan aplikasi ASP.NET Core 2.0 kontainer yang menargetkan Linux melalui Amazon ECS menggunakan tipe peluncuran Fargate.

16 Februari 2018

[Menyebarkan Layanan Kontainer menggunakan EC2](#)

Menambahkan informasi tentang cara menerapkan aplikasi ASP.NET Core 2.0 kontainer yang menargetkan Linux melalui Amazon ECS menggunakan tipe peluncuran EC2.

16 Februari 2018

[Kredensi untuk Deploying ke Amazon EC2 Container Service](#)

Menambahkan informasi tentang cara menentukan kredensi saat menerapkan ke layanan penampung Amazon EC2.

16 Februari 2018

Terjemahan disediakan oleh mesin penerjemah. Jika konten terjemahan yang diberikan bertentangan dengan versi bahasa Inggris aslinya, utamakan versi bahasa Inggris.