



Panduan Developerr

Amazon Elastic Compute Cloud



Amazon Elastic Compute Cloud: Panduan Developerr

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 antara para 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

Akses terprogram ke Amazon EC2	1
Titik akhir layanan	1
Titik akhir IPv4	7
Titik akhir tumpukan ganda (IPv4 dan IPv6)	8
Menentukan titik akhir	8
Konsistensi akhirnya	10
Idempotensi	12
Idempotensi di Amazon EC2	13
RunInstances idempotensi	16
Contoh	17
Coba lagi rekomendasi untuk permintaan idempoten	19
Throttling permintaan API	20
Bagaimana throttling diterapkan	21
Batas pelambatan	22
Pantau pelambatan API	30
Mencoba lagi dan backoff eksponensial	30
Minta kenaikan batas	31
Menggunakan AWS CLI	33
Pelajari lebih lanjut tentang AWS CLI	33
Menggunakan AWS CloudFormation	34
Amazon EC2 dan template AWS CloudFormation	34
Sumber daya untuk Amazon EC2	34
Pelajari lebih lanjut tentang AWS CloudFormation	38
Menggunakan AWS SDK	39
Contoh kode untuk Amazon EC2 API	39
Pelajari lebih lanjut tentang AWS SDK	39
API tingkat rendah untuk Amazon EC2	40
Konsol-ke-Kode	41
Cara kerjanya	41
Batasan	42
Wilayah yang Didukung	42
Format yang didukung	42
Tindakan yang dipertahankan	42
Tabel tindakan yang dicatat	42

Menggunakan Konsol-ke-Kode	43
Contoh kode	46
Tindakan	62
AcceptVpcPeeringConnection	68
AllocateAddress	70
AllocateHosts	82
AssignPrivateIpAddresses	85
AssociateAddress	86
AssociateDhcpOptions	99
AssociateRouteTable	100
AttachInternetGateway	101
AttachNetworkInterface	102
AttachVolume	104
AttachVpnGateway	105
AuthorizeSecurityGroupEgress	106
AuthorizeSecurityGroupIngress	108
CancelCapacityReservation	129
CancelImportTask	130
CancelSpotFleetRequests	131
CancelSpotInstanceRequests	133
ConfirmProductInstance	134
CopyImage	136
CopySnapshot	138
CreateCapacityReservation	140
CreateCustomerGateway	143
CreateDhcpOptions	145
CreateFlowLogs	147
CreateImage	149
CreateInstanceExportTask	152
CreateInternetGateway	154
CreateKeyPair	155
CreateLaunchTemplate	170
CreateNetworkAcl	179
CreateNetworkAclEntry	180
CreateNetworkInterface	182
CreatePlacementGroup	187

CreateRoute	188
CreateRouteTable	190
CreateSecurityGroup	195
CreateSnapshot	215
CreateSpotDatafeedSubscription	217
CreateSubnet	219
CreateTags	226
CreateVolume	229
CreateVpc	234
CreateVpcEndpoint	241
CreateVpnConnection	245
CreateVpnConnectionRoute	251
CreateVpnGateway	252
DeleteCustomerGateway	253
DeleteDhcpOptions	254
DeleteFlowLogs	256
DeleteInternetGateway	257
DeleteKeyPair	258
DeleteLaunchTemplate	268
DeleteNetworkAcl	272
DeleteNetworkAclEntry	273
DeleteNetworkInterface	274
DeletePlacementGroup	275
DeleteRoute	277
DeleteRouteTable	278
DeleteSecurityGroup	279
DeleteSnapshot	289
DeleteSpotDatafeedSubscription	290
DeleteSubnet	291
DeleteTags	292
DeleteVolume	295
DeleteVpc	296
DeleteVpnConnection	297
DeleteVpnConnectionRoute	298
DeleteVpnGateway	299
DeregisterImage	300

DescribeAccountAttributes	301
DescribeAddresses	304
DescribeAvailabilityZones	313
DescribeBundleTasks	320
DescribeCapacityReservations	321
DescribeCustomerGateways	324
DescribeDhcpOptions	326
DescribeFlowLogs	330
DescribeHostReservationOfferings	332
DescribeHosts	335
DescribeIamInstanceProfileAssociations	337
DescribeIdFormat	342
DescribeIdentityIdFormat	343
DescribeImageAttribute	345
DescribeImages	348
DescribeImportImageTasks	358
DescribeImportSnapshotTasks	361
DescribeInstanceAttribute	364
DescribeInstanceState	367
DescribeInstanceTypes	371
DescribeInstances	384
DescribeInternetGateways	413
DescribeKeyPairs	414
DescribeNetworkAcls	425
DescribeNetworkInterfaceAttribute	429
DescribeNetworkInterfaces	432
DescribePlacementGroups	437
DescribePrefixLists	439
DescribeRegions	440
DescribeRouteTables	454
DescribeScheduledInstanceAvailability	458
DescribeScheduledInstances	460
DescribeSecurityGroups	462
DescribeSnapshotAttribute	477
DescribeSnapshots	479
DescribeSpotDatafeedSubscription	485

DescribeSpotFleetInstances	486
DescribeSpotFleetRequestHistory	488
DescribeSpotFleetRequests	490
DescribeSpotInstanceRequests	494
DescribeSpotPriceHistory	498
DescribeSubnets	501
DescribeTags	509
DescribeVolumeAttribute	515
DescribeVolumeStatus	516
DescribeVolumes	518
DescribeVpcAttribute	523
DescribeVpcClassicLink	525
DescribeVpcClassicLinkDnsSupport	526
DescribeVpcEndpointServices	528
DescribeVpcEndpoints	532
DescribeVpcs	536
DescribeVpnConnections	543
DescribeVpnGateways	546
DetachInternetGateway	548
DetachNetworkInterface	549
DetachVolume	550
DetachVpnGateway	551
DisableVgwRoutePropagation	552
DisableVpcClassicLink	553
DisableVpcClassicLinkDnsSupport	554
DisassociateAddress	555
DisassociateRouteTable	563
EnableVgwRoutePropagation	564
EnableVolumeIo	565
EnableVpcClassicLink	566
EnableVpcClassicLinkDnsSupport	567
GetConsoleOutput	568
GetHostReservationPurchasePreview	570
GetPasswordData	572
ImportImage	574
ImportKeyPair	576

ImportSnapshot	577
ModifyCapacityReservation	579
ModifyHosts	581
ModifyIdFormat	583
ModifyImageAttribute	584
ModifyInstanceAttribute	586
ModifyInstanceCreditSpecification	590
ModifyNetworkInterfaceAttribute	591
ModifyReservedInstances	593
ModifySnapshotAttribute	595
ModifySpotFleetRequest	597
ModifySubnetAttribute	598
ModifyVolumeAttribute	599
ModifyVpcAttribute	600
MonitorInstances	602
MoveAddressToVpc	606
PurchaseHostReservation	608
PurchaseScheduledInstances	609
RebootInstances	611
RegisterImage	622
RejectVpcPeeringConnection	624
ReleaseAddress	625
ReleaseHosts	635
ReplaceIamInstanceProfileAssociation	637
ReplaceNetworkAclAssociation	643
ReplaceNetworkAclEntry	644
ReplaceRoute	645
ReplaceRouteTableAssociation	646
ReportInstanceStatus	647
RequestSpotFleet	648
RequestSpotInstances	653
ResetImageAttribute	658
ResetInstanceAttribute	659
ResetNetworkInterfaceAttribute	661
ResetSnapshotAttribute	662
RevokeSecurityGroupEgress	663

RevokeSecurityGroupIngress	665
RunInstances	667
RunScheduledInstances	688
StartInstances	690
StopInstances	706
TerminateInstances	722
UnassignPrivateIpAddresses	734
UnmonitorInstances	735
Skenario	738
Membangun dan mengelola layanan yang tangguh	739
Memulai instans	899
Pantau permintaan API menggunakan CloudWatch	1039
Aktifkan metrik API Amazon EC2	1039
Metrik dan dimensi API Amazon EC2	1040
Metrik	1040
Dimensi	1041
Retensi data metrik	1041
Permintaan pemantauan yang dibuat atas nama Anda	1042
Penagihan	1042
Bekerja dengan Amazon CloudWatch	1042
Melihat CloudWatch metrik	1042
Membuat CloudWatch alarm	1043
.....	mxlv

Akses terprogram ke Amazon EC2

Anda dapat membuat dan mengelola sumber daya Amazon EC2 menggunakan AWS Management Console atau antarmuka terprogram. Untuk informasi tentang menggunakan konsol Amazon EC2, lihat Panduan Pengguna [Amazon EC2](#).

Cara kerjanya

- [Titik akhir Amazon EC2](#)
- [Konsistensi akhirnya](#)
- [Idempotensi](#)
- [Minta pelambatan](#)

Antarmuka terprogram

- [AWS Command Line Interface \(AWS CLI\)](#)
- [AWS CloudFormation](#)
- [AWS SDK](#)
- [API tingkat rendah](#)

Memulai

- [Contoh kode](#)
- [Konsol-ke-Kode](#)

Pemantauan

- [AWS CloudTrail](#)
- [Memantau permintaan](#)

Titik akhir layanan Amazon EC2

Endpoint adalah URL yang berfungsi sebagai titik masuk untuk layanan AWS web. Amazon EC2 mendukung jenis titik akhir berikut:

- Titik akhir IPv4
- Titik akhir tumpukan ganda yang mendukung IPv4 dan IPv6
- Titik akhir FIPS

Saat Anda membuat permintaan, Anda dapat menentukan titik akhir dan Wilayah yang akan digunakan. Jika Anda tidak menentukan titik akhir, titik akhir IPv4 digunakan secara default. Untuk menggunakan tipe titik akhir yang berbeda, Anda harus menentukannya dalam permintaan Anda. Untuk contoh cara melakukannya, lihat [Menentukan titik akhir](#).

Nama Wilayah	Wilayah	Titik Akhir	Protokol
AS Timur (Ohio)	us-east-2	ec2.us-east-2.amazonaws.com	HTTP dan HTTPS
		ec2-fips.us-east-2.amazonaws.com	HTTPS
		ec2.us-east-2.api.aws	HTTPS
AS Timur (Virginia Utara)	us-east-1	ec2.us-east-1.amazonaws.com	HTTP dan HTTPS
		ec2-fips.us-east-1.amazonaws.com	HTTPS
		ec2.us-east-1.api.aws	HTTPS
AS Barat (California Utara)	us-west-1	ec2.us-west-1.amazonaws.com	HTTP dan HTTPS
		ec2-fips.us-west-1.amazonaws.com	HTTPS
		ec2.us-west-1.api.aws	HTTPS
AS Barat (Oregon)	us-west-2	ec2.us-west-2.amazonaws.com	HTTP dan HTTPS
		ec2-fips.us-west-2.amazonaws.com	HTTPS

Nama Wilayah	Wilayah	Titik Akhir	Protokol
		ec2.us-west-2.api.aws	HTTPS HTTPS
Afrika (Cape Town)	af-south-1	ec2.af-south-1.amazonaws.com ec2.af-south-1.api.aws	HTTP dan HTTPS HTTPS
Asia Pasifik (Hong Kong)	ap-east-1	ec2.ap-east-1.amazonaws.com ec2.ap-east-1.api.aws	HTTP dan HTTPS HTTPS
Asia Pasifik (Hyderabad)	ap-south-2	ec2.ap-south-2.amazonaws.com	HTTPS
Asia Pasifik (Jakarta)	ap-southeast-3	ec2.ap-southeast-3.amazonaws.com	HTTPS
Asia Pasifik (Melbourne)	ap-southeast-4	ec2.ap-southeast-4.amazonaws.com	HTTPS
Asia Pasifik (Mumbai)	ap-south-1	ec2.ap-south-1.amazonaws.com ec2.ap-south-1.api.aws	HTTP dan HTTPS HTTPS

Nama Wilayah	Wilayah	Titik Akhir	Protokol
Asia Pasifik (Osaka)	ap-northeast-3	ec2.ap-northeast-3.amazonaws.com	HTTP dan HTTPS
Asia Pasifik (Seoul)	ap-northeast-2	ec2.ap-northeast-2.amazonaws.com ec2.ap-northeast-2.api.aws	HTTP dan HTTPS HTTPS
Asia Pasifik (Singapura)	ap-southeast-1	ec2.ap-southeast-1.amazonaws.com ec2.ap-southeast-1.api.aws	HTTP dan HTTPS HTTPS
Asia Pasifik (Sydney)	ap-southeast-2	ec2.ap-southeast-2.amazonaws.com ec2.ap-southeast-2.api.aws	HTTP dan HTTPS HTTPS
Asia Pacific (Tokyo)	ap-northeast-1	ec2.ap-northeast-1.amazonaws.com ec2.ap-northeast-1.api.aws	HTTP dan HTTPS HTTPS
Kanada (Pusat)	ca-central-1	ec2.ca-central-1.amazonaws.com ec2-fips.ca-central-1.amazonaws.com ec2.ca-central-1.api.aws	HTTP dan HTTPS HTTPS HTTPS

Nama Wilayah	Wilayah	Titik Akhir	Protokol
Kanada Barat (Calgary)	ca-west-1	ec2.ca-west-1.amazonaws.com	HTTPS
		ec2-fips.ca-west-1.amazonaws.com	HTTPS
Eropa (Frankfurt)	eu-central-1	ec2.eu-central-1.amazonaws.com	HTTP dan HTTPS
		ec2.eu-central-1.api.aws	HTTPS
			HTTPS
Eropa (Irlandia)	eu-west-1	ec2.eu-west-1.amazonaws.com	HTTP dan HTTPS
		ec2.eu-west-1.api.aws	HTTPS
			HTTPS
Eropa (London)	eu-west-2	ec2.eu-west-2.amazonaws.com	HTTP dan HTTPS
		ec2.eu-west-2.api.aws	HTTPS
			HTTPS
Eropa (Milan)	eu-south-1	ec2.eu-south-1.amazonaws.com	HTTP dan HTTPS
		ec2.eu-south-1.api.aws	HTTPS
			HTTPS
Eropa (Paris)	eu-west-3	ec2.eu-west-3.amazonaws.com	HTTP dan HTTPS
		ec2.eu-west-3.api.aws	HTTPS
			HTTPS
Eropa (Spanyol)	eu-south-2	ec2.eu-south-2.amazonaws.com	HTTPS

Nama Wilayah	Wilayah	Titik Akhir	Protokol
Eropa (Stockholm)	eu-north-1	ec2.eu-north-1.amazonaws.com ec2.eu-north-1.api.aws	HTTP dan HTTPS HTTPS
Eropa (Zürich)	eu-central-2	ec2.eu-central-2.amazonaws.com	HTTPS
Israel (Tel Aviv)	il-central-1	ec2.il-central-1.amazonaws.com	HTTPS
Timur Tengah (Bahrain)	me-south-1	ec2.me-south-1.amazonaws.com ec2.me-south-1.api.aws	HTTP dan HTTPS HTTPS
Timur Tengah (UAE)	me-central-1	ec2.me-central-1.amazonaws.com	HTTPS
Amerika Selatan (Sao Paulo)	sa-east-1	ec2.sa-east-1.amazonaws.com ec2.sa-east-1.api.aws	HTTP dan HTTPS HTTPS
AWS GovCloud (AS-Timur)	us-gov-east-1	ec2.us-gov-east-1.amazonaws.com ec2.us-gov-east-1.api.aws	HTTPS HTTPS

Nama Wilayah	Wilayah	Titik Akhir	Protokol
AWS GovCloud (AS-Barat)	us-gov-west-1	ec2.us-gov-west-1.amazonaws.com ec2.us-gov-west-1.api.aws	HTTPS HTTPS

Untuk informasi selengkapnya tentang Wilayah, lihat [Wilayah dan Zona Ketersediaan](#) di Panduan Pengguna Amazon EC2. Untuk daftar titik akhir Amazon EC2, [lihat Wilayah dan Titik Akhir](#) di Referensi Umum Amazon Web Services.

Topik

- [Titik akhir IPv4](#)
- [Titik akhir tumpukan ganda \(IPv4 dan IPv6\)](#)
- [Menentukan titik akhir](#)

Untuk informasi selengkapnya tentang titik akhir FIPS, lihat, [titik akhir FIPS](#) di Referensi Umum Amazon Web Services.

Titik akhir IPv4

Titik akhir IPv4 hanya mendukung lalu lintas IPv4. Titik akhir IPv4 tersedia untuk semua Wilayah.

Jika Anda menentukan titik akhir umum, `ec2.amazonaws.com`, kami menggunakan titik akhir untuk `us-east-1`. Untuk menggunakan Wilayah yang berbeda, tentukan titik akhir yang terkait. Misalnya, jika Anda menentukan `ec2.us-east-2.amazonaws.com` sebagai titik akhir, kami mengarahkan permintaan Anda ke titik `us-east-2` akhir.

Nama titik akhir IPv4 menggunakan konvensi penamaan berikut:

- `service.region.amazonaws.com`

Misalnya, nama titik akhir IPv4 untuk Wilayah `eu-west-1` adalah `ec2.eu-west-1.amazonaws.com`. Untuk daftar titik akhir Amazon EC2, [lihat Wilayah dan Titik Akhir](#) di Referensi Umum Amazon Web Services.

Titik akhir tumpukan ganda (IPv4 dan IPv6)

Titik akhir tumpukan ganda mendukung lalu lintas IPv4 dan IPv6. Titik akhir tumpukan ganda hanya tersedia di Wilayah berikut:

- `us-east-1`—AS Timur (Virginia Utara)
- `us-east-2`—AS Timur (Ohio)
- `us-west-2`—AS Barat (Oregon)
- `eu-west-1`—Eropa (Irlandia)
- `ap-south-1`—Asia Pasifik (Mumbai)
- `sa-east-1`—Amerika Selatan (São Paulo)
- `us-gov-east-1`—AWS GovCloud (AS-Timur)
- `us-gov-west-1`—AWS GovCloud (AS-Barat)

Saat Anda membuat permintaan ke titik akhir tumpukan ganda, URL titik akhir memutuskan ke alamat IPv6 atau IPv4, tergantung pada protokol yang digunakan oleh jaringan dan klien Anda.

Amazon EC2 hanya mendukung titik akhir dual-stack regional, yang berarti Anda harus menentukan Wilayah sebagai bagian dari nama titik akhir. Nama titik akhir tumpukan ganda menggunakan konvensi penamaan berikut:

- `ec2.region.api.aws`

Misalnya, nama titik akhir tumpukan ganda untuk Wilayah `eu-west-1` adalah `ec2.eu-west-1.api.aws`. Untuk daftar titik akhir Amazon EC2, [lihat Wilayah dan Titik Akhir](#) di Referensi Umum Amazon Web Services.

Menentukan titik akhir

Bagian ini memberikan beberapa contoh cara menentukan titik akhir saat membuat permintaan.

AWS CLI

Contoh berikut menunjukkan cara menentukan titik akhir untuk `us-east-2` Wilayah menggunakan AWS CLI

- Tumpukan ganda

```
aws ec2 describe-regions --region us-east-2 --endpoint-url https://ec2.us-east-2.api.aws
```

- IPv4

```
aws ec2 describe-regions --region us-east-2 --endpoint-url https://ec2.us-east-2.amazonaws.com
```

AWS SDK for Java 2.x

Contoh berikut menunjukkan cara menentukan titik akhir untuk us-east-2 Wilayah menggunakan AWS SDK for Java 2.x

- Tumpukan ganda

```
Ec2Client client = Ec2Client.builder()  
    .region(Region.US_EAST_2)  
    .endpointOverride(URI.create("https://ec2.us-east-2.api.aws"))  
    .build();
```

- IPv4

```
Ec2Client client = Ec2Client.builder()  
    .region(Region.US_EAST_2)  
    .endpointOverride(URI.create("https://ec2.us-east-2.amazonaws.com"))  
    .build();
```

AWS SDK for Java 1.x

Contoh berikut menunjukkan cara menentukan titik akhir untuk eu-west-1 Wilayah menggunakan AWS SDK for Java 1.x.

- Tumpukan ganda

```
AmazonEC2 s3 = AmazonEC2ClientBuilder.standard()  
    .withEndpointConfiguration(new EndpointConfiguration(  
        "https://ec2.eu-west-1.api.aws",  
        "eu-west-1"))
```

```
.build();
```

- IPv4

```
AmazonEC2 s3 = AmazonEC2ClientBuilder.standard()  
    .withEndpointConfiguration(new EndpointConfiguration(  
        "https://ec2.eu-west-1.amazonaws.com",  
        "eu-west-1"))  
    .build();
```

AWS SDK for Go

Contoh berikut menunjukkan cara menentukan titik akhir untuk us-east-1 Wilayah menggunakan AWS SDK for Go

- Tumpukan ganda

```
sess := session.Must(session.NewSession())  
svc := ec2.New(sess, &aws.Config{  
    Region: aws.String(endpoints.UsEast1RegionID),  
    Endpoint: aws.String("https://ec2.us-east-1.api.aws")  
})
```

- IPv4

```
sess := session.Must(session.NewSession())  
svc := ec2.New(sess, &aws.Config{  
    Region: aws.String(endpoints.UsEast1RegionID),  
    Endpoint: aws.String("https://ec2.us-east-1.amazonaws.com")  
})
```

Konsistensi akhirnya di Amazon EC2 API

Amazon EC2 API mengikuti model konsistensi akhirnya, karena sifat terdistribusi dari sistem yang mendukung API. Ini berarti bahwa hasil dari perintah API yang Anda jalankan yang memengaruhi sumber daya Amazon EC2 Anda mungkin tidak langsung terlihat oleh semua perintah berikutnya yang Anda jalankan. Anda harus mengingat hal ini saat menjalankan perintah API yang segera mengikuti perintah API sebelumnya.

Konsistensi akhirnya dapat memengaruhi cara Anda mengelola sumber daya Anda. Misalnya, jika Anda menjalankan perintah untuk membuat sumber daya, pada akhirnya akan terlihat oleh perintah lain. Ini berarti bahwa jika Anda menjalankan perintah untuk memodifikasi atau mendeskripsikan sumber daya yang baru saja Anda buat, ID-nya mungkin tidak disebarkan ke seluruh sistem, dan Anda akan mendapatkan kesalahan yang merespons bahwa sumber daya tidak ada.

Untuk mengelola konsistensi akhirnya, Anda dapat melakukan hal berikut:

- Konfirmasikan status sumber daya sebelum Anda menjalankan perintah untuk memodifikasinya. Jalankan `Describe` perintah yang sesuai menggunakan algoritma backoff eksponensial untuk memastikan bahwa Anda memberikan cukup waktu untuk perintah sebelumnya untuk menyebar melalui sistem. Untuk melakukan ini, jalankan `Describe` perintah berulang kali, dimulai dengan beberapa detik waktu tunggu, dan meningkat secara bertahap hingga lima menit waktu tunggu.
- Tambahkan waktu tunggu antara perintah berikutnya, bahkan jika `Describe` perintah mengembalikan respons yang akurat. Terapkan algoritma backoff eksponensial dimulai dengan beberapa detik waktu tunggu, dan tingkatkan secara bertahap hingga sekitar lima menit waktu tunggu.

Contoh kesalahan konsistensi akhirnya

Berikut ini adalah contoh kode kesalahan yang mungkin Anda temui sebagai akibat dari konsistensi akhirnya.

- `InvalidInstanceID.NotFound`

Jika Anda berhasil menjalankan `RunInstances` perintah, dan kemudian segera menjalankan perintah lain menggunakan ID instance yang disediakan dalam `responseRunInstances`, itu mungkin mengembalikan `InvalidInstanceID.NotFound` kesalahan. Ini tidak berarti instance tidak ada.

Beberapa perintah spesifik yang mungkin terpengaruh adalah:

- `DescribeInstances`: Untuk mengonfirmasi keadaan sebenarnya dari instance, jalankan perintah ini menggunakan algoritma backoff eksponensial.
- `TerminateInstances`: Untuk mengonfirmasi status instance, pertama jalankan `DescribeInstances` perintah menggunakan algoritma backoff eksponensial.

⚠ Important

Jika Anda mendapatkan `InvalidInstanceID.NotFound` kesalahan setelah menjalankan `TerminateInstances`, ini tidak berarti bahwa instance tersebut akan atau akan dihentikan. Instance Anda masih bisa berjalan. Inilah sebabnya mengapa penting untuk terlebih dahulu mengkonfirmasi status instance menggunakan `DescribeInstances`.

- `InvalidGroup.NotFound`

Jika Anda berhasil menjalankan `CreateSecurityGroup` perintah, dan kemudian segera menjalankan perintah lain menggunakan ID grup keamanan yang disediakan dalam `responsCreateSecurityGroup`, itu mungkin mengembalikan `InvalidGroup.NotFound` kesalahan. Untuk mengonfirmasi status grup keamanan, jalankan `DescribeSecurityGroups` perintah menggunakan algoritma backoff eksponensial.

- `InstanceLimitExceeded`

Anda telah meminta lebih banyak instance daripada yang diizinkan batas instans saat ini untuk jenis instans yang ditentukan. Anda dapat mencapai batas ini secara tidak terduga jika Anda meluncurkan dan menghentikan instance dengan cepat, karena instance yang dihentikan dihitung terhadap batas instans Anda untuk sementara waktu setelah dihentikan.

Memastikan idempotensi dalam permintaan API Amazon EC2

Saat Anda membuat permintaan API yang bermutasi, permintaan biasanya menampilkan hasil sebelum alur kerja asinkron operasi selesai. Operasi mungkin juga habis atau mengalami masalah server lain sebelum selesai, meskipun permintaan telah mengembalikan hasilnya. Hal ini dapat membuat sulit untuk menentukan apakah permintaan berhasil atau tidak, dan dapat menyebabkan beberapa percobaan ulang untuk memastikan bahwa operasi selesai dengan sukses. Namun, jika permintaan asli dan percobaan ulang berikutnya berhasil, operasi selesai beberapa kali. Ini berarti Anda dapat membuat lebih banyak sumber daya daripada yang Anda inginkan.

Idempotency memastikan bahwa permintaan API selesai tidak lebih dari satu kali. Dengan permintaan idempoten, jika permintaan asli berhasil diselesaikan, percobaan ulang berikutnya berhasil diselesaikan tanpa melakukan tindakan lebih lanjut. Namun, hasilnya mungkin berisi informasi yang diperbarui, seperti status pembuatan saat ini.

Daftar Isi

- [Idempotensi di Amazon EC2](#)
- [RunInstances idempotensi](#)
- [Contoh](#)
- [Coba lagi rekomendasi untuk permintaan idempoten](#)

Idempotensi di Amazon EC2

Tindakan API berikut adalah idempoten secara default, dan tidak memerlukan konfigurasi tambahan. AWS CLI Perintah yang sesuai juga mendukung idempotensi secara default.

Idempoten secara default

- AssociateAddress
- CreateVpnConnection
- DisassociateAddress
- ReplaceNetworkAclAssociation
- TerminateInstances

Tindakan API berikut secara opsional mendukung idempotensi menggunakan token klien. AWS CLI Perintah yang sesuai juga mendukung idempotensi menggunakan token klien. Token klien adalah string unik dan peka huruf besar/kecil hingga 64 karakter ASCII. Untuk membuat permintaan API idempoten menggunakan salah satu tindakan ini, tentukan token klien dalam permintaan. Anda tidak boleh menggunakan kembali token klien yang sama untuk permintaan API lainnya. Jika Anda mencoba lagi permintaan yang berhasil diselesaikan menggunakan token klien yang sama dan parameter yang sama, percobaan ulang berhasil tanpa melakukan tindakan lebih lanjut. Jika Anda mencoba kembali permintaan yang berhasil menggunakan token klien yang sama, tetapi satu atau beberapa parameter berbeda, selain Wilayah atau Zona Ketersediaan, percobaan ulang gagal dengan kesalahan `IdempotentParameterMismatch`.

Idempoten menggunakan token klien

- AllocateHosts
- AllocateIamPoolCidr
- AssociateClientVpnTargetNetwork

- AssociateIpamResourceDiscovery
- AttachVerifiedAccessTrustProvider
- AuthorizeClientVpnIngress
- CopyFpgaImage
- CopyImage
- CreateCapacityReservation
- CreateCapacityReservationFleet
- CreateClientVpnEndpoint
- CreateClientVpnRoute
- CreateEgressOnlyInternetGateway
- CreateFleet
- CreateFlowLogs
- CreateFpgaImage
- CreateInstanceConnectEndpoint
- CreateIpam
- CreateIpamPool
- CreateIpamResourceDiscovery
- CreateIpamScope
- CreateLaunchTemplate
- CreateLaunchTemplateVersion
- CreateManagedPrefixList
- CreateNatGateway
- CreateNetworkAcl
- CreateNetworkInsightsAccessScope
- CreateNetworkInsightsPath
- CreateNetworkInterface
- CreateReplaceRootVolumeTask
- CreateReservedInstancesListing
- CreateRouteTable

- CreateTrafficMirrorFilter
- CreateTrafficMirrorFilterRule
- CreateTrafficMirrorSession
- CreateTrafficMirrorTarget
- CreateVerifiedAccessEndpoint
- CreateVerifiedAccessGroup
- CreateVerifiedAccessInstance
- CreateVerifiedAccessTrustProvider
- CreateVolume
- CreateVpcEndpoint
- CreateVpcEndpointConnectionNotification
- CreateVpcEndpointServiceConfiguration
- DeleteVerifiedAccessEndpoint
- DeleteVerifiedAccessGroup
- DeleteVerifiedAccessInstance
- DeleteVerifiedAccessTrustProvider
- DetachVerifiedAccessTrustProvider
- ExportImage
- ImportImage
- ImportSnapshot
- ModifyInstanceCreditSpecification
- ModifyLaunchTemplate
- ModifyReservedInstances
- ModifyVerifiedAccessEndpoint
- ModifyVerifiedAccessEndpointPolicy
- ModifyVerifiedAccessGroup
- ModifyVerifiedAccessGroupPolicy
- ModifyVerifiedAccessInstance
- ModifyVerifiedAccessInstanceLoggingConfiguration
- ModifyVerifiedAccessTrustProvider

- ProvisionIpamPoolCidr
- PurchaseHostReservation
- RequestSpotFleet
- RequestSpotInstances
- RunInstances
- StartNetworkInsightsAccessScopeAnalysis
- StartNetworkInsightsAnalysis

Jenis idempotensi

- Regional — Permintaan bersifat idempoten di setiap Wilayah. Namun, Anda dapat menggunakan permintaan yang sama, termasuk token klien yang sama, di Wilayah yang berbeda.
- Zonal — Permintaan bersifat idempoten di setiap Availability Zone di suatu Wilayah. Misalnya, jika Anda menentukan token klien yang sama dalam dua panggilan ke `AllocateHosts` Wilayah yang sama, panggilan berhasil jika mereka menentukan nilai yang berbeda untuk `AvailabilityZone` parameter.

RunInstances idempotensi

Tindakan [RunInstances](#) API menggunakan idempotensi Regional dan zona.

Jenis idempotensi yang digunakan bergantung pada cara Anda menentukan Availability Zone dalam permintaan `RunInstances` API Anda. Permintaan menggunakan idempotensi zonal dalam kasus berikut:

- Jika Anda secara eksplisit menentukan Availability Zone menggunakan `AvailabilityZone` parameter dalam tipe data `Penempatan`
- Jika Anda secara implisit menentukan Availability Zone menggunakan parameter `SubnetId`

Jika Anda tidak secara eksplisit atau implisit menentukan Availability Zone, permintaan tersebut menggunakan idempotensi Regional.

Idempotensi zona

Idempotensi zona memastikan bahwa permintaan `RunInstances` API idempoten di setiap Availability Zone di Region. Ini memastikan bahwa permintaan dengan token klien yang sama hanya dapat

diselesaikan sekali dalam setiap Availability Zone di Region. Namun, token klien yang sama dapat digunakan untuk meluncurkan instance di Availability Zone lainnya di Region.

Misalnya, jika Anda mengirim permintaan idempoten untuk meluncurkan instance di `us-east-1a` Availability Zone, dan kemudian menggunakan token klien yang sama dalam permintaan di `us-east-1b` Availability Zone, kami meluncurkan instance di masing-masing Availability Zone tersebut. Jika satu atau beberapa parameter berbeda, percobaan ulang berikutnya dengan token klien yang sama di Availability Zones tersebut berhasil kembali tanpa melakukan tindakan lebih lanjut atau gagal dengan `IdempotentParameterMismatch` kesalahan.

Idempotensi regional

Idempotensi regional memastikan bahwa permintaan `RunInstances` API idempoten di Wilayah. Ini memastikan bahwa permintaan dengan token klien yang sama hanya dapat diselesaikan sekali dalam Wilayah. Namun, permintaan yang sama persis, dengan token klien yang sama, dapat digunakan untuk meluncurkan instance di Wilayah yang berbeda.

Misalnya, jika Anda mengirim permintaan idempoten untuk meluncurkan instance di `us-east-1` Wilayah, dan kemudian menggunakan token klien yang sama dalam permintaan di `eu-west-1` Wilayah, kami meluncurkan instance di masing-masing Wilayah tersebut. Jika satu atau lebih parameter berbeda, percobaan ulang berikutnya dengan token klien yang sama di Wilayah tersebut berhasil kembali tanpa melakukan tindakan lebih lanjut atau gagal dengan `IdempotentParameterMismatch` kesalahan.

Tip

Jika salah satu Availability Zone di Region yang diminta tidak tersedia, `RunInstances` permintaan yang menggunakan idempotensi regional bisa gagal. Untuk memanfaatkan fitur Availability Zone yang ditawarkan oleh AWS infrastruktur, sebaiknya gunakan idempotensi zona saat meluncurkan instance. `RunInstances` permintaan yang menggunakan idempotensi zona dan menargetkan Availability Zone yang tersedia berhasil meskipun Availability Zone lain di Wilayah yang diminta tidak tersedia.

Contoh

AWS CLI contoh perintah

Untuk membuat AWS CLI perintah idempoten, tambahkan opsi. `--client-token`

Contoh 1: Idempotensi

Perintah [allocate-hosts](#) berikut menggunakan idempotency karena menyertakan token klien.

```
aws ec2 allocate-hosts --instance-type m5.large --availability-zone eu-west-1a --  
auto-placement on --quantity 1 --client-token 550e8400-e29b-41d4-a716-446655440000
```

Contoh 2: run-instance idempotensi regional

Perintah [run-instance](#) berikut menggunakan idempotensi regional karena menyertakan token klien tetapi tidak secara eksplisit atau implisit menentukan Availability Zone.

```
aws ec2 run-instances --image-id ami-b232d0db --count 1 --key-name my-key-pair --  
client-token 550e8400-e29b-41d4-a716-446655440000
```

Contoh 3: run-instance idempotensi zona

Perintah [run-instance](#) berikut menggunakan idempotensi zonal karena menyertakan token klien dan Availability Zone yang ditentukan secara eksplisit.

```
aws ec2 run-instances --placement "AvailabilityZone=us-east-1a" --image-id ami-  
b232d0db --count 1 --key-name my-key-pair --client-token 550e8400-e29b-41d4-  
a716-446655440000
```

Contoh permintaan API

Untuk membuat permintaan API idempoten, tambahkan parameter. ClientToken

Contoh 1: Idempotensi

Permintaan [AllocateHosts](#) API berikut menggunakan idempotensi karena menyertakan token klien.

```
https://ec2.amazonaws.com/?Action=AllocateHosts  
&AvailabilityZone=us-east-1b  
&InstanceType=m5.large  
&Quantity=1  
&AutoPlacement=off  
&ClientToken=550e8400-e29b-41d4-a716-446655440000  
&AUTHPARAMS
```

Contoh 2: RunInstances idempotensi regional

Permintaan [RunInstances](#) API berikut menggunakan idempotensi regional karena menyertakan token klien tetapi tidak secara eksplisit atau implisit menentukan Availability Zone.

```
https://ec2.amazonaws.com/?Action=RunInstances
&ImageId=ami-3ac33653
&MaxCount=1
&MinCount=1
&KeyName=my-key-pair
&ClientToken=550e8400-e29b-41d4-a716-446655440000
&AUTHPARAMS
```

Contoh 3: RunInstances idempotensi zonal

Permintaan [RunInstances](#) API berikut menggunakan idempotensi zona karena menyertakan token klien dan Availability Zone yang ditentukan secara eksplisit.

```
https://ec2.amazonaws.com/?Action=RunInstances
&Placement.AvailabilityZone=us-east-1d
&ImageId=ami-3ac33653
&MaxCount=1
&MinCount=1
&KeyName=my-key-pair
&ClientToken=550e8400-e29b-41d4-a716-446655440000
&AUTHPARAMS
```

Coba lagi rekomendasi untuk permintaan idempoten

Tabel berikut menunjukkan beberapa tanggapan umum yang mungkin Anda dapatkan untuk permintaan API idempoten, dan memberikan rekomendasi coba ulang.

Respons	Rekomendasi	Komentar
200 (OK)	Jangan coba lagi	Permintaan asli berhasil diselesaikan. Setiap percobaan ulang berikutnya berhasil kembali.
	Jangan coba lagi	

Respons	Rekomendasi	Komentar
Kode respons 400 seri (kesalahan klien)		<p>Ada masalah dengan permintaan, dari antara yang berikut:</p> <ul style="list-style-type: none"> • Ini termasuk parameter atau kombinasi parameter yang tidak valid. • Ini menggunakan tindakan atau sumber daya yang Anda tidak memiliki izin. • Ini menggunakan sumber daya yang sedang dalam proses mengubah keadaan. <p>Jika permintaan melibatkan sumber daya yang sedang dalam proses mengubah status, mencoba kembali permintaan mungkin berhasil.</p>
Kode respons 500 seri (kesalahan server)	Coba lagi	<p>Kesalahan ini disebabkan oleh masalah AWS sisi server dan umumnya bersifat sementara . Ulangi permintaan dengan strategi backoff yang sesuai.</p>

Minta pembatasan untuk Amazon EC2 API

Amazon EC2 membatasi permintaan API EC2 untuk setiap AWS akun per wilayah. Kami melakukan ini untuk membantu kinerja layanan, dan untuk memastikan penggunaan yang adil untuk semua pelanggan Amazon EC2. Throttling memastikan bahwa panggilan ke Amazon EC2 API tidak melebihi batas permintaan API maksimum yang diizinkan. Panggilan API tunduk pada batas permintaan apakah berasal dari:

- Aplikasi pihak ketiga
- Alat baris perintah
- Konsol Amazon EC2

Jika Anda melebihi batas pembatasan API, Anda mendapatkan kode `RequestLimitExceeded` kesalahan.

Daftar Isi

- [Bagaimana throttling diterapkan](#)
- [Batas pelambatan](#)
- [Pantau pelambatan API](#)
- [Mencoba lagi dan backoff eksponensial](#)
- [Minta kenaikan batas](#)

Bagaimana throttling diterapkan

Amazon EC2 menggunakan [algoritma token bucket](#) untuk mengimplementasikan pelambatan API. Dengan algoritme ini, akun Anda memiliki bucket yang memegang sejumlah tertentu token. Jumlah token dalam bucket mewakili batas throttling Anda pada detik tertentu.

Amazon EC2 mengimplementasikan dua jenis pelambatan API:

Jenis pelambatan API

- [Pembatasan laju permintaan](#)
- [Pembatasan tingkat sumber daya](#)

Pembatasan laju permintaan

Dengan pembatasan tingkat permintaan, Anda dibatasi pada jumlah permintaan API yang Anda buat. Setiap permintaan yang Anda buat menghapus satu token dari ember. Misalnya, ukuran bucket untuk aksi API non-mutating (`Describe*`) adalah 100 token, sehingga Anda dapat membuat hingga 100 `Describe*` permintaan dalam satu detik. Jika Anda melebihi 100 permintaan dalam satu detik, Anda dibatasi dan permintaan yang tersisa dalam detik itu gagal.

Bucket secara otomatis diisi ulang pada tingkat yang ditetapkan. Jika bucket berada di bawah kapasitas maksimumnya, sejumlah token ditambahkan kembali setiap detik hingga mencapai kapasitas maksimumnya. Jika ember penuh saat token isi ulang tiba, mereka dibuang. Bucket tidak dapat menampung lebih dari jumlah token maksimumnya. Misalnya, ukuran bucket untuk aksi API non-mutating (`Describe*`) adalah 100 token, dan tingkat isi ulang adalah 20 token per detik. Jika Anda membuat 100 permintaan `Describe*` API dalam satu detik, bucket segera dikurangi menjadi

nol (0) token. Bucket kemudian diisi ulang dengan 20 token setiap detik, hingga mencapai kapasitas maksimum 100 token. Ini berarti bahwa ember yang sebelumnya kosong mencapai kapasitas maksimumnya setelah 5 detik.

Anda tidak perlu menunggu bucket benar-benar penuh sebelum Anda dapat membuat permintaan API. Anda dapat menggunakan token karena mereka ditambahkan ke bucket. Jika Anda segera menggunakan token isi ulang, ember tidak mencapai kapasitas maksimumnya. Misalnya, ukuran bucket untuk aksi non-mutasi konsol adalah 100 token, dan tingkat isi ulang adalah 10 token per detik. Jika Anda menghabiskan bucket dengan membuat 100 permintaan API dalam satu detik, Anda dapat terus membuat 10 permintaan API per detik. Bucket dapat diisi ulang hingga kapasitas maksimum hanya jika Anda membuat kurang dari 10 permintaan API per detik.

Pembatasan tingkat sumber daya

Beberapa tindakan API, seperti `RunInstances` dan `TerminateInstances`, seperti yang dijelaskan dalam tabel berikut, menggunakan pembatasan laju sumber daya selain pembatasan tingkat permintaan. Tindakan API ini memiliki bucket token sumber daya terpisah yang habis berdasarkan jumlah sumber daya yang dipengaruhi oleh permintaan. Seperti bucket token permintaan, bucket token sumber daya memiliki bucket maksimum yang memungkinkan Anda melakukan burst, dan tingkat isi ulang yang memungkinkan Anda mempertahankan tingkat permintaan yang stabil selama diperlukan. Jika Anda melebihi batas bucket tertentu untuk API, termasuk saat bucket belum diisi ulang untuk mendukung panggilan API berikutnya, tindakan API akan terbatas meskipun Anda belum mencapai batas throttle API total.

Misalnya, ukuran bucket token sumber daya untuk `RunInstances` adalah 1000 token, dan tingkat isi ulang adalah dua token per detik. Oleh karena itu, Anda dapat segera meluncurkan 1000 instans, menggunakan sejumlah permintaan API, seperti satu permintaan untuk 1000 instans atau empat permintaan untuk 250 instans. Setelah bucket token sumber daya kosong, Anda dapat meluncurkan hingga dua instance setiap detik, menggunakan salah satu permintaan untuk dua instance atau dua permintaan untuk satu instance.

Untuk informasi selengkapnya, lihat [Ukuran bucket token sumber daya dan tarif isi ulang](#).

Batas pelambatan

Bagian berikut menjelaskan ukuran bucket token permintaan dan bucket token sumber daya serta tarif isi ulang.

Batas

- [Minta ukuran bucket token dan tarif isi ulang](#)
- [Ukuran bucket token sumber daya dan tarif isi ulang](#)

Minta ukuran bucket token dan tarif isi ulang

Untuk tujuan pembatasan tingkat permintaan, tindakan API dikelompokkan ke dalam kategori berikut:

- Tindakan non-mutasi — Tindakan API yang mengambil data tentang sumber daya. Kategori ini umumnya mencakup semua Describe* tindakan, seperti DescribeRouteTables, DescribeImages, dan DescribeHosts. Tindakan API ini biasanya memiliki batas pembatasan API tertinggi.
- [Tindakan non-mutasi tanpa filter dan tanpa paginasi — Subset spesifik dari tindakan API non-mutasi yang, ketika dipanggil tanpa menentukan pagination atau filter, menggunakan token dari bucket token yang lebih kecil.](#) Disarankan agar Anda menggunakan pagination dan filtering sehingga token dikurangkan dari bucket token standar (lebih besar).
- Tindakan mutasi — Tindakan API yang membuat, memodifikasi, atau menghapus sumber daya. Kategori ini umumnya mencakup semua tindakan API yang tidak dikategorikan sebagai tindakan non-mutasi, seperti CreateVolume, dan ModifyHosts DeleteSnapshot. Tindakan ini memiliki batas pelambatan yang lebih rendah daripada panggilan API yang tidak bermutasi.
- Tindakan intensif sumber daya — Mutasi tindakan API yang memakan waktu paling banyak dan menghabiskan sebagian besar sumber daya untuk diselesaikan. Tindakan ini memiliki batas pelambatan yang lebih rendah daripada tindakan mutasi. Mereka dibatasi secara terpisah dari tindakan mutasi lainnya.
- Tindakan non-mutasi konsol — Tindakan API non-mutasi yang dipanggil dari konsol Amazon EC2. Tindakan API ini dibatasi secara terpisah dari tindakan API non-mutasi lainnya.
- Tindakan yang tidak dikategorikan — Tindakan API ini menerima ukuran bucket token dan tarif isi ulang mereka sendiri, meskipun menurut definisi mereka cocok dalam salah satu kategori lainnya.

Tabel berikut menunjukkan ukuran bucket token permintaan dan tarif isi ulang untuk semua AWS Wilayah.

Kategori aksi API	Tindakan	Kapasitas maksimum bucket	Tingkat isi ulang ember
Tindakan yang tidak bermutasi	<ul style="list-style-type: none"> Describe* Get* 	100	20
Tindakan non-mutasi tanpa filter dan tanpa paginasi	<ul style="list-style-type: none"> DescribeInstances DescribeNetworkInterfaces DescribeVolumes DescribeInstanceStatus DescribeSnapshots DescribeSecurityGroups DescribeSpotInstanceRequests 	50	10
Tindakan bermutasi	Tindakan API yang tidak dikategorikan	200	5

Kategori aksi API	Tindakan	Kapasitas maksimum bucket	Tingkat isi ulang ember
	sebagai tindakan non-mutasi.		

Kategori aksi API	Tindakan	Kapasitas maksimum bucket	Tingkat isi ulang ember
Tindakan intensif sumber daya	<ul style="list-style-type: none"> • AuthorizeSecurityGroupIngress • CancelSpotInstanceRequests • CreateKeyPair • RequestSpotInstances • RevokeSecurityGroupIngress • CreateVpcPeeringConnection • AcceptVpcPeeringConnection • RejectVpcPeeringConnection • DeleteVpcPeeringConnection 	50	5

Kategori aksi API	Tindakan	Kapasitas maksimum bucket	Tingkat isi ulang ember
Tindakan konsol yang tidak bermutasi	<ul style="list-style-type: none"> Describe* Get* 	100	10
Tindakan yang tidak dikategorikan	RunInstances	5	2
	StartInstances	5	2
	CreateVpc Endpoint	4	0,3
	ModifyVpc Endpoint	4	0,3
	DeleteVpc Endpoints	4	0,3
	AcceptVpc EndpointC onnections	10	1
	RejectVpc EndpointC onnections	10	1
	CreateVpc EndpointS erviceCon figuration	10	1
	ModifyVpc EndpointS erviceCon figuration	10	1

Kategori aksi API	Tindakan	Kapasitas maksimum bucket	Tingkat isi ulang ember
	DeleteVpcEndpointServiceConfigurations	10	1
	CreateDefaultVpc	1	1
	CreateDefaultSubnet	1	1
	MoveAddressToVpc	1	1
	RestoreAddressToClassic	1	1
	DescribeMovingAddresses	1	1
	AdvertiseByoipCidr	1	0,1
	ProvisionByoipCidr	1	0.1
	DescribeByoipCidrs	1	0,5
	DeprovisionByoipCidr	1	0.1
	WithdrawByoipCidr	1	0.1

Kategori aksi API	Tindakan	Kapasitas maksimum bucket	Tingkat isi ulang ember
	DescribeReservedInstancesOfferings	10	10
	PurchaseReservedInstancesOffering	5	5
	DescribeSpotFleetRequests	50	3
	DescribeSpotFleetInstances	100	5
	DescribeSpotFleetRequestHistory	100	5
	AssociateEnclaveCertificateIamRole	10	1
	DisassociateEnclaveCertificateIamRole	10	1

Kategori aksi API	Tindakan	Kapasitas maksimum bucket	Tingkat isi ulang ember
	GetAssociatedEnclaveCertificateIamRoles	10	1
	GetConsoleScreenshot	5 per akun 2 per contoh	5 per akun 1 per contoh

Ukuran bucket token sumber daya dan tarif isi ulang

Tabel berikut mencantumkan ukuran bucket token resource dan laju isi ulang untuk tindakan API yang menggunakan pembatasan laju sumber daya.

Tindakan API	Kapasitas maksimum bucket	Tingkat isi ulang ember
RunInstances	1000	2
TerminateInstances	1000	20
StartInstances	1000	2
StopInstances	1000	20

Pantau pelambatan API

Anda dapat menggunakan Amazon CloudWatch untuk memantau panggilan API Amazon EC2 Anda dan untuk mengumpulkan dan melacak metrik seputar pembatasan API. Anda juga dapat membuat alarm untuk memperingatkan Anda ketika Anda hampir mencapai batas pelambatan API. Untuk informasi selengkapnya, lihat [Pantau permintaan API Amazon EC2 menggunakan Amazon CloudWatch](#).

Mencoba lagi dan backoff eksponensial

Aplikasi Anda mungkin perlu mencoba kembali permintaan API. Sebagai contoh:

- Untuk memeriksa pembaruan dalam status sumber daya
- Untuk menghitung sejumlah besar sumber daya (misalnya, semua volume Anda)
- Untuk mencoba lagi permintaan setelah gagal dengan kesalahan server (5xx) atau kesalahan pelambatan

Namun, untuk kesalahan klien (4xx), Anda harus merevisi permintaan untuk memperbaiki masalah sebelum mencoba permintaan lagi.

Perubahan status sumber daya

Sebelum Anda memulai polling untuk memeriksa pembaruan status, berikan waktu permintaan untuk berpotensi selesai. Misalnya, tunggu beberapa menit sebelum memeriksa apakah instans Anda aktif. Saat Anda memulai polling, gunakan interval tidur yang sesuai di antara permintaan berturut-turut untuk menurunkan tingkat permintaan API. Untuk hasil terbaik, gunakan interval tidur yang meningkat atau variabel.

Atau, Anda dapat menggunakan Amazon EventBridge untuk memberi tahu Anda tentang status beberapa sumber daya. Misalnya, Anda dapat menggunakan peristiwa Pemberitahuan Perubahan Status Instans EC2 untuk memberi tahu Anda tentang perubahan status untuk sebuah instans. Untuk informasi selengkapnya, lihat [Mengotomatiskan Amazon EventBridge EC2 menggunakan](#).

Mencoba lagi

Saat Anda perlu melakukan polling atau mencoba ulang permintaan API, sebaiknya gunakan algoritme backoff eksponensial untuk menghitung interval tidur antara panggilan API. Ide di balik backoff eksponensial adalah menggunakan waktu tunggu yang semakin lama antara percobaan ulang untuk respons kesalahan yang berurutan. Anda harus menerapkan interval penundaan maksimum, serta jumlah percobaan ulang maksimum. Anda juga dapat menggunakan jitter (penundaan acak) untuk mencegah tabrakan berturut-turut. Untuk informasi selengkapnya, lihat [Timeout, percobaan ulang, dan backoff](#) dengan jitter.

Setiap AWS SDK mengimplementasikan logika coba ulang otomatis. Untuk informasi selengkapnya, lihat [Mencoba lagi perilaku](#) di AWS SDK dan Panduan Referensi Alat.

Minta kenaikan batas

Anda dapat meminta peningkatan batas pembatasan API untuk Anda. Akun AWS

Untuk meminta akses ke fitur ini

1. [AWS Support Pusat](#) Terbuka.
2. Pilih Buat kasus.
3. Pilih Akun dan penagihan.
4. Untuk Layanan, pilih Info Umum dan Memulai.
5. Untuk Kategori, pilih Menggunakan AWS & Layanan.
6. Pilih Langkah selanjutnya: Informasi tambahan.
7. Untuk Subjek, masukkan **Request an increase in my Amazon EC2 API throttling limits**.
8. Untuk Deskripsi, masukkan **Please increase the API throttling limits for my account. Related page: <https://docs.aws.amazon.com/AWSEC2/latest/APIReference/throttling.html>**. Juga mencakup informasi berikut:
 - Deskripsi kasus penggunaan Anda.
 - Wilayah di mana Anda membutuhkan peningkatan.
 - Jendela satu jam, di UTC, saat pelambatan puncak atau penggunaan terjadi (untuk menghitung batas pelambatan baru).
9. Pilih Langkah selanjutnya: Selesaikan sekarang atau hubungi kami.
10. Pada tab Hubungi kami, pilih bahasa kontak pilihan Anda dan metode kontak.
11. Pilih Kirim.

Buat sumber daya Amazon EC2 menggunakan AWS CLI

Anda dapat membuat dan mengelola resource Amazon EC2 menggunakan AWS Command Line Interface (AWS CLI) di shell baris perintah. AWS CLI Ini menyediakan akses langsung ke API untuk Layanan AWS, seperti Amazon EC2.

Untuk sintaks dan contoh perintah untuk Amazon EC2, [lihat](#) ec2 di AWS CLI Referensi Perintah. Anda juga dapat menemukan contoh-contoh ini di [aws-cli/awscli/examples/ec2](https://github.com/aws-cli/awscli/examples/ec2) di github.

Pelajari lebih lanjut tentang AWS CLI

Untuk mempelajari selengkapnya AWS CLI, lihat sumber daya berikut:

- [AWS Command Line Interface](#)
- [AWS Command Line Interface Panduan Pengguna untuk Versi 2](#)
- [AWS Command Line Interface Panduan Pengguna untuk Versi 1](#)

Buat sumber daya Amazon EC2 menggunakan AWS CloudFormation

Amazon EC2 terintegrasi dengan AWS CloudFormation, layanan yang membantu Anda memodelkan dan menyiapkan AWS sumber daya sehingga Anda dapat menghabiskan lebih sedikit waktu untuk membuat dan mengelola sumber daya dan infrastruktur Anda. Anda membuat templat yang menjelaskan AWS sumber daya yang Anda butuhkan (seperti instance dan subnet), serta menyediakan serta AWS CloudFormation mengonfigurasi sumber daya tersebut untuk Anda.

Bila Anda menggunakan AWS CloudFormation, Anda dapat menggunakan kembali template Anda untuk mengatur sumber daya Amazon EC2 Anda secara konsisten dan berulang kali. Jelaskan sumber daya Anda sekali, lalu sediakan sumber daya yang sama berulang-ulang di beberapa Akun AWS dan Wilayah.

Amazon EC2 dan template AWS CloudFormation

[Untuk menyediakan dan mengonfigurasi sumber daya untuk Amazon EC2 dan layanan terkait, Anda harus memahami AWS CloudFormation templat.](#) Templat adalah file teks dengan format JSON atau YAML. Template ini menjelaskan sumber daya yang akan Anda sediakan di AWS CloudFormation tumpukan Anda. Jika Anda tidak terbiasa dengan JSON atau YAMM, Anda dapat menggunakan AWS CloudFormation Designer untuk membantu Anda memulai dengan template. AWS CloudFormation Untuk informasi lebih lanjut, lihat [Apa itu AWS CloudFormation Desainer?](#) dalam AWS CloudFormation User Guide.

Sumber daya untuk Amazon EC2

Sumber daya komputasi

- [AWS::EC2::CapacityReservation](#)
- [AWS::EC2::CapacityReservationArmada](#)
- [AWS:: EC2:: EC2Fleet](#)
- [AWS:: EC2:: EC2Fleet](#)
- [AWS::EC2::Host](#)
- [AWS::EC2::Instance](#)
- [AWS::EC2::InstanceConnectTitik akhir](#)

- [AWS::EC2::LaunchTemplate](#)
- [AWS::EC2::PlacementGroup](#)
- [AWS::EC2::SpotFleet](#)

Sumber daya jaringan

- [AWS::EC2::CarrierGateway](#)
- [AWS::EC2::ClientVpnAuthorizationRule](#)
- [AWS::EC2::ClientVpnTitik akhir](#)
- [AWS::EC2::ClientVpnRoute](#)
- [AWS::EC2::ClientVpnTargetNetworkAssociation](#)
- [AWS::EC2::CustomerGateway](#)
- [AWS::EC2::DHCPOptions](#)
- [AWS::EC2::EgressOnlyInternetGateway](#)
- [AWS::EC2::EIP](#)
- [AWS::EC2::EIPAssociation](#)
- [AWS::EC2::FlowLog](#)
- [AWS::EC2::GatewayRouteTableAssociation](#)
- [AWS::EC2::InternetGateway](#)
- [AWS::EC2::IPAM](#)
- [AWS::EC2::IPAMLocation](#)
- [AWS::EC2::IPAMPool](#)
- [AWS::EC2::IPAMPoolCidr](#)
- [AWS::EC2::IPAMResourceDiscovery](#)
- [AWS::EC2::IPAMResourceDiscoveryAssociation](#)
- [AWS::EC2::IPAMScope](#)
- [AWS::EC2::LocalGatewayRoute](#)
- [AWS::EC2::LocalGatewayRouteTable](#)
- [AWS::EC2::LocalGatewayRouteTableVirtualInterfaceGroupAssociation](#)
- [AWS::EC2::LocalGatewayRouteTableVPCLAssociation](#)
- [AWS::EC2::NatGateway](#)

- [AWS::EC2::NetworkInterface](#)
- [AWS::EC2::NetworkInsightsAccessScope](#)
- [AWS::EC2::NetworkInsightsAccessScopeAnalysis](#)
- [AWS::EC2::NetworkInsightsAnalisa](#)
- [AWS::EC2::NetworkInsightsJalan](#)
- [AWS::EC2::NetworkInterfaceLampiran](#)
- [AWS::EC2::NetworkInterfacelzin](#)
- [AWS::EC2::NetworkPerformanceMetricSubscription](#)
- [AWS::EC2::PrefixList](#)
- [AWS::EC2::Route](#)
- [AWS::EC2::RouteTable](#)
- [AWS::EC2::Subnet](#)
- [AWS::EC2::SubnetCidrBlok](#)
- [AWS::EC2::SubnetNetworkAclAssociation](#)
- [AWS::EC2::SubnetRouteTableAssociation](#)
- [AWS::EC2::TrafficMirrorFilter](#)
- [AWS::EC2::TrafficMirrorFilterRule](#)
- [AWS::EC2::TrafficMirrorSesi](#)
- [AWS::EC2::TrafficMirrorTarget](#)
- [AWS::EC2::TransitGateway](#)
- [AWS::EC2::TransitGatewayLampiran](#)
- [AWS::EC2::TransitGatewayConnect](#)
- [AWS::EC2::TransitGatewayMulticastDomain](#)
- [AWS::EC2::TransitGatewayMulticastDomainAssociation](#)
- [AWS::EC2::TransitGatewayMulticastGroupMember](#)
- [AWS::EC2::TransitGatewayMulticastGroupSource](#)
- [AWS::EC2::TransitGatewayPeeringAttachment](#)
- [AWS::EC2::TransitGatewayRule](#)
- [AWS::EC2::TransitGatewayRouteTable](#)
- [AWS::EC2::TransitGatewayRouteTableAssociation](#)

- [AWS::EC2::TransitGatewayRouteTablePropagation](#)
- [AWS::EC2::TransitGatewayVpcAttachment](#)
- [AWS::EC2::VPC](#)
- [AWS::EC2::VPC CidrBlock](#)
- [AWS::EC2::VPCDHCP OptionsAssociation](#)
- [AWS::EC2::VPCendPoint](#)
- [AWS::EC2::VPC EndpointConnectionNotification](#)
- [AWS::EC2::VPC EndpointService](#)
- [AWS::EC2::VPC EndpointServicePermissions](#)
- [AWS::EC2::VPC GatewayAttachment](#)
- [AWS::EC2::VPC PeeringConnection](#)
- [AWS::EC2::VPNKoneksi](#)
- [AWS::EC2::VPN ConnectionRoute](#)
- [AWS::EC2::VPNGateway](#)
- [AWS::EC2::VPN GatewayRoutePropagation](#)

Sumber daya keamanan

- [AWS::EC2::KeyPair](#)
- [AWS::EC2::NetworkAcl](#)
- [AWS::EC2::NetworkAclEntri](#)
- [AWS::EC2::SecurityGroup](#)
- [AWS::EC2::SecurityGroupJalan keluar](#)
- [AWS::EC2::SecurityGroupMasuknya](#)
- [AWS::EC2::VerifiedAccessTitik akhir](#)
- [AWS::EC2::VerifiedAccessKelompok](#)
- [AWS::EC2::VerifiedAccessContoh](#)
- [AWS::EC2::VerifiedAccessTrustProvider](#)

Sumber daya penyimpanan

- [AWS::EC2::SnapshotBlockPublicAccess](#)

- [AWS::EC2::Volume](#)
- [AWS::EC2::VolumeAttachment](#)

Pelajari lebih lanjut tentang AWS CloudFormation

Untuk mempelajari selengkapnya AWS CloudFormation, lihat sumber daya berikut:

- [AWS CloudFormation](#)
- [AWS CloudFormation Panduan Pengguna](#)

Membuat sumber daya Amazon EC2 menggunakan SDK AWS

AWS menyediakan kit pengembangan perangkat lunak (SDK) untuk banyak bahasa pemrograman populer. SDK membuat pengembangan lebih efisien dengan menyediakan hal-hal berikut:

- Komponen dan pustaka pra-bangun yang dapat Anda masukkan ke dalam aplikasi Anda
- Alat khusus bahasa, seperti kompiler dan debugger
- Penandatanganan kriptografi permintaan layanan
- Minta percobaan ulang
- Penanganan respons kesalahan

Contoh kode untuk Amazon EC2 API

Contoh kode yang diberikan AWS menunjukkan kepada Anda cara menggunakan API dan menyelesaikan tugas tertentu. Untuk contoh API Amazon EC2, lihat [Contoh kode untuk Amazon EC2](#). Untuk contoh tambahan, lihat [Menemukan contoh kode untuk AWS SDK](#) atau [aws-doc-sdk-examples](#) di github.

Pelajari lebih lanjut tentang AWS SDK

Untuk mempelajari lebih lanjut tentang AWS SDK, lihat sumber daya berikut:

- [AWS Panduan Referensi SDK dan Alat](#)
- [Alat untuk Dibangun AWS](#)
- [Apa itu SDK?](#)

API tingkat rendah untuk Amazon EC2

API tingkat rendah untuk Amazon EC2 adalah antarmuka tingkat protokol untuk Amazon EC2. Saat menggunakan API tingkat rendah, Anda harus memformat setiap permintaan HTTPS dengan benar dan menambahkan tanda tangan digital yang valid ke setiap permintaan. Untuk informasi selengkapnya, lihat [Membuat permintaan ke Amazon EC2 API di Referensi API](#) Amazon EC2. Atau, Anda dapat menggunakan AWS SDK, yang membuat dan menandatangani permintaan atas nama Anda. Untuk informasi selengkapnya, lihat [Menggunakan AWS SDK](#).

Amazon EC2 API terdiri dari tindakan dan tipe data untuk beberapa layanan. Untuk melihat tindakan untuk setiap layanan, lihat halaman berikut di Referensi API Amazon EC2.

- [AWS Client VPN tindakan](#)
- [Tindakan Amazon EBS](#)
- [Tindakan Amazon EC2](#)
- [AWS Network Manager tindakan](#)
- [AWS Tindakan Nitro Enclave](#)
- [AWS Outposts tindakan](#)
- [AWS PrivateLink tindakan](#)
- [Tindakan Recycle Bin](#)
- [AWS Site-to-Site VPN tindakan](#)
- [AWS Transit Gateway tindakan](#)
- [Akses Terverifikasi AWS tindakan](#)
- [Tindakan Impor/Ekspor VM](#)
- [Tindakan Amazon VPC](#)
- [Tindakan Amazon VPC IPAM](#)
- [AWS Wavelength tindakan](#)

Buat kode untuk tindakan konsol Anda menggunakan Konsol-ke-Kode

Konsol-ke-Kode dalam rilis pratinjau untuk Amazon EC2 dan dapat berubah sewaktu-waktu. Hanya tersedia di Wilayah AS Timur (Virginia Utara).

Konsol tersebut menyediakan jalur dengan panduan untuk membuat sumber daya dan menguji prototipe. Jika Anda ingin membuat sumber daya yang sama dalam skala besar, Anda memerlukan kode otomatisasi. Konsol-ke-Kode adalah fitur konsol Amazon EC2 yang dapat membantu Anda memulai dengan kode otomatisasi Anda. Konsol-ke-kode mencatat tindakan konsol Anda, termasuk nilai default dan parameter yang kompatibel. Kemudian menggunakan AI generatif untuk menyarankan kode dalam format pilihan Anda *infrastructure-as-code* (IAC) untuk tindakan yang Anda inginkan. Anda dapat menggunakan kode tersebut sebagai titik awal, sehingga Anda dapat menyesuaikannya agar siap produksi untuk kasus penggunaan spesifik Anda.

Tidak ada biaya tambahan untuk menggunakan Konsol-ke-Kode.

Cara kerjanya

Konsol-ke-Kode dapat membantu Anda memulai dengan kode otomatisasi Anda, sebagai berikut:

1. Anda melakukan tindakan di konsol, seperti meluncurkan instans atau mengaktifkan pemantauan terperinci.
2. Konsol-ke-kode mencatat semua tindakan Anda, termasuk semua pengaturan default dan parameter yang kompatibel, yang disediakan oleh konsol.
3. Anda memilih tindakan yang ingin digunakan dalam skrip otomatisasi Anda. Ini bisa berupa tindakan mutasi atau hanya-baca (tidak bermutasi), atau keduanya.
4. Konsol-ke-kode menghasilkan kode dalam format yang Anda inginkan *infrastructure-as-code* (IAC), misalnya, TypeScript
5. Anda menyalin kode untuk digunakan dalam alat pengembangan kode Anda atau mengunduhnya untuk dibagikan.
6. Anda kemudian menggunakan kode sebagai titik awal untuk skrip otomatisasi Anda. Anda harus memvalidasi bahwa kode tersebut sesuai dengan maksud Anda dan bahwa parameter akan mengonfigurasi sumber daya Anda seperti yang diharapkan. Anda harus menyesuaikan kode agar

siap produksi untuk kasus penggunaan Anda. Setelah puas dengan kode tersebut, Anda dapat menggunakannya dalam skrip otomatisasi Anda.

Untuk petunjuk tentang cara menggunakan Konsol-ke-Kode di konsol Amazon EC2, lihat [Menggunakan Konsol-ke-Kode](#).

Batasan

Pembatasan berikut berlaku saat menggunakan Konsol-ke-Kode.

Wilayah yang Didukung

Saat ini hanya tersedia di Wilayah AS Timur (Virginia Utara).

Format yang didukung

Console-to-Code saat ini dapat menghasilkan infrastructure-as-code (IAC) dalam format kode berikut:

- CDK Java
- CDK Python
- CDK TypeScript
- CloudFormation JSON
- CloudFormation YAMB

Tindakan yang dipertahankan

- Sesi saat ini: Hanya tindakan yang diambil selama sesi saat ini yang ditampilkan di tabel Tindakan yang direkam. Tindakan yang diambil selama sesi sebelumnya tidak dipertahankan.
- Penyegaran browser: Tindakan yang direkam hilang saat Anda menyegarkan tab browser.
- Isolasi tab: Tabel tindakan yang direkam khusus untuk tab browser tempat tindakan diambil. Tindakan yang dilakukan dalam satu tab tidak terlihat di tabel Tindakan yang direkam di tab lain.

Tabel tindakan yang dicatat

Tabel berikut mencantumkan dan menjelaskan kolom dalam tabel Tindakan yang dicatat di konsol Konsol-ke-Kode.

Judul kolom	Deskripsi
Halaman konsol	Halaman konsol tempat tindakan dilakukan.
Operasi	Operasi API.
Tipe	Tipe tindakan. <ul style="list-style-type: none">• Mutasi – Tindakan API yang membuat, memodifikasi, atau menghapus sumber daya.• Hanya baca – Tindakan API yang mengambil data tentang sumber daya (umumnya semua tindakan Describe*).
Perintah CLI	Detail tentang tindakan yang diambil, termasuk parameter dan nilai.
Waktu pembuatan	Waktu saat tindakan diambil.

Menggunakan Konsol-ke-Kode

Gunakan petunjuk berikut untuk membuat kode menggunakan Konsol-ke-Kode di konsol Amazon EC2.

Untuk melihat animasi dari langkah-langkah tersebut, lihat [Lihat animasi: Buat kode menggunakan Konsol-ke-Kode di konsol Amazon EC2](#).

Untuk menghasilkan kode menggunakan Konsol-ke-Kode

1. [Buka konsol Amazon EC2 di Wilayah AS Timur \(Virginia N.\) di https://console.aws.amazon.com/ec2/home?region=us-east-1](https://console.aws.amazon.com/ec2/home?region=us-east-1).


Note

Console-to-Code dalam rilis pratinjau dan saat ini hanya tersedia di Wilayah AS Timur (Virginia N.). Hanya tindakan yang dilakukan di Wilayah ini yang akan dicatat.

2. Gunakan konsol untuk membuat sumber daya dan menguji prototipe. Misalnya, gunakan konsol untuk mengonfigurasi dan meluncurkan instans serta mengaktifkan pemantauan terperinci.


Konsol-ke-Kode mencatat setiap tindakan yang Anda lakukan.

3. Di panel navigasi kanan, pilih Konsol-ke-Kode.
4. Dalam tabel Tindakan yang dicatat, tinjau tindakan yang dicatat, dan putuskan tindakan yang akan disertakan untuk pembuatan kode.
 - Gunakan bidang pencarian untuk memfilter tabel berdasarkan halaman atau tindakan konsol tertentu. Saat Anda mulai mengetik, tabel difilter.
 - Gunakan drop-down Tipe untuk memfilter berdasarkan semua tindakan, tindakan mutasi, atau tindakan hanya-baca.

 Note

Hanya tindakan yang diambil selama sesi saat ini yang terdaftar. Untuk informasi selengkapnya, lihat [Tindakan yang dipertahankan](#).

5. Pilih kotak centang di samping setiap tindakan yang ingin Anda buat kodenya.


 Note

Anda dapat memilih hingga 5 tindakan sekaligus.

6. Pilih tombol Buat kode {code}.

Tombol dilabeli secara default hingga ke format kode yang terakhir dipilih. Untuk memilih format kode yang berbeda, pilih tanda panah di samping tombol.

7. Di bawah Tinjau kode, pilih Salin untuk menyalin kode yang akan digunakan di alat pengembangan Anda atau Unduh untuk mengunduh file agar dapat dibagikan.
8. Gunakan kode sebagai titik awal untuk Anda infrastructure-as-code. Anda harus menyesuaikan kode tersebut agar siap produksi untuk kasus penggunaan Anda.

 Note

Jika Anda menemukan bahwa kode tersebut belum siap produksi, berikan kami umpan balik tentang bagaimana hal itu dapat ditingkatkan (lihat langkah 9 berikut). AWS Support

tidak dapat membantu Anda dengan kode yang dihasilkan atau pengembangan kode khusus Anda.

9. (Opsional) Pilih jempol ke atas atau jempol ke bawah untuk memberi tahu kami jika Konsol-ke-Kode membantu. Jika memilih jempol ke bawah, Anda kemudian dapat memilih Berikan umpan balik untuk memberi tahu kami cara meningkatkan kode agar dapat membantu Anda dengan lebih baik.

Lihat animasi: Buat kode menggunakan Konsol-ke-Kode di konsol Amazon EC2

The screenshot displays the Amazon EC2 console interface. On the left is a navigation sidebar with categories like EC2 Dashboard, Instances, Images, Elastic Block Store, and Network & Security. The main content area is divided into several panels:

- Resources:** A table showing EC2 resources in the US East (N. Virginia) Region.

Resources	
You are using the following Amazon EC2 resources in the US East (N. Virginia) Region:	
Instances (running)	4
Dedicated Hosts	0
Instances	5
Load balancers	0
Security groups	14
Volumes	6
Auto Scaling Groups	0
Elastic IPs	0
Key pairs	5
Placement groups	1
Snapshots	5
- Launch instance:** A panel with a "Launch instance" button and a "Migrate a server" button. Below the buttons is a note: "Note: Your instances will launch in the US East (N. Virginia) Region".
- Service health:** A panel showing the "AWS Health Dashboard" and the current region "US East (N. Virginia)". It includes a "Zones" table:

Zone name	Zone ID
us-east-1a	use1-azZ
- Account attributes:** A panel showing "Default VPC" (vpc-92304aeb) and "Settings" for data protection and security, zones, and console experiments.
- Explore AWS:** A panel with promotional messages for Spot Instances, Amazon GuardDuty Malware Protection, and AWS Graviton2.

Contoh kode untuk Amazon EC2 menggunakan SDK AWS

Contoh kode berikut menunjukkan cara menggunakan Amazon EC2 dengan kit pengembangan AWS perangkat lunak (SDK).

Tindakan adalah kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Meskipun tindakan menunjukkan cara memanggil fungsi layanan individual, Anda dapat melihat tindakan dalam konteks pada skenario terkait dan contoh lintas layanan.

Skenario adalah contoh kode yang menunjukkan cara menyelesaikan tugas tertentu dengan memanggil beberapa fungsi dalam layanan yang sama.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Ayo Memulai

Halo Amazon EC2

Contoh kode berikut ini menunjukkan cara mendapatkan data tentang tipe instans Amazon EC2.

.NET

AWS SDK for .NET

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
namespace EC2Actions;

public class HelloEc2
{
    /// <summary>
    /// HelloEc2 lists the existing security groups for the default users.
```

```
/// </summary>
/// <param name="args">Command line arguments</param>
/// <returns>A Task object.</returns>
static async Task Main(string[] args)
{
    // Set up dependency injection for Amazon Elastic Compute Cloud (Amazon
    EC2).
    using var host =
    Microsoft.Extensions.Hosting.Host.CreateDefaultBuilder(args)
        .ConfigureServices((_, services) =>
            services.AddAWSService<IAmazonEC2>()
                .AddTransient<EC2Wrapper>()
        )
        .Build();

    // Now the client is available for injection.
    var ec2Client = host.Services.GetRequiredService<IAmazonEC2>();

    var request = new DescribeSecurityGroupsRequest
    {
        MaxResults = 10,
    };

    // Retrieve information about up to 10 Amazon EC2 security groups.
    var response = await ec2Client.DescribeSecurityGroupsAsync(request);

    // Now print the security groups returned by the call to
    // DescribeSecurityGroupsAsync.
    Console.WriteLine("Security Groups:");
    response.SecurityGroups.ForEach(group =>
    {
        Console.WriteLine($"Security group: {group.GroupName} ID:
        {group.GroupId}");
    });
}
```

- Untuk detail API, lihat [DescribeSecurityGroups](#) di Referensi AWS SDK for .NET API.

C++

SDK untuk C++

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

Kode untuk file CMake MakeLists C.txt.

```
# Set the minimum required version of CMake for this project.
cmake_minimum_required(VERSION 3.13)

# Set the AWS service components used by this project.
set(SERVICE_COMPONENTS ec2)

# Set this project's name.
project("hello_ec2")

# Set the C++ standard to use to build this target.
# At least C++ 11 is required for the AWS SDK for C++.
set(CMAKE_CXX_STANDARD 11)

# Use the MSVC variable to determine if this is a Windows build.
set(WINDOWS_BUILD ${MSVC})

if (WINDOWS_BUILD) # Set the location where CMake can find the installed
  libraries for the AWS SDK.
  string(REPLACE ";" "/aws-cpp-sdk-all;" SYSTEM_MODULE_PATH
    "${CMAKE_SYSTEM_PREFIX_PATH}/aws-cpp-sdk-all")
  list(APPEND CMAKE_PREFIX_PATH ${SYSTEM_MODULE_PATH})
endif ()

# Find the AWS SDK for C++ package.
find_package(AWSSDK REQUIRED COMPONENTS ${SERVICE_COMPONENTS})

if (WINDOWS_BUILD AND AWSSDK_INSTALL_AS_SHARED_LIBS)
  # Copy relevant AWS SDK for C++ libraries into the current binary directory
  for running and debugging.
```

```
# set(BIN_SUB_DIR "/Debug") # If you are building from the command line, you
may need to uncomment this

                                # and set the proper subdirectory to the
executables' location.

    AWSSDK_CPY_DYN_LIBS(SERVICE_COMPONENTS ""
    ${CMAKE_CURRENT_BINARY_DIR}${BIN_SUB_DIR})
endif ()

add_executable(${PROJECT_NAME}
    hello_ec2.cpp)

target_link_libraries(${PROJECT_NAME}
    ${AWSSDK_LINK_LIBRARIES})
```

Kode untuk file sumber hello_ec2.cpp.

```
#include <aws/core/Aws.h>
#include <aws/ec2/EC2Client.h>
#include <aws/ec2/model/DescribeInstancesRequest.h>
#include <iomanip>
#include <iostream>

/*
 * A "Hello EC2" starter application which initializes an Amazon Elastic Compute
 * Cloud (Amazon EC2) client and describes
 * the Amazon EC2 instances.
 *
 * main function
 *
 * Usage: 'hello_ec2'
 *
 */

int main(int argc, char **argv) {
    Aws::SDKOptions options;
    // Optionally change the log level for debugging.
    // options.loggingOptions.logLevel = Utils::Logging::LogLevel::Debug;
    Aws::InitAPI(options); // Should only be called once.
    int result = 0;
    {
        Aws::Client::ClientConfiguration clientConfig;
```

```
// Optional: Set to the AWS Region (overrides config file).
// clientConfig.region = "us-east-1";

Aws::EC2::EC2Client ec2Client(clientConfig);
Aws::EC2::Model::DescribeInstancesRequest request;
bool header = false;
bool done = false;
while (!done) {
    auto outcome = ec2Client.DescribeInstances(request);
    if (outcome.IsSuccess()) {
        if (!header) {
            std::cout << std::left <<
                std::setw(48) << "Name" <<
                std::setw(20) << "ID" <<
                std::setw(25) << "Ami" <<
                std::setw(15) << "Type" <<
                std::setw(15) << "State" <<
                std::setw(15) << "Monitoring" << std::endl;
            header = true;
        }

        const std::vector<Aws::EC2::Model::Reservation> &reservations =
            outcome.GetResult().GetReservations();

        for (const auto &reservation: reservations) {
            const std::vector<Aws::EC2::Model::Instance> &instances =
                reservation.GetInstances();
            for (const auto &instance: instances) {
                Aws::String instanceStateString =

                Aws::EC2::Model::InstanceStateNameMapper::GetNameForInstanceStateName(
                    instance.GetState().GetName());

                Aws::String typeString =

                Aws::EC2::Model::InstanceTypeMapper::GetNameForInstanceType(
                    instance.GetInstanceType());

                Aws::String monitorString =

                Aws::EC2::Model::MonitoringStateMapper::GetNameForMonitoringState(
                    instance.GetMonitoring().GetState());
                Aws::String name = "Unknown";
            }
        }
    }
}
```

```

        const std::vector<Aws::EC2::Model::Tag> &tags =
instance.GetTags();
        auto nameIter = std::find_if(tags.cbegin(), tags.cend(),
        [](const
Aws::EC2::Model::Tag &tag) {
            return tag.GetKey() ==
"Name";
        });
        if (nameIter != tags.cend()) {
            name = nameIter->GetValue();
        }
        std::cout <<
            std::setw(48) << name <<
            std::setw(20) << instance.GetInstanceId() <<
            std::setw(25) << instance.GetImageId() <<
            std::setw(15) << typeString <<
            std::setw(15) << instanceStateString <<
            std::setw(15) << monitorString << std::endl;
    }
}

    if (!outcome.GetResult().GetNextToken().empty()) {
        request.SetNextToken(outcome.GetResult().GetNextToken());
    } else {
        done = true;
    }
} else {
    std::cerr << "Failed to describe EC2 instances:" <<
        outcome.GetError().GetMessage() << std::endl;
    result = 1;
    break;
}
}
}

    Aws::ShutdownAPI(options); // Should only be called once.
    return result;
}

```

- Untuk detail API, lihat [DescribeSecurityGroups](#) di Referensi AWS SDK for C++ API.

Java

SDK untuk Java 2.x

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
public static void describeSecurityGroups(Ec2Client ec2, String groupId) {
    try {
        DescribeSecurityGroupsRequest request =
DescribeSecurityGroupsRequest.builder()
            .groupIds(groupId)
            .build();

        // Use a paginator.
        DescribeSecurityGroupsIterable listGroups =
ec2.describeSecurityGroupsPaginator(request);
        listGroups.stream()
            .flatMap(r -> r.securityGroups().stream())
            .forEach(group -> System.out
                .println(" Group id: " +group.groupId() + " group name = " +
group.groupName()));

    } catch (Ec2Exception e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
}
```

- Untuk detail API, lihat [DescribeSecurityGroups](#) di Referensi AWS SDK for Java 2.x API.

JavaScript

SDK untuk JavaScript (v3)

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
import { DescribeSecurityGroupsCommand } from "@aws-sdk/client-ec2";

import { client } from "../libs/client.js";

// Call DescribeSecurityGroups and display the result.
export const main = async () => {
  try {
    const { SecurityGroups } = await client.send(
      new DescribeSecurityGroupsCommand({}),
    );

    const securityGroupList = SecurityGroups.slice(0, 9)
      .map((sg) => ` • ${sg.GroupId}: ${sg.GroupName}`)
      .join("\n");

    console.log(
      "Hello, Amazon EC2! Let's list up to 10 of your security groups:",
    );
    console.log(securityGroupList);
  } catch (err) {
    console.error(err);
  }
};
```

- Untuk detail API, lihat [DescribeSecurityGroups](#) di Referensi AWS SDK for JavaScript API.

Kotlin

SDK untuk Kotlin

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
suspend fun describeEC2SecurityGroups(groupId: String) {
    val request =
        DescribeSecurityGroupsRequest {
            groupIds = listOf(groupId)
        }

    Ec2Client { region = "us-west-2" }.use { ec2 ->

        val response = ec2.describeSecurityGroups(request)
        response.securityGroups?.forEach { group ->
            println("Found Security Group with id ${group.groupId}, vpc id
${group.vpcId} and description ${group.description}")
        }
    }
}
```

- Untuk detail API, lihat [DescribeSecurityGroups](#) di AWS SDK untuk referensi API Kotlin.

Python

SDK untuk Python (Boto3)

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
import boto3
```

```
def hello_ec2(ec2_resource):
    """
    Use the AWS SDK for Python (Boto3) to create an Amazon Elastic Compute Cloud
    (Amazon EC2) resource and list the security groups in your account.
    This example uses the default settings specified in your shared credentials
    and config files.

    :param ec2_resource: A Boto3 EC2 ServiceResource object. This object is a
    high-level
                                resource that wraps the low-level EC2 service API.
    """
    print("Hello, Amazon EC2! Let's list up to 10 of your security groups:")
    for sg in ec2_resource.security_groups.limit(10):
        print(f"\t{sg.id}: {sg.group_name}")

if __name__ == "__main__":
    hello_ec2(boto3.resource("ec2"))
```

- Untuk detail API, lihat [DescribeSecurityGroups](#) di AWS SDK for Python (Boto3) Referensi API.

Contoh kode

- [Tindakan untuk Amazon EC2 menggunakan SDK AWS](#)
 - [Gunakan AcceptVpcPeeringConnection dengan AWS SDK atau CLI](#)
 - [Gunakan AllocateAddress dengan AWS SDK atau CLI](#)
 - [Gunakan AllocateHosts dengan AWS SDK atau CLI](#)
 - [Gunakan AssignPrivateIpAddresses dengan AWS SDK atau CLI](#)
 - [Gunakan AssociateAddress dengan AWS SDK atau CLI](#)
 - [Gunakan AssociateDhcpOptions dengan AWS SDK atau CLI](#)
 - [Gunakan AssociateRouteTable dengan AWS SDK atau CLI](#)
 - [Gunakan AttachInternetGateway dengan AWS SDK atau CLI](#)
 - [Gunakan AttachNetworkInterface dengan AWS SDK atau CLI](#)
 - [Gunakan AttachVolume dengan AWS SDK atau CLI](#)

- [Gunakan AttachVpnGateway dengan AWS SDK atau CLI](#)
- [Gunakan AuthorizeSecurityGroupEgress dengan AWS SDK atau CLI](#)
- [Gunakan AuthorizeSecurityGroupIngress dengan AWS SDK atau CLI](#)
- [Gunakan CancelCapacityReservation dengan AWS SDK atau CLI](#)
- [Gunakan CancellImportTask dengan AWS SDK atau CLI](#)
- [Gunakan CancelSpotFleetRequests dengan AWS SDK atau CLI](#)
- [Gunakan CancelSpotInstanceRequests dengan AWS SDK atau CLI](#)
- [Gunakan ConfirmProductInstance dengan AWS SDK atau CLI](#)
- [Gunakan CopyImage dengan AWS SDK atau CLI](#)
- [Gunakan CopySnapshot dengan AWS SDK atau CLI](#)
- [Gunakan CreateCapacityReservation dengan AWS SDK atau CLI](#)
- [Gunakan CreateCustomerGateway dengan AWS SDK atau CLI](#)
- [Gunakan CreateDhcpOptions dengan AWS SDK atau CLI](#)
- [Gunakan CreateFlowLogs dengan AWS SDK atau CLI](#)
- [Gunakan CreateImage dengan AWS SDK atau CLI](#)
- [Gunakan CreateInstanceExportTask dengan AWS SDK atau CLI](#)
- [Gunakan CreateInternetGateway dengan AWS SDK atau CLI](#)
- [Gunakan CreateKeyPair dengan AWS SDK atau CLI](#)
- [Gunakan CreateLaunchTemplate dengan AWS SDK atau CLI](#)
- [Gunakan CreateNetworkAcl dengan AWS SDK atau CLI](#)
- [Gunakan CreateNetworkAclEntry dengan AWS SDK atau CLI](#)
- [Gunakan CreateNetworkInterface dengan AWS SDK atau CLI](#)
- [Gunakan CreatePlacementGroup dengan AWS SDK atau CLI](#)
- [Gunakan CreateRoute dengan AWS SDK atau CLI](#)
- [Gunakan CreateRouteTable dengan AWS SDK atau CLI](#)
- [Gunakan CreateSecurityGroup dengan AWS SDK atau CLI](#)
- [Gunakan CreateSnapshot dengan AWS SDK atau CLI](#)
- [Gunakan CreateSpotDatafeedSubscription dengan AWS SDK atau CLI](#)
- [Gunakan CreateSubnet dengan AWS SDK atau CLI](#)
- [Gunakan CreateTags dengan AWS SDK atau CLI](#)

- [Gunakan CreateVolume dengan AWS SDK atau CLI](#)
- [Gunakan CreateVpc dengan AWS SDK atau CLI](#)
- [Gunakan CreateVpcEndpoint dengan AWS SDK atau CLI](#)
- [Gunakan CreateVpnConnection dengan AWS SDK atau CLI](#)
- [Gunakan CreateVpnConnectionRoute dengan AWS SDK atau CLI](#)
- [Gunakan CreateVpnGateway dengan AWS SDK atau CLI](#)
- [Gunakan DeleteCustomerGateway dengan AWS SDK atau CLI](#)
- [Gunakan DeleteDhcpOptions dengan AWS SDK atau CLI](#)
- [Gunakan DeleteFlowLogs dengan AWS SDK atau CLI](#)
- [Gunakan DeleteInternetGateway dengan AWS SDK atau CLI](#)
- [Gunakan DeleteKeyPair dengan AWS SDK atau CLI](#)
- [Gunakan DeleteLaunchTemplate dengan AWS SDK atau CLI](#)
- [Gunakan DeleteNetworkAcl dengan AWS SDK atau CLI](#)
- [Gunakan DeleteNetworkAclEntry dengan AWS SDK atau CLI](#)
- [Gunakan DeleteNetworkInterface dengan AWS SDK atau CLI](#)
- [Gunakan DeletePlacementGroup dengan AWS SDK atau CLI](#)
- [Gunakan DeleteRoute dengan AWS SDK atau CLI](#)
- [Gunakan DeleteRouteTable dengan AWS SDK atau CLI](#)
- [Gunakan DeleteSecurityGroup dengan AWS SDK atau CLI](#)
- [Gunakan DeleteSnapshot dengan AWS SDK atau CLI](#)
- [Gunakan DeleteSpotDatafeedSubscription dengan AWS SDK atau CLI](#)
- [Gunakan DeleteSubnet dengan AWS SDK atau CLI](#)
- [Gunakan DeleteTags dengan AWS SDK atau CLI](#)
- [Gunakan DeleteVolume dengan AWS SDK atau CLI](#)
- [Gunakan DeleteVpc dengan AWS SDK atau CLI](#)
- [Gunakan DeleteVpnConnection dengan AWS SDK atau CLI](#)
- [Gunakan DeleteVpnConnectionRoute dengan AWS SDK atau CLI](#)
- [Gunakan DeleteVpnGateway dengan AWS SDK atau CLI](#)
- [Gunakan DeregisterImage dengan AWS SDK atau CLI](#)
- [Gunakan DescribeAccountAttributes dengan AWS SDK atau CLI](#)

- [Gunakan DescribeAddresses dengan AWS SDK atau CLI](#)
- [Gunakan DescribeAvailabilityZones dengan AWS SDK atau CLI](#)
- [Gunakan DescribeBundleTasks dengan AWS SDK atau CLI](#)
- [Gunakan DescribeCapacityReservations dengan AWS SDK atau CLI](#)
- [Gunakan DescribeCustomerGateways dengan AWS SDK atau CLI](#)
- [Gunakan DescribeDhcpOptions dengan AWS SDK atau CLI](#)
- [Gunakan DescribeFlowLogs dengan AWS SDK atau CLI](#)
- [Gunakan DescribeHostReservationOfferings dengan AWS SDK atau CLI](#)
- [Gunakan DescribeHosts dengan AWS SDK atau CLI](#)
- [Gunakan DescribeIamInstanceProfileAssociations dengan AWS SDK atau CLI](#)
- [Gunakan DescribeIdFormat dengan AWS SDK atau CLI](#)
- [Gunakan DescribeIdentityIdFormat dengan AWS SDK atau CLI](#)
- [Gunakan DescribeImageAttribute dengan AWS SDK atau CLI](#)
- [Gunakan DescribeImages dengan AWS SDK atau CLI](#)
- [Gunakan DescribeImportImageTasks dengan AWS SDK atau CLI](#)
- [Gunakan DescribeImportSnapshotTasks dengan AWS SDK atau CLI](#)
- [Gunakan DescribeInstanceAttribute dengan AWS SDK atau CLI](#)
- [Gunakan DescribeInstanceStatus dengan AWS SDK atau CLI](#)
- [Gunakan DescribeInstanceTypes dengan AWS SDK atau CLI](#)
- [Gunakan DescribeInstances dengan AWS SDK atau CLI](#)
- [Gunakan DescribeInternetGateways dengan AWS SDK atau CLI](#)
- [Gunakan DescribeKeyPairs dengan AWS SDK atau CLI](#)
- [Gunakan DescribeNetworkAcls dengan AWS SDK atau CLI](#)
- [Gunakan DescribeNetworkInterfaceAttribute dengan AWS SDK atau CLI](#)
- [Gunakan DescribeNetworkInterfaces dengan AWS SDK atau CLI](#)
- [Gunakan DescribePlacementGroups dengan AWS SDK atau CLI](#)
- [Gunakan DescribePrefixLists dengan AWS SDK atau CLI](#)
- [Gunakan DescribeRegions dengan AWS SDK atau CLI](#)
- [Gunakan DescribeRouteTables dengan AWS SDK atau CLI](#)
- [Gunakan DescribeScheduledInstanceAvailability dengan AWS SDK atau CLI](#)

- [Gunakan DescribeScheduledInstances dengan AWS SDK atau CLI](#)
- [Gunakan DescribeSecurityGroups dengan AWS SDK atau CLI](#)
- [Gunakan DescribeSnapshotAttribute dengan AWS SDK atau CLI](#)
- [Gunakan DescribeSnapshots dengan AWS SDK atau CLI](#)
- [Gunakan DescribeSpotDatafeedSubscription dengan AWS SDK atau CLI](#)
- [Gunakan DescribeSpotFleetInstances dengan AWS SDK atau CLI](#)
- [Gunakan DescribeSpotFleetRequestHistory dengan AWS SDK atau CLI](#)
- [Gunakan DescribeSpotFleetRequests dengan AWS SDK atau CLI](#)
- [Gunakan DescribeSpotInstanceRequests dengan AWS SDK atau CLI](#)
- [Gunakan DescribeSpotPriceHistory dengan AWS SDK atau CLI](#)
- [Gunakan DescribeSubnets dengan AWS SDK atau CLI](#)
- [Gunakan DescribeTags dengan AWS SDK atau CLI](#)
- [Gunakan DescribeVolumeAttribute dengan AWS SDK atau CLI](#)
- [Gunakan DescribeVolumeStatus dengan AWS SDK atau CLI](#)
- [Gunakan DescribeVolumes dengan AWS SDK atau CLI](#)
- [Gunakan DescribeVpcAttribute dengan AWS SDK atau CLI](#)
- [Gunakan DescribeVpcClassicLink dengan AWS SDK atau CLI](#)
- [Gunakan DescribeVpcClassicLinkDnsSupport dengan AWS SDK atau CLI](#)
- [Gunakan DescribeVpcEndpointServices dengan AWS SDK atau CLI](#)
- [Gunakan DescribeVpcEndpoints dengan AWS SDK atau CLI](#)
- [Gunakan DescribeVpcs dengan AWS SDK atau CLI](#)
- [Gunakan DescribeVpnConnections dengan AWS SDK atau CLI](#)
- [Gunakan DescribeVpnGateways dengan AWS SDK atau CLI](#)
- [Gunakan DetachInternetGateway dengan AWS SDK atau CLI](#)
- [Gunakan DetachNetworkInterface dengan AWS SDK atau CLI](#)
- [Gunakan DetachVolume dengan AWS SDK atau CLI](#)
- [Gunakan DetachVpnGateway dengan AWS SDK atau CLI](#)
- [Gunakan DisableVgwRoutePropagation dengan AWS SDK atau CLI](#)
- [Gunakan DisableVpcClassicLink dengan AWS SDK atau CLI](#)
- [Gunakan DisableVpcClassicLinkDnsSupport dengan AWS SDK atau CLI](#)

- [Gunakan DisassociateAddress dengan AWS SDK atau CLI](#)
- [Gunakan DisassociateRouteTable dengan AWS SDK atau CLI](#)
- [Gunakan EnableVgwRoutePropagation dengan AWS SDK atau CLI](#)
- [Gunakan EnableVolumelo dengan AWS SDK atau CLI](#)
- [Gunakan EnableVpcClassicLink dengan AWS SDK atau CLI](#)
- [Gunakan EnableVpcClassicLinkDnsSupport dengan AWS SDK atau CLI](#)
- [Gunakan GetConsoleOutput dengan AWS SDK atau CLI](#)
- [Gunakan GetHostReservationPurchasePreview dengan AWS SDK atau CLI](#)
- [Gunakan GetPasswordData dengan AWS SDK atau CLI](#)
- [Gunakan ImportImage dengan AWS SDK atau CLI](#)
- [Gunakan ImportKeyPair dengan AWS SDK atau CLI](#)
- [Gunakan ImportSnapshot dengan AWS SDK atau CLI](#)
- [Gunakan ModifyCapacityReservation dengan AWS SDK atau CLI](#)
- [Gunakan ModifyHosts dengan AWS SDK atau CLI](#)
- [Gunakan ModifyIdFormat dengan AWS SDK atau CLI](#)
- [Gunakan ModifyImageAttribute dengan AWS SDK atau CLI](#)
- [Gunakan ModifyInstanceAttribute dengan AWS SDK atau CLI](#)
- [Gunakan ModifyInstanceCreditSpecification dengan AWS SDK atau CLI](#)
- [Gunakan ModifyNetworkInterfaceAttribute dengan AWS SDK atau CLI](#)
- [Gunakan ModifyReservedInstances dengan AWS SDK atau CLI](#)
- [Gunakan ModifySnapshotAttribute dengan AWS SDK atau CLI](#)
- [Gunakan ModifySpotFleetRequest dengan AWS SDK atau CLI](#)
- [Gunakan ModifySubnetAttribute dengan AWS SDK atau CLI](#)
- [Gunakan ModifyVolumeAttribute dengan AWS SDK atau CLI](#)
- [Gunakan ModifyVpcAttribute dengan AWS SDK atau CLI](#)
- [Gunakan MonitorInstances dengan AWS SDK atau CLI](#)
- [Gunakan MoveAddressToVpc dengan AWS SDK atau CLI](#)
- [Gunakan PurchaseHostReservation dengan AWS SDK atau CLI](#)
- [Gunakan PurchaseScheduledInstances dengan AWS SDK atau CLI](#)
- [Gunakan RebootInstances dengan AWS SDK atau CLI](#)

- [Gunakan RegisterImage dengan AWS SDK atau CLI](#)
- [Gunakan RejectVpcPeeringConnection dengan AWS SDK atau CLI](#)
- [Gunakan ReleaseAddress dengan AWS SDK atau CLI](#)
- [Gunakan ReleaseHosts dengan AWS SDK atau CLI](#)
- [Gunakan ReplacelamInstanceProfileAssociation dengan AWS SDK atau CLI](#)
- [Gunakan ReplaceNetworkAclAssociation dengan AWS SDK atau CLI](#)
- [Gunakan ReplaceNetworkAclEntry dengan AWS SDK atau CLI](#)
- [Gunakan ReplaceRoute dengan AWS SDK atau CLI](#)
- [Gunakan ReplaceRouteTableAssociation dengan AWS SDK atau CLI](#)
- [Gunakan ReportInstanceStatus dengan AWS SDK atau CLI](#)
- [Gunakan RequestSpotFleet dengan AWS SDK atau CLI](#)
- [Gunakan RequestSpotInstances dengan AWS SDK atau CLI](#)
- [Gunakan ResetImageAttribute dengan AWS SDK atau CLI](#)
- [Gunakan ResetInstanceAttribute dengan AWS SDK atau CLI](#)
- [Gunakan ResetNetworkInterfaceAttribute dengan AWS SDK atau CLI](#)
- [Gunakan ResetSnapshotAttribute dengan AWS SDK atau CLI](#)
- [Gunakan RevokeSecurityGroupEgress dengan AWS SDK atau CLI](#)
- [Gunakan RevokeSecurityGroupIngress dengan AWS SDK atau CLI](#)
- [Gunakan RunInstances dengan AWS SDK atau CLI](#)
- [Gunakan RunScheduledInstances dengan AWS SDK atau CLI](#)
- [Gunakan StartInstances dengan AWS SDK atau CLI](#)
- [Gunakan StopInstances dengan AWS SDK atau CLI](#)
- [Gunakan TerminatInstances dengan AWS SDK atau CLI](#)
- [Gunakan UnassignPrivateIpAddresses dengan AWS SDK atau CLI](#)
- [Gunakan UnmonitorInstances dengan AWS SDK atau CLI](#)
- [Skenario untuk Amazon EC2 menggunakan SDK AWS](#)
 - [Membangun dan mengelola layanan tangguh menggunakan SDK AWS](#)
 - [Memulai instans Amazon EC2 menggunakan SDK AWS](#)

Tindakan untuk Amazon EC2 menggunakan SDK AWS

Contoh kode berikut menunjukkan cara melakukan tindakan Amazon EC2 individual dengan AWS SDK. Kutipan ini memanggil API Amazon EC2 dan merupakan kutipan kode dari program yang lebih besar yang harus dijalankan dalam konteks. Setiap contoh menyertakan tautan ke GitHub, di mana Anda dapat menemukan instruksi untuk mengatur dan menjalankan kode.

Contoh berikut hanya mencakup tindakan yang paling umum digunakan. Untuk daftar selengkapnya, [lihat Referensi API Amazon Elastic Compute Cloud \(Amazon EC2\)](#).

Contoh

- [Gunakan AcceptVpcPeeringConnection dengan AWS SDK atau CLI](#)
- [Gunakan AllocateAddress dengan AWS SDK atau CLI](#)
- [Gunakan AllocateHosts dengan AWS SDK atau CLI](#)
- [Gunakan AssignPrivateIpAddresses dengan AWS SDK atau CLI](#)
- [Gunakan AssociateAddress dengan AWS SDK atau CLI](#)
- [Gunakan AssociateDhcpOptions dengan AWS SDK atau CLI](#)
- [Gunakan AssociateRouteTable dengan AWS SDK atau CLI](#)
- [Gunakan AttachInternetGateway dengan AWS SDK atau CLI](#)
- [Gunakan AttachNetworkInterface dengan AWS SDK atau CLI](#)
- [Gunakan AttachVolume dengan AWS SDK atau CLI](#)
- [Gunakan AttachVpnGateway dengan AWS SDK atau CLI](#)
- [Gunakan AuthorizeSecurityGroupEgress dengan AWS SDK atau CLI](#)
- [Gunakan AuthorizeSecurityGroupIngress dengan AWS SDK atau CLI](#)
- [Gunakan CancelCapacityReservation dengan AWS SDK atau CLI](#)
- [Gunakan CancellImportTask dengan AWS SDK atau CLI](#)
- [Gunakan CancelSpotFleetRequests dengan AWS SDK atau CLI](#)
- [Gunakan CancelSpotInstanceRequests dengan AWS SDK atau CLI](#)
- [Gunakan ConfirmProductInstance dengan AWS SDK atau CLI](#)
- [Gunakan CopyImage dengan AWS SDK atau CLI](#)
- [Gunakan CopySnapshot dengan AWS SDK atau CLI](#)
- [Gunakan CreateCapacityReservation dengan AWS SDK atau CLI](#)

- [Gunakan CreateCustomerGateway dengan AWS SDK atau CLI](#)
- [Gunakan CreateDhcpOptions dengan AWS SDK atau CLI](#)
- [Gunakan CreateFlowLogs dengan AWS SDK atau CLI](#)
- [Gunakan CreateImage dengan AWS SDK atau CLI](#)
- [Gunakan CreateInstanceExportTask dengan AWS SDK atau CLI](#)
- [Gunakan CreateInternetGateway dengan AWS SDK atau CLI](#)
- [Gunakan CreateKeyPair dengan AWS SDK atau CLI](#)
- [Gunakan CreateLaunchTemplate dengan AWS SDK atau CLI](#)
- [Gunakan CreateNetworkAcl dengan AWS SDK atau CLI](#)
- [Gunakan CreateNetworkAclEntry dengan AWS SDK atau CLI](#)
- [Gunakan CreateNetworkInterface dengan AWS SDK atau CLI](#)
- [Gunakan CreatePlacementGroup dengan AWS SDK atau CLI](#)
- [Gunakan CreateRoute dengan AWS SDK atau CLI](#)
- [Gunakan CreateRouteTable dengan AWS SDK atau CLI](#)
- [Gunakan CreateSecurityGroup dengan AWS SDK atau CLI](#)
- [Gunakan CreateSnapshot dengan AWS SDK atau CLI](#)
- [Gunakan CreateSpotDatafeedSubscription dengan AWS SDK atau CLI](#)
- [Gunakan CreateSubnet dengan AWS SDK atau CLI](#)
- [Gunakan CreateTags dengan AWS SDK atau CLI](#)
- [Gunakan CreateVolume dengan AWS SDK atau CLI](#)
- [Gunakan CreateVpc dengan AWS SDK atau CLI](#)
- [Gunakan CreateVpcEndpoint dengan AWS SDK atau CLI](#)
- [Gunakan CreateVpnConnection dengan AWS SDK atau CLI](#)
- [Gunakan CreateVpnConnectionRoute dengan AWS SDK atau CLI](#)
- [Gunakan CreateVpnGateway dengan AWS SDK atau CLI](#)
- [Gunakan DeleteCustomerGateway dengan AWS SDK atau CLI](#)
- [Gunakan DeleteDhcpOptions dengan AWS SDK atau CLI](#)
- [Gunakan DeleteFlowLogs dengan AWS SDK atau CLI](#)
- [Gunakan DeleteInternetGateway dengan AWS SDK atau CLI](#)

- [Gunakan DeleteKeyPair dengan AWS SDK atau CLI](#)
- [Gunakan DeleteLaunchTemplate dengan AWS SDK atau CLI](#)
- [Gunakan DeleteNetworkAcl dengan AWS SDK atau CLI](#)
- [Gunakan DeleteNetworkAclEntry dengan AWS SDK atau CLI](#)
- [Gunakan DeleteNetworkInterface dengan AWS SDK atau CLI](#)
- [Gunakan DeletePlacementGroup dengan AWS SDK atau CLI](#)
- [Gunakan DeleteRoute dengan AWS SDK atau CLI](#)
- [Gunakan DeleteRouteTable dengan AWS SDK atau CLI](#)
- [Gunakan DeleteSecurityGroup dengan AWS SDK atau CLI](#)
- [Gunakan DeleteSnapshot dengan AWS SDK atau CLI](#)
- [Gunakan DeleteSpotDatafeedSubscription dengan AWS SDK atau CLI](#)
- [Gunakan DeleteSubnet dengan AWS SDK atau CLI](#)
- [Gunakan DeleteTags dengan AWS SDK atau CLI](#)
- [Gunakan DeleteVolume dengan AWS SDK atau CLI](#)
- [Gunakan DeleteVpc dengan AWS SDK atau CLI](#)
- [Gunakan DeleteVpnConnection dengan AWS SDK atau CLI](#)
- [Gunakan DeleteVpnConnectionRoute dengan AWS SDK atau CLI](#)
- [Gunakan DeleteVpnGateway dengan AWS SDK atau CLI](#)
- [Gunakan DeregisterImage dengan AWS SDK atau CLI](#)
- [Gunakan DescribeAccountAttributes dengan AWS SDK atau CLI](#)
- [Gunakan DescribeAddresses dengan AWS SDK atau CLI](#)
- [Gunakan DescribeAvailabilityZones dengan AWS SDK atau CLI](#)
- [Gunakan DescribeBundleTasks dengan AWS SDK atau CLI](#)
- [Gunakan DescribeCapacityReservations dengan AWS SDK atau CLI](#)
- [Gunakan DescribeCustomerGateways dengan AWS SDK atau CLI](#)
- [Gunakan DescribeDhcpOptions dengan AWS SDK atau CLI](#)
- [Gunakan DescribeFlowLogs dengan AWS SDK atau CLI](#)
- [Gunakan DescribeHostReservationOfferings dengan AWS SDK atau CLI](#)
- [Gunakan DescribeHosts dengan AWS SDK atau CLI](#)

- [Gunakan DescribeInstanceProfileAssociations dengan AWS SDK atau CLI](#)
- [Gunakan DescribeIdFormat dengan AWS SDK atau CLI](#)
- [Gunakan DescribeIdentityIdFormat dengan AWS SDK atau CLI](#)
- [Gunakan DescribeImageAttribute dengan AWS SDK atau CLI](#)
- [Gunakan DescribeImages dengan AWS SDK atau CLI](#)
- [Gunakan DescribeImportImageTasks dengan AWS SDK atau CLI](#)
- [Gunakan DescribeImportSnapshotTasks dengan AWS SDK atau CLI](#)
- [Gunakan DescribeInstanceAttribute dengan AWS SDK atau CLI](#)
- [Gunakan DescribeInstanceStatus dengan AWS SDK atau CLI](#)
- [Gunakan DescribeInstanceTypes dengan AWS SDK atau CLI](#)
- [Gunakan DescribeInstances dengan AWS SDK atau CLI](#)
- [Gunakan DescribeInternetGateways dengan AWS SDK atau CLI](#)
- [Gunakan DescribeKeyPairs dengan AWS SDK atau CLI](#)
- [Gunakan DescribeNetworkAcls dengan AWS SDK atau CLI](#)
- [Gunakan DescribeNetworkInterfaceAttribute dengan AWS SDK atau CLI](#)
- [Gunakan DescribeNetworkInterfaces dengan AWS SDK atau CLI](#)
- [Gunakan DescribePlacementGroups dengan AWS SDK atau CLI](#)
- [Gunakan DescribePrefixLists dengan AWS SDK atau CLI](#)
- [Gunakan DescribeRegions dengan AWS SDK atau CLI](#)
- [Gunakan DescribeRouteTables dengan AWS SDK atau CLI](#)
- [Gunakan DescribeScheduledInstanceAvailability dengan AWS SDK atau CLI](#)
- [Gunakan DescribeScheduledInstances dengan AWS SDK atau CLI](#)
- [Gunakan DescribeSecurityGroups dengan AWS SDK atau CLI](#)
- [Gunakan DescribeSnapshotAttribute dengan AWS SDK atau CLI](#)
- [Gunakan DescribeSnapshots dengan AWS SDK atau CLI](#)
- [Gunakan DescribeSpotDatafeedSubscription dengan AWS SDK atau CLI](#)
- [Gunakan DescribeSpotFleetInstances dengan AWS SDK atau CLI](#)
- [Gunakan DescribeSpotFleetRequestHistory dengan AWS SDK atau CLI](#)
- [Gunakan DescribeSpotFleetRequests dengan AWS SDK atau CLI](#)

- [Gunakan DescribeSpotInstanceRequests dengan AWS SDK atau CLI](#)
- [Gunakan DescribeSpotPriceHistory dengan AWS SDK atau CLI](#)
- [Gunakan DescribeSubnets dengan AWS SDK atau CLI](#)
- [Gunakan DescribeTags dengan AWS SDK atau CLI](#)
- [Gunakan DescribeVolumeAttribute dengan AWS SDK atau CLI](#)
- [Gunakan DescribeVolumeStatus dengan AWS SDK atau CLI](#)
- [Gunakan DescribeVolumes dengan AWS SDK atau CLI](#)
- [Gunakan DescribeVpcAttribute dengan AWS SDK atau CLI](#)
- [Gunakan DescribeVpcClassicLink dengan AWS SDK atau CLI](#)
- [Gunakan DescribeVpcClassicLinkDnsSupport dengan AWS SDK atau CLI](#)
- [Gunakan DescribeVpcEndpointServices dengan AWS SDK atau CLI](#)
- [Gunakan DescribeVpcEndpoints dengan AWS SDK atau CLI](#)
- [Gunakan DescribeVpcs dengan AWS SDK atau CLI](#)
- [Gunakan DescribeVpnConnections dengan AWS SDK atau CLI](#)
- [Gunakan DescribeVpnGateways dengan AWS SDK atau CLI](#)
- [Gunakan DetachInternetGateway dengan AWS SDK atau CLI](#)
- [Gunakan DetachNetworkInterface dengan AWS SDK atau CLI](#)
- [Gunakan DetachVolume dengan AWS SDK atau CLI](#)
- [Gunakan DetachVpnGateway dengan AWS SDK atau CLI](#)
- [Gunakan DisableVgwRoutePropagation dengan AWS SDK atau CLI](#)
- [Gunakan DisableVpcClassicLink dengan AWS SDK atau CLI](#)
- [Gunakan DisableVpcClassicLinkDnsSupport dengan AWS SDK atau CLI](#)
- [Gunakan DisassociateAddress dengan AWS SDK atau CLI](#)
- [Gunakan DisassociateRouteTable dengan AWS SDK atau CLI](#)
- [Gunakan EnableVgwRoutePropagation dengan AWS SDK atau CLI](#)
- [Gunakan EnableVolumelo dengan AWS SDK atau CLI](#)
- [Gunakan EnableVpcClassicLink dengan AWS SDK atau CLI](#)
- [Gunakan EnableVpcClassicLinkDnsSupport dengan AWS SDK atau CLI](#)
- [Gunakan GetConsoleOutput dengan AWS SDK atau CLI](#)

- [Gunakan GetHostReservationPurchasePreview dengan AWS SDK atau CLI](#)
- [Gunakan GetPasswordData dengan AWS SDK atau CLI](#)
- [Gunakan ImportImage dengan AWS SDK atau CLI](#)
- [Gunakan ImportKeyPair dengan AWS SDK atau CLI](#)
- [Gunakan ImportSnapshot dengan AWS SDK atau CLI](#)
- [Gunakan ModifyCapacityReservation dengan AWS SDK atau CLI](#)
- [Gunakan ModifyHosts dengan AWS SDK atau CLI](#)
- [Gunakan ModifyIdFormat dengan AWS SDK atau CLI](#)
- [Gunakan ModifyImageAttribute dengan AWS SDK atau CLI](#)
- [Gunakan ModifyInstanceAttribute dengan AWS SDK atau CLI](#)
- [Gunakan ModifyInstanceCreditSpecification dengan AWS SDK atau CLI](#)
- [Gunakan ModifyNetworkInterfaceAttribute dengan AWS SDK atau CLI](#)
- [Gunakan ModifyReservedInstances dengan AWS SDK atau CLI](#)
- [Gunakan ModifySnapshotAttribute dengan AWS SDK atau CLI](#)
- [Gunakan ModifySpotFleetRequest dengan AWS SDK atau CLI](#)
- [Gunakan ModifySubnetAttribute dengan AWS SDK atau CLI](#)
- [Gunakan ModifyVolumeAttribute dengan AWS SDK atau CLI](#)
- [Gunakan ModifyVpcAttribute dengan AWS SDK atau CLI](#)
- [Gunakan MonitorInstances dengan AWS SDK atau CLI](#)
- [Gunakan MoveAddressToVpc dengan AWS SDK atau CLI](#)
- [Gunakan PurchaseHostReservation dengan AWS SDK atau CLI](#)
- [Gunakan PurchaseScheduledInstances dengan AWS SDK atau CLI](#)
- [Gunakan RebootInstances dengan AWS SDK atau CLI](#)
- [Gunakan RegisterImage dengan AWS SDK atau CLI](#)
- [Gunakan RejectVpcPeeringConnection dengan AWS SDK atau CLI](#)
- [Gunakan ReleaseAddress dengan AWS SDK atau CLI](#)
- [Gunakan ReleaseHosts dengan AWS SDK atau CLI](#)
- [Gunakan ReplacelamInstanceProfileAssociation dengan AWS SDK atau CLI](#)
- [Gunakan ReplaceNetworkAclAssociation dengan AWS SDK atau CLI](#)
- [Gunakan ReplaceNetworkAclEntry dengan AWS SDK atau CLI](#)

- [Gunakan ReplaceRoute dengan AWS SDK atau CLI](#)
- [Gunakan ReplaceRouteTableAssociation dengan AWS SDK atau CLI](#)
- [Gunakan ReportInstanceStatus dengan AWS SDK atau CLI](#)
- [Gunakan RequestSpotFleet dengan AWS SDK atau CLI](#)
- [Gunakan RequestSpotInstances dengan AWS SDK atau CLI](#)
- [Gunakan ResetImageAttribute dengan AWS SDK atau CLI](#)
- [Gunakan ResetInstanceAttribute dengan AWS SDK atau CLI](#)
- [Gunakan ResetNetworkInterfaceAttribute dengan AWS SDK atau CLI](#)
- [Gunakan ResetSnapshotAttribute dengan AWS SDK atau CLI](#)
- [Gunakan RevokeSecurityGroupEgress dengan AWS SDK atau CLI](#)
- [Gunakan RevokeSecurityGroupIngress dengan AWS SDK atau CLI](#)
- [Gunakan RunInstances dengan AWS SDK atau CLI](#)
- [Gunakan RunScheduledInstances dengan AWS SDK atau CLI](#)
- [Gunakan StartInstances dengan AWS SDK atau CLI](#)
- [Gunakan StopInstances dengan AWS SDK atau CLI](#)
- [Gunakan TerminateInstances dengan AWS SDK atau CLI](#)
- [Gunakan UnassignPrivateIpAddresses dengan AWS SDK atau CLI](#)
- [Gunakan UnmonitorInstances dengan AWS SDK atau CLI](#)

Gunakan **AcceptVpcPeeringConnection** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `AcceptVpcPeeringConnection`.

CLI

AWS CLI

Untuk menerima koneksi peering VPC

Contoh ini menerima permintaan koneksi peering VPC yang ditentukan.

Perintah:

```
aws ec2 accept-vpc-peering-connection --vpc-peering-connection-id pcx-1a2b3c4d
```

Output:

```
{
  "VpcPeeringConnection": {
    "Status": {
      "Message": "Provisioning",
      "Code": "provisioning"
    },
    "Tags": [],
    "AccepterVpcInfo": {
      "OwnerId": "444455556666",
      "VpcId": "vpc-44455566",
      "CidrBlock": "10.0.1.0/28"
    },
    "VpcPeeringConnectionId": "pcx-1a2b3c4d",
    "RequesterVpcInfo": {
      "OwnerId": "444455556666",
      "VpcId": "vpc-111abc45",
      "CidrBlock": "10.0.0.0/28"
    }
  }
}
```

- Untuk detail API, lihat [AcceptVpcPeeringConnection](#) di Referensi AWS CLI Perintah.

PowerShell**Alat untuk PowerShell**

Contoh 1: Contoh ini menyetujui pcx-1dfad234b56ff78be yang diminta VpcPeeringConnectionId

```
Approve-EC2VpcPeeringConnection -VpcPeeringConnectionId pcx-1dfad234b56ff78be
```

Output:

```
AccepterVpcInfo      : Amazon.EC2.Model.VpcPeeringConnectionVpcInfo
ExpirationTime       : 1/1/0001 12:00:00 AM
RequesterVpcInfo     : Amazon.EC2.Model.VpcPeeringConnectionVpcInfo
Status                : Amazon.EC2.Model.VpcPeeringConnectionStateReason
Tags                  : {}
```

```
VpcPeeringConnectionId : pcx-1dfad234b56ff78be
```

- Untuk detail API, lihat [AcceptVpcPeeringConnection](#) di Referensi AWS Tools for PowerShell Cmdlet.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **AllocateAddress** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `AllocateAddress`.

Contoh tindakan adalah kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Anda dapat melihat tindakan ini dalam konteks dalam contoh kode berikut:

- [Memulai instans](#)

.NET

AWS SDK for .NET

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
/// <summary>
/// Allocate an Elastic IP address.
/// </summary>
/// <returns>The allocation Id of the allocated address.</returns>
public async Task<string> AllocateAddress()
{
    var request = new AllocateAddressRequest();

    var response = await _amazonEC2.AllocateAddressAsync(request);
    return response.AllocationId;
}
```

- Untuk detail API, lihat [AllocateAddress](#) di Referensi AWS SDK for .NET API.

Bash

AWS CLI dengan skrip Bash

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
#####
# function ec2_allocate_address
#
# This function allocates an Elastic IP address for use with Amazon Elastic
# Compute Cloud (Amazon EC2) instances in a specific AWS Region.
#
# Parameters:
#     -d domain - The domain for the Elastic IP address (either 'vpc' or
#     'standard').
#
# Returns:
#     The allocated Elastic IP address, or an error message if the operation
#     fails.
# And:
#     0 - If successful.
#     1 - If it fails.
#
#####
function ec2_allocate_address() {
    local domain response

    # Function to display usage information
    function usage() {
        echo "function ec2_allocate_address"
        echo "Allocates an Elastic IP address for use with Amazon Elastic Compute
        Cloud (Amazon EC2) instances in a specific AWS Region."
        echo "  -d domain - The domain for the Elastic IP address (either 'vpc' or
        'standard')."
    }
}
```



```
    echo ""
}

# Parse the command-line arguments
while getopts "d:h" option; do
    case "${option}" in
        d) domain="${OPTARG}" ;;
        h)
            usage
            return 0
            ;;
        \?)
            echo "Invalid parameter"
            usage
            return 1
            ;;
    esac
done
export OPTIND=1

# Validate the input parameters
if [[ -z "$domain" ]]; then
    errecho "ERROR: You must provide a domain with the -d parameter (either 'vpc'
or 'standard')."
    return 1
fi

if [[ "$domain" != "vpc" && "$domain" != "standard" ]]; then
    errecho "ERROR: Invalid domain value. Must be either 'vpc' or 'standard'."
    return 1
fi

# Allocate the Elastic IP address
response=$(aws ec2 allocate-address \
    --domain "$domain" \
    --query "[PublicIp,AllocationId]" \
    --output text) || {
    aws_cli_error_log ${?}
    errecho "ERROR: AWS reports allocate-address operation failed."
    errecho "$response"
    return 1
}

echo "$response"
```

```

return 0
}

```

Fungsi utilitas yang digunakan dalam contoh ini.

```

#####
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
#####
function errecho() {
    printf "%s\n" "$*" 1>&2
}

#####
# function aws_cli_error_log()
#
# This function is used to log the error messages from the AWS CLI.
#
# The function expects the following argument:
#     $1 - The error code returned by the AWS CLI.
#
# Returns:
#     0: - Success.
#
#####
function aws_cli_error_log() {
    local err_code=$1
    errecho "Error code : $err_code"
    if [ "$err_code" == 1 ]; then
        errecho " One or more S3 transfers failed."
    elif [ "$err_code" == 2 ]; then
        errecho " Command line failed to parse."
    elif [ "$err_code" == 130 ]; then
        errecho " Process received SIGINT."
    elif [ "$err_code" == 252 ]; then
        errecho " Command syntax invalid."
    elif [ "$err_code" == 253 ]; then
        errecho " The system environment or configuration was invalid."
    elif [ "$err_code" == 254 ]; then
        errecho " The service returned an error."
    elif [ "$err_code" == 255 ]; then

```

```
errecho " 255 is a catch-all error."  
fi  
  
return 0  
}
```

- Untuk detail API, lihat [AllocateAddress](#) di Referensi AWS CLI Perintah.

C++

SDK untuk C++

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
Aws::EC2::EC2Client ec2Client(clientConfiguration);  
  
Aws::EC2::Model::AllocateAddressRequest request;  
request.SetDomain(Aws::EC2::Model::DomainType::vpc);  
  
const Aws::EC2::Model::AllocateAddressOutcome outcome =  
    ec2Client.AllocateAddress(request);  
if (!outcome.IsSuccess()) {  
    std::cerr << "Failed to allocate Elastic IP address:" <<  
        outcome.GetError().GetMessage() << std::endl;  
    return false;  
}  
  
allocationId = outcome.GetResult().GetAllocationId();
```

- Untuk detail API, lihat [AllocateAddress](#) di Referensi AWS SDK for C++ API.

CLI

AWS CLI

Contoh 1: Untuk mengalokasikan alamat IP Elastis dari kolam alamat Amazon

Contoh `allocate-address` berikut mengalokasikan alamat IP Elastis. Amazon EC2 memilih alamat dari kolam alamat Amazon.

```
aws ec2 allocate-address
```

Output:

```
{
  "PublicIp": "70.224.234.241",
  "AllocationId": "eipalloc-01435ba59eEXAMPLE",
  "PublicIpv4Pool": "amazon",
  "NetworkBorderGroup": "us-west-2",
  "Domain": "vpc"
}
```

Untuk informasi selengkapnya, lihat [Alamat IP Elastis](#) di Panduan Pengguna Amazon EC2.

Contoh 2: Untuk mengalokasikan alamat IP Elastis dan mengaitkannya dengan grup batas jaringan

Contoh `allocate-address` berikut mengalokasikan alamat IP Elastis dan mengaitkannya dengan grup batas jaringan tertentu.

```
aws ec2 allocate-address \
  --network-border-group us-west-2-lax-1
```

Output:

```
{
  "PublicIp": "70.224.234.241",
  "AllocationId": "eipalloc-e03dd489ceEXAMPLE",
  "PublicIpv4Pool": "amazon",
  "NetworkBorderGroup": "us-west-2-lax-1",
  "Domain": "vpc"
}
```

Untuk informasi selengkapnya, lihat [Alamat IP Elastis](#) di Panduan Pengguna Amazon EC2.

Contoh 3: Untuk mengalokasikan alamat IP Elastis dari kolam alamat milik Anda

Contoh `allocate-address` berikut mengalokasikan alamat IP Elastis dari kolam alamat yang Anda bawa ke akun Amazon Web Services. Amazon EC2 memilih alamat dari kolam alamat.

```
aws ec2 allocate-address \  
  --public-ipv4-pool ipv4pool-ec2-1234567890abcdef0
```

Output:

```
{  
  "AllocationId": "eipalloc-02463d08ceEXAMPLE",  
  "NetworkBorderGroup": "us-west-2",  
  "CustomerOwnedIp": "18.218.95.81",  
  "CustomerOwnedIpv4Pool": "ipv4pool-ec2-1234567890abcdef0",  
  "Domain": "vpc"  
  "NetworkBorderGroup": "us-west-2",  
}
```

Untuk informasi selengkapnya, lihat [Alamat IP Elastis](#) di Panduan Pengguna Amazon EC2.

- Untuk detail API, lihat [AllocateAddress](#) di Referensi AWS CLI Perintah.

Java

SDK untuk Java 2.x

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
public static String allocateAddress(Ec2Client ec2) {  
    try {  
        AllocateAddressRequest allocateRequest =  
        AllocateAddressRequest.builder()
```

```
        .domain(DomainType.VPC)
        .build();

        AllocateAddressResponse allocateResponse =
ec2.allocateAddress(allocateRequest);
        return allocateResponse.allocationId();

    } catch (Ec2Exception e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
    return "";
}
```

- Untuk detail API, lihat [AllocateAddress](#) di Referensi AWS SDK for Java 2.x API.

JavaScript

SDK untuk JavaScript (v3)

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
import { AllocateAddressCommand } from "@aws-sdk/client-ec2";

import { client } from "../libs/client.js";

export const main = async () => {
    const command = new AllocateAddressCommand({});

    try {
        const { AllocationId, PublicIp } = await client.send(command);
        console.log("A new IP address has been allocated to your account:");
        console.log(`ID: ${AllocationId} Public IP: ${PublicIp}`);
        console.log(
            "You can view your IP addresses in the AWS Management Console for Amazon EC2. Look under Network & Security > Elastic IPs",
        );
    }
};
```

```
    } catch (err) {  
        console.error(err);  
    }  
};
```

- Untuk detail API, lihat [AllocateAddress](#) di Referensi AWS SDK for JavaScript API.

Kotlin

SDK untuk Kotlin

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
suspend fun getAllocateAddress(instanceIdVal: String?): String? {  
    val allocateRequest =  
        AllocateAddressRequest {  
            domain = DomainType.Vpc  
        }  
  
    Ec2Client { region = "us-west-2" }.use { ec2 ->  
        val allocateResponse = ec2.allocateAddress(allocateRequest)  
        val allocationIdVal = allocateResponse.allocationId  
  
        val request =  
            AssociateAddressRequest {  
                instanceId = instanceIdVal  
                allocationId = allocationIdVal  
            }  
  
        val associateResponse = ec2.associateAddress(request)  
        return associateResponse.associationId  
    }  
}
```

- Untuk detail API, lihat [AllocateAddress](#) di AWS SDK untuk referensi API Kotlin.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini mengalokasikan alamat IP Elastis untuk digunakan dengan instance di VPC.

```
New-EC2Address -Domain Vpc
```

Output:

AllocationId	Domain	PublicIp
-----	-----	-----
eipalloc-12345678	vpc	198.51.100.2

Contoh 2: Contoh ini mengalokasikan alamat IP Elastis untuk digunakan dengan instance di EC2-Classic.

```
New-EC2Address
```

Output:

AllocationId	Domain	PublicIp
-----	-----	-----
	standard	203.0.113.17

- Untuk detail API, lihat [AllocateAddress](#) di Referensi AWS Tools for PowerShell Cmdlet.

Python

SDK untuk Python (Boto3)

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
class ElasticIpWrapper:
```



```
"""Encapsulates Amazon Elastic Compute Cloud (Amazon EC2) Elastic IP address
actions."""

def __init__(self, ec2_resource, elastic_ip=None):
    """
    :param ec2_resource: A Boto3 Amazon EC2 resource. This high-level
resource
                           is used to create additional high-level objects
                           that wrap low-level Amazon EC2 service actions.
    :param elastic_ip: A Boto3 VpcAddress object. This is a high-level object
that
                           wraps Elastic IP actions.
    """
    self.ec2_resource = ec2_resource
    self.elastic_ip = elastic_ip

    @classmethod
    def from_resource(cls):
        ec2_resource = boto3.resource("ec2")
        return cls(ec2_resource)

    def allocate(self):
        """
        Allocates an Elastic IP address that can be associated with an Amazon EC2
address
        instance. By using an Elastic IP address, you can keep the public IP
        constant even when you restart the associated instance.

        :return: The newly created Elastic IP object. By default, the address is
not
                   associated with any instance.
        """
        try:
            response =
self.ec2_resource.meta.client.allocate_address(Domain="vpc")
            self.elastic_ip =
self.ec2_resource.VpcAddress(response["AllocationId"])
        except ClientError as err:
            logger.error(
                "Couldn't allocate Elastic IP. Here's why: %s: %s",
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
```

```
        raise
    else:
        return self.elastic_ip
```

- Untuk detail API, lihat [AllocateAddress](#) di AWS SDK for Python (Boto3) Referensi API.

Ruby

SDK untuk Ruby

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
# Creates an Elastic IP address in Amazon Virtual Private Cloud (Amazon VPC).
#
# @param ec2_client [Aws::EC2::Client] An initialized EC2 client.
# @return [String] The allocation ID corresponding to the Elastic IP address.
# @example
#   puts allocate_elastic_ip_address(Aws::EC2::Client.new(region: 'us-west-2'))
def allocate_elastic_ip_address(ec2_client)
  response = ec2_client.allocate_address(domain: "vpc")
  return response.allocation_id
rescue StandardError => e
  puts "Error allocating Elastic IP address: #{e.message}"
  return "Error"
end
```

- Untuk detail API, lihat [AllocateAddress](#) di Referensi AWS SDK for Ruby API.

SAP ABAP

SDK untuk SAP ABAP

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
TRY.  
    oo_result = lo_ec2->allocateaddress( iv_domain = 'vpc' ).    " oo_result  
is returned for testing purposes. "  
    MESSAGE 'Allocated an Elastic IP address.' TYPE 'I'.  
    CATCH /aws1/cx_rt_service_generic INTO DATA(lo_exception).  
    DATA(lv_error) = |"{ lo_exception->av_err_code }" - { lo_exception-  
>av_err_msg }|.  
    MESSAGE lv_error TYPE 'E'.  
ENDTRY.
```

- Untuk detail API, lihat [AllocateAddress](#) di AWS SDK untuk referensi SAP ABAP API.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **AllocateHosts** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `AllocateHosts`.

CLI

AWS CLI

Contoh 1: Untuk mengalokasikan Host Khusus

`allocate-hosts` Contoh berikut mengalokasikan satu Host Khusus di `eu-west-1a` Availability Zone, tempat Anda dapat meluncurkan `m5.large` instance. Secara default, Host Khusus hanya menerima peluncuran instans target, dan tidak mendukung pemulihan host.

```
aws ec2 allocate-hosts \  
  --instance-type m5.large \  
  --availability-zone eu-west-1a \  
  --quantity 1
```

Output:

```
{  
  "HostIds": [  
    "h-07879acf49EXAMPLE"  
  ]  
}
```

Contoh 2: Untuk mengalokasikan Host Khusus dengan penempatan otomatis dan pemulihan host diaktifkan

`allocate-hosts` Contoh berikut mengalokasikan satu Host Khusus di `eu-west-1a` Availability Zone dengan penempatan otomatis dan pemulihan host diaktifkan.

```
aws ec2 allocate-hosts \  
  --instance-type m5.large \  
  --availability-zone eu-west-1a \  
  --auto-placement on \  
  --host-recovery on \  
  --quantity 1
```

Output:

```
{  
  "HostIds": [  
    "h-07879acf49EXAMPLE"  
  ]  
}
```

Contoh 3: Untuk mengalokasikan Host Khusus dengan tag

`allocate-hosts` Contoh berikut mengalokasikan satu Host Khusus dan menerapkan tag dengan kunci bernama `purpose` dan nilai `production`

```
aws ec2 allocate-hosts \  
  --tag-key purpose
```

```
--instance-type m5.large \  
--availability-zone eu-west-1a \  
--quantity 1 \  
--tag-specifications 'ResourceType=dedicated-  
host,Tags={Key=purpose,Value=production}'
```

Output:

```
{  
  "HostIds": [  
    "h-07879acf49EXAMPLE"  
  ]  
}
```

Untuk informasi selengkapnya, lihat [Mengalokasikan Host Khusus](#) di Panduan Pengguna Amazon Elastic Compute Cloud untuk Instans Linux.

- Untuk detail API, lihat [AllocateHosts](#) di Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini mengalokasikan Host Khusus ke akun Anda untuk jenis instans dan zona ketersediaan yang diberikan

```
New-EC2Host -AutoPlacement on -AvailabilityZone eu-west-1b -InstanceType  
m4.xlarge -Quantity 1
```

Output:

```
h-01e23f4cd567890f3
```

- Untuk detail API, lihat [AllocateHosts](#) di Referensi AWS Tools for PowerShell Cmdlet.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan `AssignPrivateIpAddresses` dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `AssignPrivateIpAddresses`.

CLI

AWS CLI

Untuk menetapkan alamat IP pribadi sekunder tertentu antarmuka jaringan

Contoh ini memberikan alamat IP pribadi sekunder yang ditentukan ke antarmuka jaringan yang ditentukan. Jika perintah berhasil, tidak ada output yang akan ditampilkan.

Perintah:

```
aws ec2 assign-private-ip-addresses --network-interface-id eni-e5aa89a3 --private-ip-addresses 10.0.0.82
```

Untuk menetapkan alamat IP pribadi sekunder yang Amazon EC2 pilih ke antarmuka jaringan

Contoh ini memberikan dua alamat IP pribadi sekunder ke antarmuka jaringan yang ditentukan. Amazon EC2 secara otomatis menetapkan alamat IP ini dari alamat IP yang tersedia dalam rentang blok CIDR dari subnet yang terkait dengan antarmuka jaringan. Jika perintah berhasil, tidak ada output yang akan ditampilkan.

Perintah:

```
aws ec2 assign-private-ip-addresses --network-interface-id eni-e5aa89a3 --secondary-private-ip-address-count 2
```

- Untuk detail API, lihat [AssignPrivateIpAddresses](#) di Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini memberikan alamat IP pribadi sekunder yang ditentukan ke antarmuka jaringan yang ditentukan.

```
Register-EC2PrivateIpAddress -NetworkInterfaceId eni-1a2b3c4d -PrivateIpAddress 10.0.0.82
```

Contoh 2: Contoh ini membuat dua alamat IP pribadi sekunder dan menentukannya ke antarmuka jaringan yang ditentukan.

```
Register-EC2PrivateIpAddress -NetworkInterfaceId eni-1a2b3c4d -  
SecondaryPrivateIpAddressCount 2
```

- Untuk detail API, lihat [AssignPrivateIpAddresses](#) di Referensi AWS Tools for PowerShell Cmdlet.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **AssociateAddress** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `AssociateAddress`.

Contoh tindakan adalah kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Anda dapat melihat tindakan ini dalam konteks dalam contoh kode berikut:

- [Memulai instans](#)

.NET

AWS SDK for .NET

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
/// <summary>  
/// Associate an Elastic IP address to an EC2 instance.  
/// </summary>  
/// <param name="allocationId">The allocation Id of an Elastic IP address.</  
param>  
/// <param name="instanceId">The instance Id of the EC2 instance to  
/// associate the address with.</param>  
/// <returns>The association Id that represents
```

```

    /// the association of the Elastic IP address with an instance.</returns>
    public async Task<string> AssociateAddress(string allocationId, string
instanceId)
    {
        var request = new AssociateAddressRequest
        {
            AllocationId = allocationId,
            InstanceId = instanceId
        };

        var response = await _amazonEC2.AssociateAddressAsync(request);
        return response.AssociationId;
    }

```

- Untuk detail API, lihat [AssociateAddress](#) di Referensi AWS SDK for .NET API.

Bash

AWS CLI dengan skrip Bash

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```

#####
# function ec2_associate_address
#
# This function associates an Elastic IP address with an Amazon Elastic Compute
Cloud (Amazon EC2) instance.
#
# Parameters:
#     -a allocation_id - The allocation ID of the Elastic IP address to
associate.
#     -i instance_id - The ID of the EC2 instance to associate the Elastic IP
address with.
#
# Returns:
#     0 - If successful.
#     1 - If it fails.

```



```
#
#####
function ec2_associate_address() {
    local allocation_id instance_id response

    # Function to display usage information
    function usage() {
        echo "function ec2_associate_address"
        echo "Associates an Elastic IP address with an Amazon Elastic Compute Cloud
(Amazon EC2) instance."
        echo " -a allocation_id - The allocation ID of the Elastic IP address to
associate."
        echo " -i instance_id - The ID of the EC2 instance to associate the Elastic
IP address with."
        echo ""
    }

    # Parse the command-line arguments
    while getopts "a:i:h" option; do
        case "${option}" in
            a) allocation_id="${OPTARG}" ;;
            i) instance_id="${OPTARG}" ;;
            h)
                usage
                return 0
                ;;
            \?)
                echo "Invalid parameter"
                usage
                return 1
                ;;
        esac
    done
    export OPTIND=1

    # Validate the input parameters
    if [[ -z "$allocation_id" ]]; then
        errecho "ERROR: You must provide an allocation ID with the -a parameter."
        return 1
    fi

    if [[ -z "$instance_id" ]]; then
        errecho "ERROR: You must provide an instance ID with the -i parameter."
        return 1
    fi
}
```

```

fi

# Associate the Elastic IP address
response=$(aws ec2 associate-address \
  --allocation-id "$allocation_id" \
  --instance-id "$instance_id" \
  --query "AssociationId" \
  --output text) || {
  aws_cli_error_log ${?}
  errecho "ERROR: AWS reports associate-address operation failed."
  errecho "$response"
  return 1
}

echo "$response"
return 0
}

```

Fungsi utilitas yang digunakan dalam contoh ini.

```

#####
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
#####
function errecho() {
  printf "%s\n" "$*" 1>&2
}

#####
# function aws_cli_error_log()
#
# This function is used to log the error messages from the AWS CLI.
#
# The function expects the following argument:
#     $1 - The error code returned by the AWS CLI.
#
# Returns:
#     0: - Success.
#
#####
function aws_cli_error_log() {

```

```
local err_code=$1
errecho "Error code : $err_code"
if [ "$err_code" == 1 ]; then
    errecho " One or more S3 transfers failed."
elif [ "$err_code" == 2 ]; then
    errecho " Command line failed to parse."
elif [ "$err_code" == 130 ]; then
    errecho " Process received SIGINT."
elif [ "$err_code" == 252 ]; then
    errecho " Command syntax invalid."
elif [ "$err_code" == 253 ]; then
    errecho " The system environment or configuration was invalid."
elif [ "$err_code" == 254 ]; then
    errecho " The service returned an error."
elif [ "$err_code" == 255 ]; then
    errecho " 255 is a catch-all error."
fi

return 0
}
```

- Untuk detail API, lihat [AssociateAddress](#) di Referensi AWS CLI Perintah.

C++

SDK untuk C++

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
Aws::EC2::EC2Client ec2Client(clientConfiguration);

Aws::EC2::Model::AssociateAddressRequest associate_request;
associate_request.SetInstanceId(instanceId);
associate_request.SetAllocationId(allocationId);

const Aws::EC2::Model::AssociateAddressOutcome associate_outcome =
    ec2Client.AssociateAddress(associate_request);
```

```
if (!associate_outcome.IsSuccess()) {
    std::cerr << "Failed to associate Elastic IP address " << allocationId
              << " with instance " << instanceId << ":" <<
              associate_outcome.GetError().GetMessage() << std::endl;
    return false;
}

std::cout << "Successfully associated Elastic IP address " << allocationId
          << " with instance " << instanceId << std::endl;
```

- Untuk detail API, lihat [AssociateAddress](#) di Referensi AWS SDK for C++ API.

CLI

AWS CLI

Untuk mengaitkan alamat IP Elastis di EC2-Classic

Contoh ini mengaitkan alamat IP Elastis dengan instans di EC2-Classic. Jika perintah berhasil, tidak ada output yang akan ditampilkan.

Perintah:

```
aws ec2 associate-address --instance-id i-07ffe74c7330ebf53 --public-ip
198.51.100.0
```

Untuk mengaitkan alamat IP Elastis di EC2-VPC

Contoh ini mengaitkan alamat IP Elastis dengan instans di VPC.

Perintah:

```
aws ec2 associate-address --instance-id i-0b263919b6498b123 --allocation-id
eipalloc-64d5890a
```

Output:

```
{
  "AssociationId": "eipassoc-2bebb745"
```

```
}
```

Contoh ini mengaitkan alamat IP Elastis dengan antarmuka jaringan.

Perintah:

```
aws ec2 associate-address --allocation-id eipalloc-64d5890a --network-interface-id eni-1a2b3c4d
```

Contoh ini mengaitkan IP Elastis dengan alamat IP privat yang terkait dengan antarmuka jaringan.

Perintah:

```
aws ec2 associate-address --allocation-id eipalloc-64d5890a --network-interface-id eni-1a2b3c4d --private-ip-address 10.0.0.85
```

- Untuk detail API, lihat [AssociateAddress](#) di Referensi AWS CLI Perintah.

Java

SDK untuk Java 2.x

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
public static String associateAddress(Ec2Client ec2, String instanceId,
String allocationId) {
    try {
        AssociateAddressRequest associateRequest =
AssociateAddressRequest.builder()
            .instanceId(instanceId)
            .allocationId(allocationId)
            .build();

        AssociateAddressResponse associateResponse =
ec2.associateAddress(associateRequest);
```

```
        return associateResponse.associationId();

    } catch (Ec2Exception e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
    return "";
}
```

- Untuk detail API, lihat [AssociateAddress](#) di Referensi AWS SDK for Java 2.x API.

JavaScript

SDK untuk JavaScript (v3)

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
import { AssociateAddressCommand } from "@aws-sdk/client-ec2";

import { client } from "../libs/client.js";

export const main = async () => {
    // You need to allocate an Elastic IP address before associating it with an
    // instance.
    // You can do that with the AllocateAddressCommand.
    const allocationId = "ALLOCATION_ID";
    // You need to create an EC2 instance before an IP address can be associated
    // with it.
    // You can do that with the RunInstancesCommand.
    const instanceId = "INSTANCE_ID";
    const command = new AssociateAddressCommand({
        AllocationId: allocationId,
        InstanceId: instanceId,
    });

    try {
        const { AssociationId } = await client.send(command);
    }
}
```

```
console.log(
  `Address with allocation ID ${allocationId} is now associated with instance
  ${instanceId}.`,
  `The association ID is ${AssociationId}.`,
);
} catch (err) {
  console.error(err);
}
};
```

- Untuk detail API, lihat [AssociateAddress](#) di Referensi AWS SDK for JavaScript API.

Kotlin

SDK untuk Kotlin

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
suspend fun associateAddressSc(
  instanceIdVal: String?,
  allocationIdVal: String?,
): String? {
  val associateRequest =
    AssociateAddressRequest {
      instanceId = instanceIdVal
      allocationId = allocationIdVal
    }

  Ec2Client { region = "us-west-2" }.use { ec2 ->
    val associateResponse = ec2.associateAddress(associateRequest)
    return associateResponse.associationId
  }
}
```

- Untuk detail API, lihat [AssociateAddress](#) di AWS SDK untuk referensi API Kotlin.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini mengaitkan alamat IP Elastis yang ditentukan dengan instance yang ditentukan dalam VPC.

```
C:\> Register-EC2Address -InstanceId i-12345678 -AllocationId eipalloc-12345678
```

Output:

```
eipassoc-12345678
```

Contoh 2: Contoh ini mengaitkan alamat IP Elastis yang ditentukan dengan instance yang ditentukan di EC2-Classic.

```
C:\> Register-EC2Address -InstanceId i-12345678 -PublicIp 203.0.113.17
```

- Untuk detail API, lihat [AssociateAddress](#) di Referensi AWS Tools for PowerShell Cmdlet.

Python

SDK untuk Python (Boto3)

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
class ElasticIpWrapper:
    """Encapsulates Amazon Elastic Compute Cloud (Amazon EC2) Elastic IP address
    actions."""

    def __init__(self, ec2_resource, elastic_ip=None):
        """
        :param ec2_resource: A Boto3 Amazon EC2 resource. This high-level
        resource
                                is used to create additional high-level objects
                                that wrap low-level Amazon EC2 service actions.
```



```

        :param elastic_ip: A Boto3 VpcAddress object. This is a high-level object
that
        wraps Elastic IP actions.
        """
        self.ec2_resource = ec2_resource
        self.elastic_ip = elastic_ip

    @classmethod
    def from_resource(cls):
        ec2_resource = boto3.resource("ec2")
        return cls(ec2_resource)

    def associate(self, instance):
        """
        Associates an Elastic IP address with an instance. When this association
is
        created, the Elastic IP's public IP address is immediately used as the
public
        IP address of the associated instance.

        :param instance: A Boto3 Instance object. This is a high-level object
that wraps
        Amazon EC2 instance actions.
        :return: A response that contains the ID of the association.
        """
        if self.elastic_ip is None:
            logger.info("No Elastic IP to associate.")
            return

        try:
            response = self.elastic_ip.associate(InstanceId=instance.id)
        except ClientError as err:
            logger.error(
                "Couldn't associate Elastic IP %s with instance %s. Here's why:
%s: %s",
                self.elastic_ip.allocation_id,
                instance.id,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
        return response

```

- Untuk detail API, lihat [AssociateAddress](#) di AWS SDK for Python (Boto3) Referensi API.

Ruby

SDK untuk Ruby

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
# Associates an Elastic IP address with an Amazon Elastic Compute Cloud
# (Amazon EC2) instance.
#
# Prerequisites:
#
# - The allocation ID corresponding to the Elastic IP address.
# - The Amazon EC2 instance.
#
# @param ec2_client [Aws::EC2::Client] An initialized EC2 client.
# @param allocation_id [String] The ID of the allocation corresponding to
#   the Elastic IP address.
# @param instance_id [String] The ID of the instance.
# @return [String] The association ID corresponding to the association of the
#   Elastic IP address to the instance.
# @example
#   puts allocate_elastic_ip_address(
#     Aws::EC2::Client.new(region: 'us-west-2'),
#     'eipalloc-04452e528a66279EX',
#     'i-033c48ef067af3dEX')
def associate_elastic_ip_address_with_instance(
  ec2_client,
  allocation_id,
  instance_id
)
  response = ec2_client.associate_address(
    allocation_id: allocation_id,
    instance_id: instance_id,
  )
end
```

```

return response.association_id
rescue StandardError => e
  puts "Error associating Elastic IP address with instance: #{e.message}"
  return "Error"
end

```

- Untuk detail API, lihat [AssociateAddress](#) di Referensi AWS SDK for Ruby API.

SAP ABAP

SDK untuk SAP ABAP

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```

TRY.
    oo_result = lo_ec2->associateaddress(                                " oo_result
is returned for testing purposes. "
    iv_allocationid = iv_allocation_id
    iv_instanceid = iv_instance_id
    ).
    MESSAGE 'Associated an Elastic IP address with an EC2 instance.' TYPE
'I'.
    CATCH /aws1/cx_rt_service_generic INTO DATA(lo_exception).
    DATA(lv_error) = |"{ lo_exception->av_err_code }" - { lo_exception-
>av_err_msg }|.
    MESSAGE lv_error TYPE 'E'.
ENDTRY.

```

- Untuk detail API, lihat [AssociateAddress](#) di AWS SDK untuk referensi SAP ABAP API.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan `AssociateDhcpOptions` dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `AssociateDhcpOptions`.

CLI

AWS CLI

Untuk mengaitkan opsi DHCP yang disetel dengan VPC Anda

Contoh ini mengaitkan opsi DHCP tertentu yang ditetapkan dengan VPC yang ditentukan. Jika perintah berhasil, tidak ada output yang akan ditampilkan.

Perintah:

```
aws ec2 associate-dhcp-options --dhcp-options-id dopt-d9070ebb --vpc-id vpc-a01106c2
```

Untuk mengaitkan opsi DHCP default yang disetel dengan VPC Anda

Contoh ini mengaitkan opsi DHCP default yang ditetapkan dengan VPC yang ditentukan. Jika perintah berhasil, tidak ada output yang akan ditampilkan.

Perintah:

```
aws ec2 associate-dhcp-options --dhcp-options-id default --vpc-id vpc-a01106c2
```

- Untuk detail API, lihat [AssociateDhcpOptions](#) di Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini mengaitkan opsi DHCP tertentu yang ditetapkan dengan VPC yang ditentukan.

```
Register-EC2DhcpOption -DhcpOptionsId dopt-1a2b3c4d -VpcId vpc-12345678
```

Contoh 2: Contoh ini mengaitkan opsi DHCP default yang ditetapkan dengan VPC yang ditentukan.

```
Register-EC2DhcpOption -DhcpOptionsId default -VpcId vpc-12345678
```

- Untuk detail API, lihat [AssociateDhcpOptions](#) di Referensi AWS Tools for PowerShell Cmdlet.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **AssociateRouteTable** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `AssociateRouteTable`.

CLI

AWS CLI

Untuk mengaitkan tabel rute dengan subnet

Contoh ini mengaitkan tabel rute yang ditentukan dengan subnet yang ditentukan.

Perintah:

```
aws ec2 associate-route-table --route-table-id rtb-22574640 --subnet-id subnet-9d4a7b6c
```

Output:

```
{
  "AssociationId": "rtbassoc-781d0d1a"
}
```

- Untuk detail API, lihat [AssociateRouteTable](#) di Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini mengaitkan tabel rute yang ditentukan dengan subnet yang ditentukan.

```
Register-EC2RouteTable -RouteTableId rtb-1a2b3c4d -SubnetId subnet-1a2b3c4d
```

Output:

```
rtbassoc-12345678
```

- Untuk detail API, lihat [AssociateRouteTable](#) di Referensi AWS Tools for PowerShell Cmdlet.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **AttachInternetGateway** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `AttachInternetGateway`.

CLI

AWS CLI

Untuk melampirkan gateway internet ke VPC Anda

`attach-internet-gateway` Contoh berikut melampirkan gateway internet yang ditentukan ke VPC tertentu.

```
aws ec2 attach-internet-gateway \  
  --internet-gateway-id igw-0d0fb496b3EXAMPLE \  
  --vpc-id vpc-0a60eb65b4EXAMPLE
```

Perintah ini tidak menghasilkan output.

Untuk informasi lebih lanjut, lihat [Gateway internet](#) di Panduan Pengguna Amazon VPC.

- Untuk detail API, lihat [AttachInternetGateway](#) di Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini melampirkan gateway Internet yang ditentukan ke VPC yang ditentukan.

```
Add-EC2InternetGateway -InternetGatewayId igw-1a2b3c4d -VpcId vpc-12345678
```

Contoh 2: Contoh ini membuat VPC dan gateway Internet, dan kemudian melampirkan gateway Internet ke VPC.

```
$vpc = New-EC2Vpc -CidrBlock 10.0.0.0/16  
New-EC2InternetGateway | Add-EC2InternetGateway -VpcId $vpc.VpcId
```

- Untuk detail API, lihat [AttachInternetGateway](#) di Referensi AWS Tools for PowerShell Cmdlet.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **AttachNetworkInterface** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `AttachNetworkInterface`.

CLI

AWS CLI

Contoh 1: Untuk melampirkan antarmuka jaringan ke sebuah instance

`attach-network-interface` Contoh berikut melampirkan antarmuka jaringan yang ditentukan untuk contoh yang ditentukan.

```
aws ec2 attach-network-interface \  
  --network-interface-id eni-0dc56a8d4640ad10a \  
  --instance-id i-1234567890abcdef0 \  
  --device-index 1
```

Output:

```
{  
  "AttachmentId": "eni-attach-01a8fc87363f07cf9"  
}
```

Untuk informasi selengkapnya, lihat [Antarmuka jaringan elastis](#) di Panduan Pengguna Amazon EC2.

Contoh 2: Untuk melampirkan antarmuka jaringan ke instance dengan beberapa kartu jaringan `attach-network-interface` Contoh berikut melampirkan antarmuka jaringan yang ditentukan untuk contoh yang ditentukan dan kartu jaringan.

```
aws ec2 attach-network-interface \  
  --network-interface-id eni-07483b1897541ad83 \  
  --instance-id i-01234567890abcdef \  
  --network-card-index 1 \  
  --device-index 1
```

Output:

```
{  
  "AttachmentId": "eni-attach-0fbd7ee87a88cd06c"  
}
```

Untuk informasi selengkapnya, lihat [Antarmuka jaringan elastis](#) di Panduan Pengguna Amazon EC2.

- Untuk detail API, lihat [AttachNetworkInterface](#) di Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini melampirkan antarmuka jaringan yang ditentukan ke instance tertentu.

```
Add-EC2NetworkInterface -NetworkInterfaceId eni-12345678 -InstanceId i-1a2b3c4d -  
DeviceIndex 1
```

Output:

```
eni-attach-1a2b3c4d
```

- Untuk detail API, lihat [AttachNetworkInterface](#) di Referensi AWS Tools for PowerShell Cmdlet.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **AttachVolume** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `AttachVolume`.

CLI

AWS CLI

Untuk melampirkan volume ke sebuah instance

Perintah contoh ini melampirkan volume (`vol-1234567890abcdef0`) ke instance (`i-01474ef662b89480`) sebagai `/dev/sdf`.

Perintah:

```
aws ec2 attach-volume --volume-id vol-1234567890abcdef0 --instance-id i-01474ef662b89480 --device /dev/sdf
```

Output:

```
{
  "AttachTime": "YYYY-MM-DDTHH:MM:SS.000Z",
  "InstanceId": "i-01474ef662b89480",
  "VolumeId": "vol-1234567890abcdef0",
  "State": "attaching",
  "Device": "/dev/sdf"
}
```

- Untuk detail API, lihat [AttachVolume](#) di Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini melampirkan volume yang ditentukan ke instance yang ditentukan dan memaparkannya dengan nama perangkat yang ditentukan.

```
Add-EC2Volume -VolumeId vol-12345678 -InstanceId i-1a2b3c4d -Device /dev/sdh
```

Output:

```
AttachTime      : 12/22/2015 1:53:58 AM
DeleteOnTermination : False
Device          : /dev/sdh
InstanceId      : i-1a2b3c4d
State          : attaching
VolumeId       : vol-12345678
```

- Untuk detail API, lihat [AttachVolume](#) di Referensi AWS Tools for PowerShell Cmdlet.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **AttachVpnGateway** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `AttachVpnGateway`.

CLI

AWS CLI

Untuk melampirkan gateway pribadi virtual ke VPC Anda

`attach-vpn-gateway` Contoh berikut melampirkan gateway pribadi virtual yang ditentukan ke VPC yang ditentukan.

```
aws ec2 attach-vpn-gateway \
  --vpn-gateway-id vgw-9a4cacf3 \
  --vpc-id vpc-a01106c2
```

Output:

```
{
  "VpcAttachment": {
    "State": "attaching",
```

```
    "VpcId": "vpc-a01106c2"  
  }  
}
```

- Untuk detail API, lihat [AttachVpnGateway](#) di Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini melampirkan gateway pribadi virtual yang ditentukan ke VPC yang ditentukan.

```
Add-EC2VpnGateway -VpnGatewayId vgw-1a2b3c4d -VpcId vpc-12345678
```

Output:

```
State      VpcId  
-----  
attaching  vpc-12345678
```

- Untuk detail API, lihat [AttachVpnGateway](#) di Referensi AWS Tools for PowerShell Cmdlet.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **AuthorizeSecurityGroupEgress** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `AuthorizeSecurityGroupEgress`.

CLI

AWS CLI

Untuk menambahkan aturan yang memungkinkan lalu lintas keluar ke rentang alamat tertentu

Perintah contoh ini menambahkan aturan yang memberikan akses ke rentang alamat yang ditentukan pada port TCP 80.

Perintah (Linux):

```
aws ec2 authorize-security-group-egress --group-id sg-1a2b3c4d --ip-permissions IpProtocol=tcp,FromPort=80,ToPort=80,IpRanges='[{"CidrIp=10.0.0.0/16}]'
```

Perintah (Windows):

```
aws ec2 authorize-security-group-egress --group-id sg-1a2b3c4d --ip-permissions IpProtocol=tcp,FromPort=80,ToPort=80,IpRanges=[{"CidrIp=10.0.0.0/16}]
```

Untuk menambahkan aturan yang memungkinkan lalu lintas keluar ke grup keamanan tertentu

Perintah contoh ini menambahkan aturan yang memberikan akses ke grup keamanan yang ditentukan pada port TCP 80.

Perintah (Linux):

```
aws ec2 authorize-security-group-egress --group-id sg-1a2b3c4d --ip-permissions IpProtocol=tcp,FromPort=80,ToPort=80,UserIdGroupPairs='[{"GroupId=sg-4b51a32f}]'
```

Perintah (Windows):

```
aws ec2 authorize-security-group-egress --group-id sg-1a2b3c4d --ip-permissions IpProtocol=tcp,FromPort=80,ToPort=80,UserIdGroupPairs=[{"GroupId=sg-4b51a32f}]
```

- Untuk detail API, lihat [AuthorizeSecurityGroupEgress](#) di Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini mendefinisikan aturan jalan keluar untuk grup keamanan yang ditentukan untuk EC2-VPC. Aturan memberikan akses ke rentang alamat IP yang ditentukan pada port TCP 80. Sintaks yang digunakan oleh contoh ini membutuhkan PowerShell versi 3 atau lebih tinggi.

```
$ip = @{ IpProtocol="tcp"; FromPort="80"; ToPort="80"; IpRanges="203.0.113.0/24" }
```

```
Grant-EC2SecurityGroupEgress -GroupId sg-12345678 -IpPermission $ip
```

Contoh 2: Dengan PowerShell versi 2, Anda harus menggunakan New-Object untuk membuat objek. IpPermission

```
$ip = New-Object Amazon.EC2.Model.IpPermission
$ip.IpProtocol = "tcp"
$ip.FromPort = 80
$ip.ToPort = 80
$ip.IpRanges.Add("203.0.113.0/24")

Grant-EC2SecurityGroupEgress -GroupId sg-12345678 -IpPermission $ip
```

Contoh 3: Contoh ini memberikan akses ke grup keamanan sumber yang ditentukan pada port TCP 80.

```
$ug = New-Object Amazon.EC2.Model.UserIdGroupPair
$ug.GroupId = "sg-1a2b3c4d"
$ug.UserId = "123456789012"

Grant-EC2SecurityGroupEgress -GroupId sg-12345678 -IpPermission
@( @{ IpProtocol="tcp"; FromPort="80"; ToPort="80"; UserIdGroupPairs=$ug } )
```

- Untuk detail API, lihat [AuthorizeSecurityGroupEgress](#) di Referensi AWS Tools for PowerShell Cmdlet.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **AuthorizeSecurityGroupIngress** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `AuthorizeSecurityGroupIngress`.

Contoh tindakan adalah kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Anda dapat melihat tindakan ini dalam konteks dalam contoh kode berikut:

- [Memulai instans](#)

.NET

AWS SDK for .NET

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
/// <summary>
/// Authorize the local computer ingress to EC2 instances associated
/// with the virtual private cloud (VPC) security group.
/// </summary>
/// <param name="groupName">The name of the security group.</param>
/// <returns>A Boolean value indicating the success of the action.</returns>
public async Task<bool> AuthorizeSecurityGroupIngress(string groupName)
{
    // Get the IP address for the local computer.
    var ipAddress = await GetIpAddress();
    Console.WriteLine($"Your IP address is: {ipAddress}");
    var ipRanges = new List<IpRange> { new IpRange { CidrIp =
    $"{ipAddress}/32" } };
    var permission = new IpPermission
    {
        Ipv4Ranges = ipRanges,
        IpProtocol = "tcp",
        FromPort = 22,
        ToPort = 22
    };
    var permissions = new List<IpPermission> { permission };
    var response = await _amazonEC2.AuthorizeSecurityGroupIngressAsync(
        new AuthorizeSecurityGroupIngressRequest(groupName, permissions));
    return response.HttpStatusCode == HttpStatusCode.OK;
}

/// <summary>
/// Authorize the local computer for ingress to
/// the Amazon EC2 SecurityGroup.
/// </summary>
/// <returns>The IPv4 address of the computer running the scenario.</returns>
private static async Task<string> GetIpAddress()
```

```

{
    var httpClient = new HttpClient();
    var ipString = await httpClient.GetStringAsync("https://
checkip.amazonaws.com");

    // The IP address is returned with a new line
    // character on the end. Trim off the whitespace and
    // return the value to the caller.
    return ipString.Trim();
}

```

- Untuk detail API, lihat [AuthorizeSecurityGroupIngress](#) di Referensi AWS SDK for .NET API.

Bash

AWS CLI dengan skrip Bash

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```

#####
# function ec2_authorize_security_group_ingress
#
# This function authorizes an ingress rule for an Amazon Elastic Compute Cloud
# (Amazon EC2) security group.
#
# Parameters:
#     -g security_group_id - The ID of the security group.
#     -i ip_address - The IP address or CIDR block to authorize.
#     -p protocol - The protocol to authorize (e.g., tcp, udp, icmp).
#     -f from_port - The start of the port range to authorize.
#     -t to_port - The end of the port range to authorize.
#
# And:
#     0 - If successful.
#     1 - If it fails.
#####
function ec2_authorize_security_group_ingress() {

```

```
local security_group_id ip_address protocol from_port to_port response
local option OPTARG # Required to use getopt command in a function.

# bashsupport disable=BP5008
function usage() {
    echo "function ec2_authorize_security_group_ingress"
    echo "Authorizes an ingress rule for an Amazon Elastic Compute Cloud (Amazon
EC2) security group."
    echo "  -g security_group_id - The ID of the security group."
    echo "  -i ip_address - The IP address or CIDR block to authorize."
    echo "  -p protocol - The protocol to authorize (e.g., tcp, udp, icmp)."
    echo "  -f from_port - The start of the port range to authorize."
    echo "  -t to_port - The end of the port range to authorize."
    echo ""
}

# Retrieve the calling parameters.
while getopt "g:i:p:f:t:h" option; do
    case "${option}" in
        g) security_group_id="${OPTARG}" ;;
        i) ip_address="${OPTARG}" ;;
        p) protocol="${OPTARG}" ;;
        f) from_port="${OPTARG}" ;;
        t) to_port="${OPTARG}" ;;
        h)
            usage
            return 0
            ;;
        \?)
            echo "Invalid parameter"
            usage
            return 1
            ;;
    esac
done
export OPTIND=1

if [[ -z "$security_group_id" ]]; then
    errecho "ERROR: You must provide a security group ID with the -g parameter."
    usage
    return 1
fi

if [[ -z "$ip_address" ]]; then
```



```

    errecho "ERROR: You must provide an IP address or CIDR block with the -i
parameter."
    usage
    return 1
fi

if [[ -z "$protocol" ]]; then
    errecho "ERROR: You must provide a protocol with the -p parameter."
    usage
    return 1
fi

if [[ -z "$from_port" ]]; then
    errecho "ERROR: You must provide a start port with the -f parameter."
    usage
    return 1
fi

if [[ -z "$to_port" ]]; then
    errecho "ERROR: You must provide an end port with the -t parameter."
    usage
    return 1
fi

response=$(aws ec2 authorize-security-group-ingress \
    --group-id "$security_group_id" \
    --cidr "${ip_address}/32" \
    --protocol "$protocol" \
    --port "$from_port-$to_port" \
    --output text) || {
    aws_cli_error_log ${?}
    errecho "ERROR: AWS reports authorize-security-group-ingress operation
failed.$response"
    return 1
}

return 0
}

```

Fungsi utilitas yang digunakan dalam contoh ini.

```
#####
```

```

# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
#####
function errecho() {
    printf "%s\n" "$*" 1>&2
}

#####
# function aws_cli_error_log()
#
# This function is used to log the error messages from the AWS CLI.
#
# The function expects the following argument:
#     $1 - The error code returned by the AWS CLI.
#
# Returns:
#     0: - Success.
#
#####
function aws_cli_error_log() {
    local err_code=$1
    errecho "Error code : $err_code"
    if [ "$err_code" == 1 ]; then
        errecho " One or more S3 transfers failed."
    elif [ "$err_code" == 2 ]; then
        errecho " Command line failed to parse."
    elif [ "$err_code" == 130 ]; then
        errecho " Process received SIGINT."
    elif [ "$err_code" == 252 ]; then
        errecho " Command syntax invalid."
    elif [ "$err_code" == 253 ]; then
        errecho " The system environment or configuration was invalid."
    elif [ "$err_code" == 254 ]; then
        errecho " The service returned an error."
    elif [ "$err_code" == 255 ]; then
        errecho " 255 is a catch-all error."
    fi


    return 0
}

```

- Untuk detail API, lihat [AuthorizeSecurityGroupIngress](#) di Referensi AWS CLI Perintah.

C++

SDK untuk C++

 Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
Aws::EC2::EC2Client ec2Client(clientConfiguration);

Aws::EC2::Model::IpRange ip_range;
ip_range.SetCidrIp("0.0.0.0/0");

Aws::EC2::Model::IpPermission permission1;
permission1.SetIpProtocol("tcp");
permission1.SetToPort(80);
permission1.SetFromPort(80);
permission1.AddIpRanges(ip_range);

authorize_request.AddIpPermissions(permission1);

Aws::EC2::Model::IpPermission permission2;
permission2.SetIpProtocol("tcp");
permission2.SetToPort(22);
permission2.SetFromPort(22);
permission2.AddIpRanges(ip_range);

authorize_request.AddIpPermissions(permission2);

const Aws::EC2::Model::AuthorizeSecurityGroupIngressOutcome authorizeOutcome
=
    ec2Client.AuthorizeSecurityGroupIngress(authorizeRequest);

if (!authorizeOutcome.IsSuccess()) {
    std::cerr << "Failed to set ingress policy for security group " <<
        groupName << ":" << authorizeOutcome.GetError().GetMessage() <<
        std::endl;
    return false;
}
```

```
std::cout << "Successfully added ingress policy to security group " <<
    groupName << std::endl;
```

- Untuk detail API, lihat [AuthorizeSecurityGroupIngress](#) di Referensi AWS SDK for C++ API.

CLI

AWS CLI

Contoh 1: Untuk menambahkan aturan yang mengizinkan lalu lintas SSH masuk

Contoh `authorize-security-group-ingress` berikut menambahkan aturan yang mengizinkan lalu lintas masuk pada TCP port 22 (SSH).

```
aws ec2 authorize-security-group-ingress \
  --group-id sg-1234567890abcdef0 \
  --protocol tcp \
  --port 22 \
  --cidr 203.0.113.0/24
```

Output:

```
{
  "Return": true,
  "SecurityGroupRules": [
    {
      "SecurityGroupRuleId": "sgr-01afa97ef3e1bedfc",
      "GroupId": "sg-1234567890abcdef0",
      "GroupOwnerId": "123456789012",
      "IsEgress": false,
      "IpProtocol": "tcp",
      "FromPort": 22,
      "ToPort": 22,
      "CidrIpv4": "203.0.113.0/24"
    }
  ]
}
```

Contoh 2: Untuk menambahkan aturan yang mengizinkan lalu lintas HTTP masuk dari grup keamanan lain

Contoh `authorize-security-group-ingress` berikut menambahkan aturan yang mengizinkan akses masuk pada TCP port 80 dari grup keamanan sumber `sg-1a2b3c4d`. Grup sumber harus berada di VPC atau VPC peer yang sama (memerlukan koneksi peering VPC). Lalu lintas masuk diizinkan berdasarkan alamat IP privat dari instans yang dikaitkan dengan grup keamanan sumber (bukan alamat IP publik atau alamat IP Elastis).

```
aws ec2 authorize-security-group-ingress \
  --group-id sg-1234567890abcdef0 \
  --protocol tcp \
  --port 80 \
  --source-group sg-1a2b3c4d
```

Output:

```
{
  "Return": true,
  "SecurityGroupRules": [
    {
      "SecurityGroupRuleId": "sgr-01f4be99110f638a7",
      "GroupId": "sg-1234567890abcdef0",
      "GroupOwnerId": "123456789012",
      "IsEgress": false,
      "IpProtocol": "tcp",
      "FromPort": 80,
      "ToPort": 80,
      "ReferencedGroupInfo": {
        "GroupId": "sg-1a2b3c4d",
        "UserId": "123456789012"
      }
    }
  ]
}
```

Contoh 3: Untuk menambahkan banyak aturan dalam panggilan yang sama

Contoh `authorize-security-group-ingress` berikut menggunakan parameter `ip-permissions` untuk menambahkan dua aturan masuk, satu yang mengizinkan akses masuk pada TCP port 3389 (RDP) dan satu yang mengizinkan ping/ICMP.

```
aws ec2 authorize-security-group-ingress --group-id sg-1234567890abcdef0 --ip-permissions
IpProtocol =tcp, =3389, FromPort =3389, = "[{=172.31.0.0/16}]" =icmp, =-1, =-1, =
"[{=172.31.0.0/16}]ToPort]" IpRanges CidrIp IpProtocol FromPort ToPort IpRanges CidrIp
```

Output:

```
{
  "Return": true,
  "SecurityGroupRules": [
    {
      "SecurityGroupRuleId": "sgr-00e06e5d3690f29f3",
      "GroupId": "sg-1234567890abcdef0",
      "GroupOwnerId": "123456789012",
      "IsEgress": false,
      "IpProtocol": "tcp",
      "FromPort": 3389,
      "ToPort": 3389,
      "CidrIpv4": "172.31.0.0/16"
    },
    {
      "SecurityGroupRuleId": "sgr-0a133dd4493944b87",
      "GroupId": "sg-1234567890abcdef0",
      "GroupOwnerId": "123456789012",
      "IsEgress": false,
      "IpProtocol": "tcp",
      "FromPort": -1,
      "ToPort": -1,
      "CidrIpv4": "172.31.0.0/16"
    }
  ]
}
```

Contoh 4: Guna menambahkan aturan untuk lalu lintas ICMP

Contoh `authorize-security-group-ingress` berikut menggunakan parameter `ip-permissions` untuk menambahkan aturan masuk yang mengizinkan pesan ICMP Destination Unreachable: Fragmentation Needed and Don't Fragment was Set (Tipe 3, Kode 4) dari mana saja.

```
aws ec2 authorize-security-group-ingress --group-id sg-1234567890abcdef0 --ip-permissions
=icmp, =3, =4, = "[{=0.0.0.0/0}]" IpProtocol FromPort ToPort IpRanges CidrIp
```

Output:

```
{
  "Return": true,
  "SecurityGroupRules": [
    {
      "SecurityGroupRuleId": "sgr-0de3811019069b787",
      "GroupId": "sg-1234567890abcdef0",
      "GroupOwnerId": "123456789012",
      "IsEgress": false,
      "IpProtocol": "icmp",
      "FromPort": 3,
      "ToPort": 4,
      "CidrIpv4": "0.0.0.0/0"
    }
  ]
}
```

Contoh 5: Guna menambahkan aturan untuk lalu lintas IPv6

Contoh `authorize-security-group-ingress` berikut menggunakan parameter `ip-permissions` untuk menambahkan aturan masuk yang mengizinkan akses SSH (port 22) dari rentang IPv6 `2001:db8:1234:1a00::/64`.

```
aws ec2 authorize-security-group-ingress --group-id sg-1234567890abcdef0 --ip-permissions
=tc, =22, =22, Ipv6Ranges= "[{6=2001:db 8:1234:1 a00: IpProtocol :/64}]" FromPort ToPort
CidrIpv
```

Output:

```
{
  "Return": true,
  "SecurityGroupRules": [
    {
      "SecurityGroupRuleId": "sgr-0455bc68b60805563",
      "GroupId": "sg-1234567890abcdef0",
      "GroupOwnerId": "123456789012",
      "IsEgress": false,
      "IpProtocol": "tcp",
      "FromPort": 22,
      "ToPort": 22,
      "CidrIpv6": "2001:db8:1234:1a00::/64"
    }
  ]
}
```

```
}

```

Contoh 6: Guna menambahkan aturan untuk lalu lintas ICMPv6

Contoh `authorize-security-group-ingress` berikut menggunakan parameter `ip-permissions` untuk menambahkan aturan masuk yang mengizinkan lalu lintas ICMPv6 dari mana saja.

```
aws ec2 authorize-security-group-ingress --group-id sg-1234567890abcdef0 --ip-permissions
=icmpv6, Ipv6Ranges= "[{6=: :/0}]" IpProtocol CidrIpv
```

Output:

```
{
  "Return": true,
  "SecurityGroupRules": [
    {
      "SecurityGroupRuleId": "sgr-04b612d9363ab6327",
      "GroupId": "sg-1234567890abcdef0",
      "GroupOwnerId": "123456789012",
      "IsEgress": false,
      "IpProtocol": "icmpv6",
      "FromPort": -1,
      "ToPort": -1,
      "CidrIpv6": "::/0"
    }
  ]
}
```

Contoh 7: Tambahkan aturan dengan deskripsi

Contoh `authorize-security-group-ingress` berikut menggunakan parameter `ip-permissions` untuk menambahkan aturan masuk yang mengizinkan lalu lintas RDP dari rentang alamat IPv4 tertentu. Aturan mencakup deskripsi untuk membantu Anda mengidentifikasinya nanti.

```
aws ec2 authorize-security-group-ingress --group-id sg-1234567890abcdef0 --ip-permissions
IpProtocol =tcp, =3389, =3389, = "[{=203.0.113.0/24, description='akses RDP dari kantor NY'}]"
FromPort ToPort IpRanges CidrIp
```

Output:


```
{
  "Return": true,
  "SecurityGroupRules": [
    {
      "SecurityGroupRuleId": "sgr-0397bbcc01e974db3",
      "GroupId": "sg-1234567890abcdef0",
      "GroupOwnerId": "123456789012",
      "IsEgress": false,
      "IpProtocol": "tcp",
      "FromPort": 3389,
      "ToPort": 3389,
      "CidrIpv4": "203.0.113.0/24",
      "Description": "RDP access from NY office"
    }
  ]
}
```

Contoh 8: Untuk menambahkan aturan masuk yang menggunakan daftar prefixs

Contoh `authorize-security-group-ingress` berikut menggunakan parameter `ip-permissions` untuk menambahkan aturan masuk yang mengizinkan semua lalu lintas untuk rentang CIDR dalam daftar prefixs tertentu.

```
aws ec2 authorize-security-group-ingress --group-id sg-04a351bfe432d4e71 --ip-permissions
=all, = [{"=pl-002dc3ec097de1514}]" IpProtocol PrefixListIds PrefixListId
```

Output:

```
{
  "Return": true,
  "SecurityGroupRules": [
    {
      "SecurityGroupRuleId": "sgr-09c74b32f677c6c7c",
      "GroupId": "sg-1234567890abcdef0",
      "GroupOwnerId": "123456789012",
      "IsEgress": false,
      "IpProtocol": "-1",
      "FromPort": -1,
      "ToPort": -1,
      "PrefixListId": "pl-0721453c7ac4ec009"
    }
  ]
}
```

```
}
```

Untuk informasi selengkapnya, lihat [Grup keamanan](#) di Panduan Pengguna Amazon VPC.

- Untuk detail API, lihat [AuthorizeSecurityGroupIngress](#) di Referensi AWS CLI Perintah.

Java

SDK untuk Java 2.x

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
public static String createSecurityGroup(Ec2Client ec2, String groupName,
String groupDesc, String vpcId,
    String myIpAddress) {
    try {
        CreateSecurityGroupRequest createRequest =
CreateSecurityGroupRequest.builder()
            .groupName(groupName)
            .description(groupDesc)
            .vpcId(vpcId)
            .build();

        CreateSecurityGroupResponse resp =
ec2.createSecurityGroup(createRequest);
        IpRange ipRange = IpRange.builder()
            .cidrIp(myIpAddress + "/0")
            .build();

        IpPermission ipPerm = IpPermission.builder()
            .ipProtocol("tcp")
            .toPort(80)
            .fromPort(80)
            .ipRanges(ipRange)
            .build();

        IpPermission ipPerm2 = IpPermission.builder()
            .ipProtocol("tcp")
```

```
        .toPort(22)
        .fromPort(22)
        .ipRanges(ipRange)
        .build();

        AuthorizeSecurityGroupIngressRequest authRequest =
AuthorizeSecurityGroupIngressRequest.builder()
        .groupName(groupName)
        .ipPermissions(ipPerm, ipPerm2)
        .build();

        ec2.authorizeSecurityGroupIngress(authRequest);
        System.out.println("Successfully added ingress policy to security
group " + groupName);
        return resp.groupId();

    } catch (Ec2Exception e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
    return "";
}
```

- Untuk detail API, lihat [AuthorizeSecurityGroupIngress](#) di Referensi AWS SDK for Java 2.x API.

JavaScript

SDK untuk JavaScript (v3)

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
import { AuthorizeSecurityGroupIngressCommand } from "@aws-sdk/client-ec2";

import { client } from "../libs/client.js";
```

```
// Grant permissions for a single IP address to ssh into instances
// within the provided security group.
export const main = async () => {
  const command = new AuthorizeSecurityGroupIngressCommand({
    // Replace with a security group ID from the AWS console or
    // the DescribeSecurityGroupsCommand.
    GroupId: "SECURITY_GROUP_ID",
    IpPermissions: [
      {
        IpProtocol: "tcp",
        FromPort: 22,
        ToPort: 22,
        // Replace 0.0.0.0 with the IP address to authorize.
        // For more information on this notation, see
        // https://en.wikipedia.org/wiki/Classless_Inter-
Domain_Routing#CIDR_notation
        IpRanges: [{ CidrIp: "0.0.0.0/32" }],
      },
    ],
  });

  try {
    const { SecurityGroupRules } = await client.send(command);
    console.log(JSON.stringify(SecurityGroupRules, null, 2));
  } catch (err) {
    console.error(err);
  }
};
```

- Untuk detail API, lihat [AuthorizeSecurityGroupIngress](#) di Referensi AWS SDK for JavaScript API.

Kotlin

SDK untuk Kotlin

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
suspend fun createEC2SecurityGroupSc(
    groupNameVal: String?,
    groupDescVal: String?,
    vpcIdVal: String?,
    myIpAddress: String?,
): String? {
    val request =
        CreateSecurityGroupRequest {
            groupName = groupNameVal
            description = groupDescVal
            vpcId = vpcIdVal
        }

    Ec2Client { region = "us-west-2" }.use { ec2 ->
        val resp = ec2.createSecurityGroup(request)
        val ipRange =
            IpRange {
                cidrIp = "$myIpAddress/0"
            }

        val ipPerm =
            IpPermission {
                ipProtocol = "tcp"
                toPort = 80
                fromPort = 80
                ipRanges = listOf(ipRange)
            }

        val ipPerm2 =
            IpPermission {
                ipProtocol = "tcp"
                toPort = 22
                fromPort = 22
                ipRanges = listOf(ipRange)
            }

        val authRequest =
            AuthorizeSecurityGroupIngressRequest {
                groupName = groupNameVal
                ipPermissions = listOf(ipPerm, ipPerm2)
            }
        ec2.authorizeSecurityGroupIngress(authRequest)
    }
}
```

```

        println("Successfully added ingress policy to Security Group
$groupNameVal")
        return resp.groupId
    }
}

```

- Untuk detail API, lihat [AuthorizeSecurityGroupIngress](#) di AWS SDK untuk referensi API Kotlin.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini mendefinisikan aturan masuk untuk grup keamanan untuk EC2-VPC. Aturan ini memberikan akses ke alamat IP tertentu untuk SSH (port 22) dan RDC (port 3389). Perhatikan bahwa Anda harus mengidentifikasi grup keamanan untuk EC2-VPC menggunakan ID grup keamanan bukan nama grup keamanan. Sintaks yang digunakan oleh contoh ini membutuhkan PowerShell versi 3 atau lebih tinggi.

```

$ip1 = @{ IpProtocol="tcp"; FromPort="22"; ToPort="22";
  IpRanges="203.0.113.25/32" }
$ip2 = @{ IpProtocol="tcp"; FromPort="3389"; ToPort="3389";
  IpRanges="203.0.113.25/32" }

Grant-EC2SecurityGroupIngress -GroupId sg-12345678 -IpPermission @( $ip1, $ip2 )

```

Contoh 2: Dengan PowerShell versi 2, Anda harus menggunakan New-Object untuk membuat objek. IpPermission

```

$ip1 = New-Object Amazon.EC2.Model.IpPermission
$ip1.IpProtocol = "tcp"
$ip1.FromPort = 22
$ip1.ToPort = 22
$ip1.IpRanges.Add("203.0.113.25/32")

$ip2 = new-object Amazon.EC2.Model.IpPermission
$ip2.IpProtocol = "tcp"
$ip2.FromPort = 3389
$ip2.ToPort = 3389
$ip2.IpRanges.Add("203.0.113.25/32")

```

```
Grant-EC2SecurityGroupIngress -GroupId sg-12345678 -IpPermission @( $ip1, $ip2 )
```

Contoh 3: Contoh ini mendefinisikan aturan ingress untuk grup keamanan untuk EC2-Classic. Aturan ini memberikan akses ke alamat IP tertentu untuk SSH (port 22) dan RDC (port 3389). Sintaks yang digunakan oleh contoh ini membutuhkan PowerShell versi 3 atau lebih tinggi.

```
$ip1 = @{ IpProtocol="tcp"; FromPort="22"; ToPort="22";  
  IpRanges="203.0.113.25/32" }  
$ip2 = @{ IpProtocol="tcp"; FromPort="3389"; ToPort="3389";  
  IpRanges="203.0.113.25/32" }  
  
Grant-EC2SecurityGroupIngress -GroupName "my-security-group" -IpPermission  
  @( $ip1, $ip2 )
```

Contoh 4: Dengan PowerShell versi 2, Anda harus menggunakan New-Object untuk membuat objek. IpPermission

```
$ip1 = New-Object Amazon.EC2.Model.IpPermission  
$ip1.IpProtocol = "tcp"  
$ip1.FromPort = 22  
$ip1.ToPort = 22  
$ip1.IpRanges.Add("203.0.113.25/32")  
  
$ip2 = new-object Amazon.EC2.Model.IpPermission  
$ip2.IpProtocol = "tcp"  
$ip2.FromPort = 3389  
$ip2.ToPort = 3389  
$ip2.IpRanges.Add("203.0.113.25/32")  
  
Grant-EC2SecurityGroupIngress -GroupName "my-security-group" -IpPermission  
  @( $ip1, $ip2 )
```

Contoh 5: Contoh ini memberikan akses port TCP 8081 dari grup keamanan sumber tertentu (sg-1a2b3c4d) ke grup keamanan yang ditentukan (sg-12345678).

```
$ug = New-Object Amazon.EC2.Model.UserIdGroupPair  
$ug.GroupId = "sg-1a2b3c4d"  
$ug.UserId = "123456789012"
```

```
Grant-EC2SecurityGroupIngress -GroupId sg-12345678 -IpPermission
@( @{ IpProtocol="tcp"; FromPort="8081"; ToPort="8081"; UserIdGroupPairs=$ug } )
```

Contoh 6: Contoh ini menambahkan CIDR 5.5.5.5/32 ke aturan Ingress dari Grup keamanan sg-1234abcd untuk lalu lintas port TCP 22 dengan deskripsi.

```
$IpRange = New-Object -TypeName Amazon.EC2.Model.IpRange
$IpRange.CidrIp = "5.5.5.5/32"
$IpRange.Description = "SSH from Office"
$IpPermission = New-Object Amazon.EC2.Model.IpPermission
$IpPermission.IpProtocol = "tcp"
$IpPermission.ToPort = 22
$IpPermission.FromPort = 22
$IpPermission.Ipv4Ranges = $IpRange
Grant-EC2SecurityGroupIngress -GroupId sg-1234abcd -IpPermission $IpPermission
```

- Untuk detail API, lihat [AuthorizeSecurityGroupIngress](#) di Referensi AWS Tools for PowerShell Cmdlet.

Python

SDK untuk Python (Boto3)

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
class SecurityGroupWrapper:
    """Encapsulates Amazon Elastic Compute Cloud (Amazon EC2) security group
    actions."""

    def __init__(self, ec2_resource, security_group=None):
        """
        :param ec2_resource: A Boto3 Amazon EC2 resource. This high-level
        resource
                                is used to create additional high-level objects
                                that wrap low-level Amazon EC2 service actions.
        :param security_group: A Boto3 SecurityGroup object. This is a high-level
        object
```



```
        that wraps security group actions.

    """
    self.ec2_resource = ec2_resource
    self.security_group = security_group

    @classmethod
    def from_resource(cls):
        ec2_resource = boto3.resource("ec2")
        return cls(ec2_resource)

    def authorize_ingress(self, ssh_ingress_ip):
        """
        Adds a rule to the security group to allow access to SSH.

        :param ssh_ingress_ip: The IP address that is granted inbound access to
connect
                                to port 22 over TCP, used for SSH.
        :return: The response to the authorization request. The 'Return' field of
the
                response indicates whether the request succeeded or failed.
        """
        if self.security_group is None:
            logger.info("No security group to update.")
            return

        try:
            ip_permissions = [
                {
                    # SSH ingress open to only the specified IP address.
                    "IpProtocol": "tcp",
                    "FromPort": 22,
                    "ToPort": 22,
                    "IpRanges": [{"CidrIp": f"{ssh_ingress_ip}/32"}],
                }
            ]
            response = self.security_group.authorize_ingress(
                IpPermissions=ip_permissions
            )
        except ClientError as err:
            logger.error(
                "Couldn't authorize inbound rules for %s. Here's why: %s: %s",
                self.security_group.id,
                err.response["Error"]["Code"],
```

```
        err.response["Error"]["Message"],
    )
    raise
else:
    return response
```

- Untuk detail API, lihat [AuthorizeSecurityGroupIngress](#) di AWS SDK for Python (Boto3) Referensi API.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **CancelCapacityReservation** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `CancelCapacityReservation`.

CLI

AWS CLI

Untuk membatalkan reservasi kapasitas

`cancel-capacity-reservation` Contoh berikut membatalkan reservasi kapasitas yang ditentukan.

```
aws ec2 cancel-capacity-reservation \
    --capacity-reservation-id cr-1234abcd56EXAMPLE
```

Output:

```
{
  "Return": true
}
```

Untuk informasi selengkapnya, lihat [Membatalkan Reservasi Kapasitas](#) di Panduan Pengguna Amazon Elastic Compute Cloud untuk Instans Linux.

- Untuk detail API, lihat [CancelCapacityReservation](#) di Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini membatalkan reservasi kapasitas cr-0c1f2345db6f7cdba

```
Remove-EC2CapacityReservation -CapacityReservationId cr-0c1f2345db6f7cdba
```

Output:

```
Confirm
Are you sure you want to perform this action?
Performing the operation "Remove-EC2CapacityReservation
(CancelCapacityReservation)" on target "cr-0c1f2345db6f7cdba".
[Y] Yes [A] Yes to All [N] No [L] No to All [S] Suspend [?] Help (default is
"Y"): y
True
```

- Untuk detail API, lihat [CancelCapacityReservation](#) di Referensi AWS Tools for PowerShell Cmdlet.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **CancelImportTask** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `CancelImportTask`.

CLI

AWS CLI

Untuk membatalkan tugas impor

`cancel-import-task` Contoh berikut membatalkan tugas gambar impor yang ditentukan.

```
aws ec2 cancel-import-task \
```

```
--import-task-id import-ami-1234567890abcdef0
```

Output:

```
{
  "ImportTaskId": "import-ami-1234567890abcdef0",
  "PreviousState": "active",
  "State": "deleting"
}
```

- Untuk detail API, lihat [CancelImportTask](#) di Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini membatalkan tugas impor yang ditentukan (baik snapshot atau impor gambar). Jika diperlukan, alasan dapat menyediakan menggunakan **-CancelReason** parameter.

```
Stop-EC2ImportTask -ImportTaskId import-ami-abcdefgh
```

- Untuk detail API, lihat [CancelImportTask](#) di Referensi AWS Tools for PowerShell Cmdlet.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **CancelSpotFleetRequests** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `CancelSpotFleetRequests`.

CLI

AWS CLI

Contoh 1: Untuk membatalkan permintaan armada Spot dan menghentikan instans terkait

`cancel-spot-fleet-requests` Contoh berikut membatalkan permintaan Armada Spot dan mengakhiri Instans Sesuai Permintaan dan Instans Spot terkait.

```
aws ec2 cancel-spot-fleet-requests \  
  --spot-fleet-request-ids sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE \  
  --terminate-instances
```

Output:

```
{  
  "SuccessfulFleetRequests": [  
    {  
      "SpotFleetRequestId": "sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE",  
      "CurrentSpotFleetRequestState": "cancelled_terminating",  
      "PreviousSpotFleetRequestState": "active"  
    }  
  ],  
  "UnsuccessfulFleetRequests": []  
}
```

Untuk informasi selengkapnya, lihat [Membatalkan permintaan Armada Spot](#) di Panduan Pengguna Amazon Elastic Compute Cloud untuk Instans Linux.

Contoh 2: Untuk membatalkan permintaan armada Spot tanpa menghentikan instance terkait `cancel-spot-fleet-requests` Contoh berikut membatalkan permintaan Armada Spot tanpa menghentikan Instans Sesuai Permintaan dan Instans Spot terkait.

```
aws ec2 cancel-spot-fleet-requests \  
  --spot-fleet-request-ids sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE \  
  --no-terminate-instances
```

Output:

```
{  
  "SuccessfulFleetRequests": [  
    {  
      "SpotFleetRequestId": "sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE",  
      "CurrentSpotFleetRequestState": "cancelled_running",  
      "PreviousSpotFleetRequestState": "active"  
    }  
  ],  
  "UnsuccessfulFleetRequests": []  
}
```

Untuk informasi selengkapnya, lihat [Membatalkan permintaan Armada Spot](#) di Panduan Pengguna Amazon Elastic Compute Cloud untuk Instans Linux.

- Untuk detail API, lihat [CancelSpotFleetRequests](#) di Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini membatalkan permintaan armada Spot yang ditentukan dan mengakhiri instance Spot terkait.

```
Stop-EC2SpotFleetRequest -SpotFleetRequestId sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE -TerminateInstance $true
```

Contoh 2: Contoh ini membatalkan permintaan armada Spot yang ditentukan tanpa menghentikan instans Spot terkait.

```
Stop-EC2SpotFleetRequest -SpotFleetRequestId sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE -TerminateInstance $false
```

- Untuk detail API, lihat [CancelSpotFleetRequests](#) di Referensi AWS Tools for PowerShell Cmdlet.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **CancelSpotInstanceRequests** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `CancelSpotInstanceRequests`.

CLI

AWS CLI

Untuk membatalkan permintaan Instans Spot

Perintah contoh ini membatalkan permintaan Instans Spot.

Perintah:

```
aws ec2 cancel-spot-instance-requests --spot-instance-request-ids sir-08b93456
```

Output:

```
{
  "CancelledSpotInstanceRequests": [
    {
      "State": "cancelled",
      "SpotInstanceRequestId": "sir-08b93456"
    }
  ]
}
```

- Untuk detail API, lihat [CancelSpotInstanceRequests](#) di Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini membatalkan permintaan instance Spot yang ditentukan.

```
Stop-EC2SpotInstanceRequest -SpotInstanceRequestId sir-12345678
```

Output:

SpotInstanceRequestId	State
-----	-----
sir-12345678	cancelled

- Untuk detail API, lihat [CancelSpotInstanceRequests](#) di Referensi AWS Tools for PowerShell Cmdlet.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **ConfirmProductInstance** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `ConfirmProductInstance`.

CLI

AWS CLI

Untuk mengonfirmasi contoh produk

Contoh ini menentukan apakah kode produk yang ditentukan dikaitkan dengan contoh yang ditentukan.

Perintah:

```
aws ec2 confirm-product-instance --product-code 774F4FF8 --instance-id  
i-1234567890abcdef0
```

Output:

```
{  
  "OwnerId": "123456789012"  
}
```

- Untuk detail API, lihat [ConfirmProductInstance](#) di Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini menentukan apakah kode produk yang ditentukan dikaitkan dengan instance yang ditentukan.

```
Confirm-EC2ProductInstance -ProductCode 774F4FF8 -InstanceId i-12345678
```

- Untuk detail API, lihat [ConfirmProductInstance](#) di Referensi AWS Tools for PowerShell Cmdlet.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **CopyImage** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan CopyImage.

CLI

AWS CLI

Contoh 1: Untuk menyalin AMI ke Wilayah lain

copy-image Contoh perintah berikut menyalin AMI yang ditentukan dari us-west-2 Region ke us-east-1 Region dan menambahkan deskripsi singkat.

```
aws ec2 copy-image \  
  --region us-east-1 \  
  --name ami-name \  
  --source-region us-west-2 \  
  --source-image-id ami-066877671789bd71b \  
  --description "This is my copied image."
```

Output:

```
{  
  "ImageId": "ami-0123456789abcdefg"  
}
```

Untuk informasi selengkapnya, lihat [Menyalin AMI](#) di Panduan Pengguna Amazon EC2.

Contoh 2: Untuk menyalin AMI ke Wilayah lain dan mengenkripsi snapshot dukungan

copy-image Perintah berikut menyalin AMI yang ditentukan dari us-west-2 Wilayah ke Wilayah saat ini dan mengenkripsi snapshot dukungan menggunakan kunci KMS yang ditentukan.

```
aws ec2 copy-image \  
  --source-region us-west-2 \  
  --name ami-name \  
  --source-image-id ami-066877671789bd71b \  
  --encrypted \  
  --kms-key-id alias/my-kms-key
```

Output:

```
{
  "ImageId": "ami-0123456789abcdefg"
}
```

Untuk informasi selengkapnya, lihat [Menyalin AMI](#) di Panduan Pengguna Amazon EC2.

Contoh 3: Untuk menyertakan tag AMI yang ditentukan pengguna saat menyalin AMI

copy-image Perintah berikut menggunakan `--copy-image-tags` parameter untuk menyalin tag AMI yang ditentukan pengguna saat menyalin AMI.

```
aws ec2 copy-image \
  --region us-east-1 \
  --name ami-name \
  --source-region us-west-2 \
  --source-image-id ami-066877671789bd71b \
  --description "This is my copied image."
  --copy-image-tags
```

Output:

```
{
  "ImageId": "ami-0123456789abcdefg"
}
```

Untuk informasi selengkapnya, lihat [Menyalin AMI](#) di Panduan Pengguna Amazon EC2.

- Untuk detail API, lihat [CopyImage](#) di Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini menyalin AMI yang ditentukan di wilayah 'UE (Irlandia)' ke wilayah 'AS Barat (Oregon)'. Jika `-Region` tidak ditentukan, wilayah default saat ini digunakan sebagai wilayah tujuan.

```
Copy-EC2Image -SourceRegion eu-west-1 -SourceImageId ami-12345678 -Region us-west-2 -Name "Copy of ami-12345678"
```

Output:

```
ami-87654321
```

- Untuk detail API, lihat [CopyImage](#) di Referensi AWS Tools for PowerShell Cmdlet.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **CopySnapshot** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan CopySnapshot.

CLI

AWS CLI

Contoh 1: Untuk menyalin snapshot ke Wilayah lain

copy-snapshot Contoh perintah berikut menyalin snapshot yang ditentukan dari us-west-2 Region ke us-east-1 Region dan menambahkan deskripsi singkat.

```
aws ec2 copy-snapshot \  
  --region us-east-1 \  
  --source-region us-west-2 \  
  --source-snapshot-id snap-066877671789bd71b \  
  --description "This is my copied snapshot."
```

Output:

```
{  
  "SnapshotId": "snap-066877671789bd71b"  
}
```

Untuk informasi selengkapnya, lihat [Menyalin snapshot Amazon EBS](#) di Panduan Pengguna Amazon EC2.

Contoh 2: Untuk menyalin snapshot yang tidak terenkripsi dan mengenkripsi snapshot baru

copy-snapshot Perintah berikut menyalin snapshot tak terenkripsi yang ditentukan dari us-west-2 Wilayah ke Wilayah saat ini dan mengenkripsi snapshot baru menggunakan kunci KMS yang ditentukan.

```
aws ec2 copy-snapshot \  
  --source-region us-west-2 \  
  --source-snapshot-id snap-066877671789bd71b \  
  --encrypted \  
  --kms-key-id alias/my-kms-key
```

Output:

```
{  
  "SnapshotId": "snap-066877671789bd71b"  
}
```

Untuk informasi selengkapnya, lihat [Menyalin snapshot Amazon EBS](#) di Panduan Pengguna Amazon EC2.

- Untuk detail API, lihat [CopySnapshot](#) di Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini menyalin snapshot yang ditentukan dari wilayah UE (Irlandia) ke wilayah AS Barat (Oregon).

```
Copy-EC2Snapshot -SourceRegion eu-west-1 -SourceSnapshotId snap-12345678 -Region  
us-west-2
```

Contoh 2: Jika Anda menetapkan wilayah default dan menghilangkan parameter Region, wilayah tujuan default adalah wilayah default.

```
Set-DefaultAWSRegion us-west-2  
Copy-EC2Snapshot -SourceRegion eu-west-1 -SourceSnapshotId snap-12345678
```

- Untuk detail API, lihat [CopySnapshot](#) di Referensi AWS Tools for PowerShell Cmdlet.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan `CreateCapacityReservation` dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `CreateCapacityReservation`.

CLI

AWS CLI

Contoh 1: Untuk membuat Reservasi Kapasitas

`create-capacity-reservation` Contoh berikut membuat reservasi kapasitas di `eu-west-1a` Availability Zone, di mana Anda dapat meluncurkan tiga `t2.medium` instance yang menjalankan sistem operasi Linux/Unix. Secara default, reservasi kapasitas dibuat dengan kriteria pencocokan instans terbuka dan tidak ada dukungan untuk penyimpanan sementara, dan tetap aktif hingga Anda membatalkannya secara manual.

```
aws ec2 create-capacity-reservation \
  --availability-zone eu-west-1a \
  --instance-type t2.medium \
  --instance-platform Linux/UNIX \
  --instance-count 3
```

Output:

```
{
  "CapacityReservation": {
    "CapacityReservationId": "cr-1234abcd56EXAMPLE ",
    "EndDateType": "unlimited",
    "AvailabilityZone": "eu-west-1a",
    "InstanceMatchCriteria": "open",
    "EphemeralStorage": false,
    "CreateDate": "2019-08-16T09:27:35.000Z",
    "AvailableInstanceCount": 3,
    "InstancePlatform": "Linux/UNIX",
    "TotalInstanceCount": 3,
    "State": "active",
    "Tenancy": "default",
    "EbsOptimized": false,
    "InstanceType": "t2.medium"
  }
}
```

Contoh 2: Untuk membuat Reservasi Kapasitas yang secara otomatis berakhir pada tanggal/waktu tertentu

`create-capacity-reservation` Contoh berikut membuat reservasi kapasitas di `eu-west-1a` Availability Zone, di mana Anda dapat meluncurkan tiga `m5.large` instance yang menjalankan sistem operasi Linux/Unix. Reservasi kapasitas ini secara otomatis berakhir pada 31/08/2019 pukul 23:59:59.

```
aws ec2 create-capacity-reservation \  
  --availability-zone eu-west-1a \  
  --instance-type m5.large \  
  --instance-platform Linux/UNIX \  
  --instance-count 3 \  
  --end-date-type limited \  
  --end-date 2019-08-31T23:59:59Z
```

Output:

```
{  
  "CapacityReservation": {  
    "CapacityReservationId": "cr-1234abcd56EXAMPLE ",  
    "EndDateType": "limited",  
    "AvailabilityZone": "eu-west-1a",  
    "EndDate": "2019-08-31T23:59:59.000Z",  
    "InstanceMatchCriteria": "open",  
    "EphemeralStorage": false,  
    "CreateDate": "2019-08-16T10:15:53.000Z",  
    "AvailableInstanceCount": 3,  
    "InstancePlatform": "Linux/UNIX",  
    "TotalInstanceCount": 3,  
    "State": "active",  
    "Tenancy": "default",  
    "EbsOptimized": false,  
    "InstanceType": "m5.large"  
  }  
}
```

Contoh 3: Untuk membuat Reservasi Kapasitas yang hanya menerima peluncuran instans yang ditargetkan

`create-capacity-reservation` Contoh berikut membuat reservasi kapasitas yang hanya menerima peluncuran instans yang ditargetkan.

```
aws ec2 create-capacity-reservation \  
  --availability-zone eu-west-1a \  
  --instance-type m5.large \  
  --instance-platform Linux/UNIX \  
  --instance-count 3 \  
  --instance-match-criteria targeted
```

Output:

```
{  
  "CapacityReservation": {  
    "CapacityReservationId": "cr-1234abcd56EXAMPLE ",  
    "EndDateType": "unlimited",  
    "AvailabilityZone": "eu-west-1a",  
    "InstanceMatchCriteria": "targeted",  
    "EphemeralStorage": false,  
    "CreateDate": "2019-08-16T10:21:57.000Z",  
    "AvailableInstanceCount": 3,  
    "InstancePlatform": "Linux/UNIX",  
    "TotalInstanceCount": 3,  
    "State": "active",  
    "Tenancy": "default",  
    "EbsOptimized": false,  
    "InstanceType": "m5.large"  
  }  
}
```

Untuk informasi selengkapnya, lihat [Membuat Reservasi Kapasitas](#) di Panduan Pengguna Amazon Elastic Compute Cloud untuk Instans Linux.

- Untuk detail API, lihat [CreateCapacityReservation](#) di Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini membuat Reservasi Kapasitas baru dengan atribut yang ditentukan

```
Add-EC2CapacityReservation -InstanceType m4.xlarge -InstanceCount 2 -  
AvailabilityZone eu-west-1b -EbsOptimized True -InstancePlatform Windows
```

Output:

```
AvailabilityZone      : eu-west-1b
AvailableInstanceCount : 2
CapacityReservationId : cr-0c1f2345db6f7cdba
CreateDate           : 3/28/2019 9:29:41 AM
EbsOptimized         : True
EndDate              : 1/1/0001 12:00:00 AM
EndDateType          : unlimited
EphemeralStorage     : False
InstanceMatchCriteria : open
InstancePlatform     : Windows
InstanceType         : m4.xlarge
State                : active
Tags                 : {}
Tenancy              : default
TotalInstanceCount   : 2
```

- Untuk detail API, lihat [CreateCapacityReservation](#) di Referensi AWS Tools for PowerShell Cmdlet.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan `CreateCustomerGateway` dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `CreateCustomerGateway`.

CLI

AWS CLI

Untuk membuat gateway pelanggan

Contoh ini membuat gateway pelanggan dengan alamat IP yang ditentukan untuk antarmuka luarnya.

Perintah:

```
aws ec2 create-customer-gateway --type ipsec.1 --public-ip 12.1.2.3 --bgp-asn 65534
```


Output:

```
{
  "CustomerGateway": {
    "CustomerGatewayId": "cgw-0e11f167",
    "IpAddress": "12.1.2.3",
    "State": "available",
    "Type": "ipsec.1",
    "BgpAsn": "65534"
  }
}
```

- Untuk detail API, lihat [CreateCustomerGateway](#) di Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini membuat gateway pelanggan yang ditentukan.

```
New-EC2CustomerGateway -Type ipsec.1 -PublicIp 203.0.113.12 -BgpAsn 65534
```

Output:

```
BgpAsn          : 65534
CustomerGatewayId : cgw-1a2b3c4d
IpAddress       : 203.0.113.12
State          : available
Tags           : {}
Type           : ipsec.1
```

- Untuk detail API, lihat [CreateCustomerGateway](#) di Referensi AWS Tools for PowerShell Cmdlet.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan `CreateDhcpOptions` dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `CreateDhcpOptions`.

CLI

AWS CLI

Untuk membuat satu set opsi DHCP

`create-dhcp-options` Contoh berikut membuat satu set opsi DHCP yang menentukan nama domain, server nama domain, dan jenis node NetBIOS.

```
aws ec2 create-dhcp-options \  
  --dhcp-configuration \  
    "Key=domain-name-servers,Values=10.2.5.1,10.2.5.2" \  
    "Key=domain-name,Values=example.com" \  
    "Key=netbios-node-type,Values=2"
```

Output:

```
{  
  "DhcpOptions": {  
    "DhcpConfigurations": [  
      {  
        "Key": "domain-name",  
        "Values": [  
          {  
            "Value": "example.com"  
          }  
        ]  
      },  
      {  
        "Key": "domain-name-servers",  
        "Values": [  
          {  
            "Value": "10.2.5.1"  
          },  
          {  
            "Value": "10.2.5.2"  
          }  
        ]  
      }  
    ],  
  },  
}
```

```

        {
            "Key": "netbios-node-type",
            "Values": [
                {
                    "Value": "2"
                }
            ]
        },
        "DhcpOptionsId": "dopt-06d52773eff4c55f3"
    }
}

```

- Untuk detail API, lihat [CreateDhcpOptions](#) di Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini menciptakan kumpulan opsi DHCP yang ditentukan. Sintaks yang digunakan oleh contoh ini memerlukan PowerShell versi 3 atau yang lebih baru.

```

$options = @( @{{Key="domain-name";Values=@("abc.local")}}, @{{Key="domain-name-servers";Values=@("10.0.0.101","10.0.0.102")}})
New-EC2DhcpOption -DhcpConfiguration $options

```

Output:

DhcpConfigurations	DhcpOptionsId	Tags
{domain-name, domain-name-servers}	dopt-1a2b3c4d	{}

Contoh 2: Dengan PowerShell versi 2, Anda harus menggunakan New-Object untuk membuat setiap opsi DHCP.

```

$option1 = New-Object Amazon.EC2.Model.DhcpConfiguration
$option1.Key = "domain-name"
$option1.Values = "abc.local"

$option2 = New-Object Amazon.EC2.Model.DhcpConfiguration
$option2.Key = "domain-name-servers"

```

```
$option2.Values = @"10.0.0.101", "10.0.0.102")

New-EC2DhcpOption -DhcpConfiguration @($option1, $option2)
```

Output:

```
DhcpConfigurations          DhcpOptionsId      Tags
-----
{domain-name, domain-name-servers} dopt-2a3b4c5d     {}
```

- Untuk detail API, lihat [CreateDhcpOptions](#) di Referensi AWS Tools for PowerShell Cmdlet.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **CreateFlowLogs** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `CreateFlowLogs`.

CLI

AWS CLI

Contoh 1: Untuk membuat log aliran

`create-flow-logs` Contoh berikut membuat log aliran yang menangkap semua lalu lintas ditolak untuk antarmuka jaringan yang ditentukan. Log aliran dikirim ke grup log di CloudWatch Log menggunakan izin dalam peran IAM yang ditentukan.

```
aws ec2 create-flow-logs \
  --resource-type NetworkInterface \
  --resource-ids eni-11223344556677889 \
  --traffic-type REJECT \
  --log-group-name my-flow-logs \
  --deliver-logs-permission-arn arn:aws:iam::123456789101:role/publishFlowLogs
```

Output:

```
{
  "ClientToken": "so0eNA2uSHUN1HI0S2cJ305GuIX1CezaRdGtexample",
```

```

    "FlowLogIds": [
      "fl-12345678901234567"
    ],
    "Unsuccessful": []
  }

```

Untuk informasi selengkapnya, lihat [Log Alur VPC](#) di Panduan Pengguna Amazon VPC.

Contoh 2: Untuk membuat log alur dengan format kustom

`create-flow-logs` Contoh berikut membuat log alur yang menangkap semua lalu lintas untuk VPC yang ditentukan dan mengirimkan log aliran ke bucket Amazon S3. Parameter `--log-format` menentukan format kustom untuk catatan log alur. Untuk menjalankan perintah ini di Windows, ubah tanda kutip tunggal (') menjadi tanda kutip ganda (").

```

aws ec2 create-flow-logs \
  --resource-type VPC \
  --resource-ids vpc-00112233344556677 \
  --traffic-type ALL \
  --log-destination-type s3 \
  --log-destination arn:aws:s3:::flow-log-bucket/my-custom-flow-logs/ \
  --log-format '${version} ${vpc-id} ${subnet-id} ${instance-id} ${srcaddr}
${dstaddr} ${srcport} ${dstport} ${protocol} ${tcp-flags} ${type} ${pkt-srcaddr}
${pkt-dstaddr}'

```

Untuk informasi selengkapnya, lihat [Log Alur VPC](#) di Panduan Pengguna Amazon VPC.

Contoh 3: Untuk membuat log aliran dengan interval agregasi maksimum satu menit

`create-flow-logs` Contoh berikut membuat log alur yang menangkap semua lalu lintas untuk VPC yang ditentukan dan mengirimkan log aliran ke bucket Amazon S3. `--max-aggregation-interval` Parameter menentukan interval agregasi maksimum 60 detik (1 menit).

```

aws ec2 create-flow-logs \
  --resource-type VPC \
  --resource-ids vpc-00112233344556677 \
  --traffic-type ALL \
  --log-destination-type s3 \
  --log-destination arn:aws:s3:::flow-log-bucket/my-custom-flow-logs/ \
  --max-aggregation-interval 60

```

Untuk informasi selengkapnya, lihat [Log Alur VPC](#) di Panduan Pengguna Amazon VPC.

- Untuk detail API, lihat [CreateFlowLogs](#) di Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini membuat flowlog EC2 untuk subnet subnet-1d234567 ke 'subnet1-log' cloud-watch-log bernama untuk semua lalu lintas 'TOLAK' menggunakan batas peran 'Admin'

```
New-EC2FlowLog -ResourceId "subnet-1d234567" -LogDestinationType cloud-watch-logs -LogGroupName subnet1-log -TrafficType "REJECT" -ResourceType Subnet -DeliverLogsPermissionArn "arn:aws:iam::98765432109:role/Admin"
```

Output:

ClientToken	FlowLogIds	Unsuccessful
-----	-----	-----
m1VN2cxP3iB4qo//VUK15EU6cF7gQL0xcqNefvjeTGw=	{f1-012fc34eed5678c9d}	{}

- Untuk detail API, lihat [CreateFlowLogs](#) di Referensi AWS Tools for PowerShell Cmdlet.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **CreateImage** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `CreateImage`.

CLI

AWS CLI

Contoh 1: Untuk membuat AMI dari instans yang didukung Amazon EBS

`create-image` Contoh berikut membuat AMI dari instance yang ditentukan.

```
aws ec2 create-image \
```

```
--instance-id i-1234567890abcdef0 \  
--name "My server" \  
--description "An AMI for my server"
```

Output:

```
{  
  "ImageId": "ami-abcdef01234567890"  
}
```

Untuk informasi selengkapnya tentang menentukan pemetaan perangkat blokir untuk AMI Anda, lihat [Menentukan pemetaan perangkat blokir untuk AMI di Panduan Pengguna Amazon EC2](#).

Contoh 2: Untuk membuat AMI dari instans yang didukung Amazon EBS tanpa reboot

`create-image` Contoh berikut membuat AMI dan menetapkan parameter `--no-reboot`, sehingga instance tidak di-reboot sebelum gambar dibuat.

```
aws ec2 create-image \  
  --instance-id i-1234567890abcdef0 \  
  --name "My server" \  
  --no-reboot
```

Output:

```
{  
  "ImageId": "ami-abcdef01234567890"  
}
```

Untuk informasi selengkapnya tentang menentukan pemetaan perangkat blokir untuk AMI Anda, lihat [Menentukan pemetaan perangkat blokir untuk AMI di Panduan Pengguna Amazon EC2](#).

Contoh 3: Untuk menandai AMI dan snapshot pada pembuatan

`create-image` Contoh berikut membuat AMI, dan menandai AMI dan snapshot dengan tag yang sama `cost-center=cc123`

```
aws ec2 create-image \  
  --instance-id i-1234567890abcdef0 \  
  --name "My server" \  
  --no-reboot
```

```
--instance-id i-1234567890abcdef0 \  
--name "My server" \  
--tag-specifications "ResourceType=image,Tags=[{Key=cost-  
center,Value=cc123}]" "ResourceType=snapshot,Tags=[{Key=cost-  
center,Value=cc123}]"
```

Output:

```
{  
  "ImageId": "ami-abcdef01234567890"  
}
```

Untuk informasi selengkapnya tentang menandai sumber daya Anda saat pembuatan, lihat [Menambahkan tag pada pembuatan sumber daya](#) di Panduan Pengguna Amazon EC2.

- Untuk detail API, lihat [CreateImage](#) di Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini membuat AMI dengan nama dan deskripsi yang ditentukan, dari instance yang ditentukan. Amazon EC2 mencoba mematikan instance dengan bersih sebelum membuat gambar, dan memulai ulang instance setelah selesai.

```
New-EC2Image -InstanceId i-12345678 -Name "my-web-server" -Description "My web  
server AMI"
```

Contoh 2: Contoh ini membuat AMI dengan nama dan deskripsi yang ditentukan, dari instance yang ditentukan. Amazon EC2 membuat gambar tanpa mematikan dan memulai ulang instance; oleh karena itu, integritas sistem file pada gambar yang dibuat tidak dapat dijamin.

```
New-EC2Image -InstanceId i-12345678 -Name "my-web-server" -Description "My web  
server AMI" -NoReboot $true
```

Contoh 3: Contoh ini membuat AMI dengan tiga volume. Volume pertama didasarkan pada snapshot Amazon EBS. Volume kedua adalah volume 100 GiB Amazon EBS kosong. Volume ketiga adalah volume penyimpanan instance. Sintaks yang digunakan oleh contoh ini membutuhkan PowerShell versi 3 atau lebih tinggi.


```
$ebsBlock1 = @{SnapshotId="snap-1a2b3c4d"}
$ebsBlock2 = @{VolumeSize=100}

New-EC2Image -InstanceId i-12345678 -Name "my-web-server" -Description
  "My web server AMI" -BlockDeviceMapping @( @{DeviceName="/dev/sdf";Ebs=
  $ebsBlock1}, @{DeviceName="/dev/sdg";Ebs=$ebsBlock2}, @{DeviceName="/dev/
  sdc";VirtualName="ephemeral0"})
```

- Untuk detail API, lihat [CreateImage](#) di Referensi AWS Tools for PowerShell Cmdlet.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **CreateInstanceExportTask** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `CreateInstanceExportTask`.

CLI

AWS CLI

Untuk mengekspor sebuah instance

Perintah contoh ini membuat tugas untuk mengekspor instance `i-1234567890abcdef0` ke bucket Amazon S3 `myexportbucket`.

Perintah:

```
aws ec2 create-instance-export-task --description "RHEL5 instance" --instance-
id i-1234567890abcdef0 --target-environment vmware --export-to-s3-task
  DiskImageFormat=vmdk,ContainerFormat=ova,S3Bucket=myexportbucket,S3Prefix=RHEL5
```

Output:

```
{
  "ExportTask": {
    "State": "active",
    "InstanceExportDetails": {
      "InstanceId": "i-1234567890abcdef0",
      "TargetEnvironment": "vmware"
    }
  }
}
```

```

    },
    "ExportToS3Task": {
      "S3Bucket": "myexportbucket",
      "S3Key": "RHEL5export-i-fh8sjjsq.ova",
      "DiskImageFormat": "vmdk",
      "ContainerFormat": "ova"
    },
    "Description": "RHEL5 instance",
    "ExportTaskId": "export-i-fh8sjjsq"
  }
}

```

- Untuk detail API, lihat [CreateInstanceExportTask](#) di Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini mengekspor instance yang dihentikan, **i-0800b00a00EXAMPLE**, sebagai hard disk virtual (VHD) ke bucket S3. **testbucket-export-instances-2019** Lingkungan target adalah **Microsoft**, dan parameter wilayah ditambahkan karena instance ada di **us-east-1** wilayah, sedangkan AWS Wilayah default pengguna bukan us-east-1. Untuk mendapatkan status tugas ekspor, salin **ExportTaskId** nilai dari hasil perintah ini, lalu jalankan **Get-EC2ExportTask -ExportTaskId export_task_ID_from_results**.

```

New-EC2InstanceExportTask -InstanceId i-0800b00a00EXAMPLE -
ExportToS3Task_DiskImageFormat VHD -ExportToS3Task_S3Bucket "testbucket-export-
instances-2019" -TargetEnvironment Microsoft -Region us-east-1

```

Output:

```

Description           :
ExportTaskId          : export-i-077c73108aEXAMPLE
ExportToS3Task        : Amazon.EC2.Model.ExportToS3Task
InstanceExportDetails : Amazon.EC2.Model.InstanceExportDetails
State                 : active
StatusMessage         :

```

- Untuk detail API, lihat [CreateInstanceExportTask](#) di Referensi AWS Tools for PowerShell Cmdlet.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan `CreateInternetGateway` dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `CreateInternetGateway`.

CLI

AWS CLI

Untuk membuat gateway internet

`create-internet-gateway` Contoh berikut membuat gateway internet dengan `tagName=my-igw`.

```
aws ec2 create-internet-gateway \
  --tag-specifications ResourceType=internet-gateway,Tags=[{Key=Name,Value=my-igw}]
```

Output:

```
{
  "InternetGateway": {
    "Attachments": [],
    "InternetGatewayId": "igw-0d0fb496b3994d755",
    "OwnerId": "123456789012",
    "Tags": [
      {
        "Key": "Name",
        "Value": "my-igw"
      }
    ]
  }
}
```

Untuk informasi lebih lanjut, lihat [Gateway internet](#) di Panduan Pengguna Amazon VPC.

- Untuk detail API, lihat [CreateInternetGateway](#) di Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini membuat gateway Internet.

```
New-EC2InternetGateway
```

Output:

Attachments	InternetGatewayId	Tags
-----	-----	----
{}	igw-1a2b3c4d	{}

- Untuk detail API, lihat [CreateInternetGateway](#) di Referensi AWS Tools for PowerShell Cmdlet.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **CreateKeyPair** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `CreateKeyPair`.

Contoh tindakan adalah kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Anda dapat melihat tindakan ini dalam konteks dalam contoh kode berikut:

- [Memulai instans](#)

.NET

AWS SDK for .NET

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
/// <summary>
/// Create an Amazon EC2 key pair.
/// </summary>
/// <param name="keyPairName">The name for the new key pair.</param>
/// <returns>The Amazon EC2 key pair created.</returns>
public async Task<KeyPair?> CreateKeyPair(string keyPairName)
{
    var request = new CreateKeyPairRequest
    {
        KeyName = keyPairName,
    };

    var response = await _amazonEC2.CreateKeyPairAsync(request);

    if (response.HttpStatusCode == HttpStatusCode.OK)
    {
        var kp = response.KeyPair;
        return kp;
    }
    else
    {
        Console.WriteLine("Could not create key pair.");
        return null;
    }
}

/// <summary>
/// Save KeyPair information to a temporary file.
/// </summary>
/// <param name="keyPair">The name of the key pair.</param>
/// <returns>The full path to the temporary file.</returns>
public string SaveKeyPair(KeyPair keyPair)
{
    var tempPath = Path.GetTempPath();
    var tempFileName = $"{tempPath}\\{Path.GetRandomFileName()}";
    var pemFileName = Path.ChangeExtension(tempFileName, "pem");

    // Save the key pair to a file in a temporary folder.
    using var stream = new FileStream(pemFileName, FileMode.Create);
    using var writer = new StreamWriter(stream);
    writer.WriteLine(keyPair.KeyMaterial);

    return pemFileName;
}
```

```
}

```

- Untuk detail API, lihat [CreateKeyPair](#) di Referensi AWS SDK for .NET API.

Bash

AWS CLI dengan skrip Bash

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
#####
# function ec2_create_keypair
#
# This function creates an Amazon Elastic Compute Cloud (Amazon EC2) ED25519 or
# 2048-bit RSA key pair
# and writes it to a file.
#
# Parameters:
#     -n key_pair_name - A key pair name.
#     -f file_path - File to store the key pair.
#
# And:
#     0 - If successful.
#     1 - If it fails.
#####
function ec2_create_keypair() {
    local key_pair_name file_path response
    local option OPTARG # Required to use getopt command in a function.

    # bashsupport disable=BP5008
    function usage() {
        echo "function ec2_create_keypair"
        echo "Creates an Amazon Elastic Compute Cloud (Amazon EC2) ED25519 or 2048-
bit RSA key pair"
        echo " and writes it to a file."
        echo "  -n key_pair_name - A key pair name."
        echo "  -f file_path - File to store the key pair."
    }
}

```

```
    echo ""
}

# Retrieve the calling parameters.
while getopts "n:f:h" option; do
    case "${option}" in
        n) key_pair_name="${OPTARG}" ;;
        f) file_path="${OPTARG}" ;;
        h)
            usage
            return 0
            ;;
        \?)
            echo "Invalid parameter"
            usage
            return 1
            ;;
    esac
done
export OPTIND=1

if [[ -z "$key_pair_name" ]]; then
    errecho "ERROR: You must provide a key name with the -n parameter."
    usage
    return 1
fi

if [[ -z "$file_path" ]]; then
    errecho "ERROR: You must provide a file path with the -f parameter."
    usage
    return 1
fi

response=$(aws ec2 create-key-pair \
    --key-name "$key_pair_name" \
    --query 'KeyMaterial' \
    --output text) || {
    aws_cli_error_log ${?}
    errecho "ERROR: AWS reports create-access-key operation failed.$response"
    return 1
}

if [[ -n "$file_path" ]]; then
    echo "$response" >"$file_path"
```

```

fi

return 0
}

```

Fungsi utilitas yang digunakan dalam contoh ini.

```

#####
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
#####
function errecho() {
    printf "%s\n" "$*" 1>&2
}

#####
# function aws_cli_error_log()
#
# This function is used to log the error messages from the AWS CLI.
#
# The function expects the following argument:
#     $1 - The error code returned by the AWS CLI.
#
# Returns:
#     0: - Success.
#
#####
function aws_cli_error_log() {
    local err_code=$1
    errecho "Error code : $err_code"
    if [ "$err_code" == 1 ]; then
        errecho " One or more S3 transfers failed."
    elif [ "$err_code" == 2 ]; then
        errecho " Command line failed to parse."
    elif [ "$err_code" == 130 ]; then
        errecho " Process received SIGINT."
    elif [ "$err_code" == 252 ]; then
        errecho " Command syntax invalid."
    elif [ "$err_code" == 253 ]; then
        errecho " The system environment or configuration was invalid."
    elif [ "$err_code" == 254 ]; then

```



```
    errecho " The service returned an error."
elif [ "$err_code" == 255 ]; then
    errecho " 255 is a catch-all error."
fi

return 0
}
```

- Untuk detail API, lihat [CreateKeyPair](#) di Referensi AWS CLI Perintah.

C++

SDK untuk C++

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
Aws::EC2::EC2Client ec2Client(clientConfiguration);
Aws::EC2::Model::CreateKeyPairRequest request;
request.SetKeyName(keyPairName);

Aws::EC2::Model::CreateKeyPairOutcome outcome =
ec2Client.CreateKeyPair(request);
if (!outcome.IsSuccess()) {
    std::cerr << "Failed to create key pair:" <<
        outcome.GetError().GetMessage() << std::endl;
}
else {
    std::cout << "Successfully created key pair named " <<
        keyPairName << std::endl;
}
```

- Untuk detail API, lihat [CreateKeyPair](#) di Referensi AWS SDK for C++ API.

CLI

AWS CLI

Untuk membuat pasangan kunci

Contoh ini membuat pasangan kunci bernama `MyKeyPair`.

Perintah:

```
aws ec2 create-key-pair --key-name MyKeyPair
```

Output-nya adalah versi ASCII dari kunci privat dan sidik jari kunci. Anda perlu menyimpan kunci ke file.

Untuk informasi selengkapnya, lihat [Menggunakan Pasangan Kunci di Panduan Pengguna Antarmuka Baris Perintah AWS](#).

- Untuk detail API, lihat [CreateKeyPair](#) di Referensi AWS CLI Perintah.

Java

SDK untuk Java 2.x

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
public static void createKeyPair(Ec2Client ec2, String keyName, String
fileName) {
    try {
        CreateKeyPairRequest request = CreateKeyPairRequest.builder()
            .keyName(keyName)
            .build();

        CreateKeyPairResponse response = ec2.createKeyPair(request);
        String content = response.keyMaterial();
        BufferedWriter writer = new BufferedWriter(new FileWriter(fileName));
        writer.write(content);
    }
}
```

```
        writer.close();
        System.out.println("Successfully created key pair named " + keyName);

    } catch (Ec2Exception | IOException e) {
        System.err.println(e.getMessage());
        System.exit(1);
    }
}
```

- Untuk detail API, lihat [CreateKeyPair](#) di Referensi AWS SDK for Java 2.x API.

JavaScript

SDK untuk JavaScript (v3)

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
import { CreateKeyPairCommand } from "@aws-sdk/client-ec2";

import { client } from "../libs/client.js";

export const main = async () => {
  try {
    // Create a key pair in Amazon EC2.
    const { KeyMaterial, KeyName } = await client.send(
      // A unique name for the key pair. Up to 255 ASCII characters.
      new CreateKeyPairCommand({ KeyName: "KEY_PAIR_NAME" }),
    );
    // This logs your private key. Be sure to save it.
    console.log(KeyName);
    console.log(KeyMaterial);
  } catch (err) {
    console.error(err);
  }
};
```

- Untuk detail API, lihat [CreateKeyPair](#) di Referensi AWS SDK for JavaScript API.

Kotlin

SDK untuk Kotlin

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
suspend fun createEC2KeyPair(keyNameVal: String) {
    val request =
        CreateKeyPairRequest {
            keyName = keyNameVal
        }

    Ec2Client { region = "us-west-2" }.use { ec2 ->
        val response = ec2.createKeyPair(request)
        println("The key ID is ${response.keyPairId}")
    }
}
```

- Untuk detail API, lihat [CreateKeyPair](#) di AWS SDK untuk referensi API Kotlin.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini membuat key pair dan menangkap kunci pribadi RSA yang dikodekan PEM dalam file dengan nama yang ditentukan. Saat Anda menggunakan PowerShell, pengkodean harus diatur ke ascii untuk menghasilkan kunci yang valid. Untuk informasi selengkapnya, lihat Membuat, Menampilkan, dan Menghapus Pasangan Kunci Amazon EC2 (<https://docs.aws.amazon.com/cli/latest/userguide/cli-services-ec2-keypairs.html>) di Panduan Pengguna Antarmuka Baris AWS Perintah.

```
(New-EC2KeyPair -KeyName "my-key-pair").KeyMaterial | Out-File -Encoding ascii -
FilePath C:\path\my-key-pair.pem
```

- Untuk detail API, lihat [CreateKeyPair](#) di Referensi AWS Tools for PowerShell Cmdlet.

Python

SDK untuk Python (Boto3)

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
class KeyPairWrapper:
    """Encapsulates Amazon Elastic Compute Cloud (Amazon EC2) key pair
    actions."""

    def __init__(self, ec2_resource, key_file_dir, key_pair=None):
        """
        :param ec2_resource: A Boto3 Amazon EC2 resource. This high-level
        resource
                                is used to create additional high-level objects
                                that wrap low-level Amazon EC2 service actions.
        :param key_file_dir: The folder where the private key information is
        stored.
                                This should be a secure folder.
        :param key_pair: A Boto3 KeyPair object. This is a high-level object that
        wraps key pair actions.
        """
        self.ec2_resource = ec2_resource
        self.key_pair = key_pair
        self.key_file_path = None
        self.key_file_dir = key_file_dir

    @classmethod
    def from_resource(cls):
        ec2_resource = boto3.resource("ec2")
        return cls(ec2_resource, tempfile.TemporaryDirectory())
```

```
def create(self, key_name):
    """
    Creates a key pair that can be used to securely connect to an EC2
instance.
    The returned key pair contains private key information that cannot be
retrieved
again. The private key data is stored as a .pem file.

:param key_name: The name of the key pair to create.
:return: A Boto3 KeyPair object that represents the newly created key
pair.
    """
    try:
        self.key_pair = self.ec2_resource.create_key_pair(KeyName=key_name)
        self.key_file_path = os.path.join(
            self.key_file_dir.name, f"{self.key_pair.name}.pem"
        )
        with open(self.key_file_path, "w") as key_file:
            key_file.write(self.key_pair.key_material)
    except ClientError as err:
        logger.error(
            "Couldn't create key %s. Here's why: %s: %s",
            key_name,
            err.response["Error"]["Code"],
            err.response["Error"]["Message"],
        )
        raise
    else:
        return self.key_pair
```

- Untuk detail API, lihat [CreateKeyPair](#) di AWS SDK for Python (Boto3) Referensi API.

Ruby

SDK untuk Ruby

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
# This code example does the following:
# 1. Creates a key pair in Amazon Elastic Compute Cloud (Amazon EC2).
# 2. Displays information about available key pairs.
# 3. Deletes the key pair.

require "aws-sdk-ec2"

# @param ec2_client [Aws::EC2::Client] An initialized EC2 client.
# @param key_pair_name [String] The name for the key pair and private
#   key file.
# @return [Boolean] true if the key pair and private key file were
#   created; otherwise, false.
# @example
#   exit 1 unless key_pair_created?(
#     Aws::EC2::Client.new(region: 'us-west-2'),
#     'my-key-pair'
#   )
def key_pair_created?(ec2_client, key_pair_name)
  key_pair = ec2_client.create_key_pair(key_name: key_pair_name)
  puts "Created key pair '#{key_pair.key_name}' with fingerprint " \
    "'#{key_pair.key_fingerprint}' and ID '#{key_pair.key_pair_id}'."
  filename = File.join(Dir.home, key_pair_name + ".pem")
  File.open(filename, "w") { |file| file.write(key_pair.key_material) }
  puts "Private key file saved locally as '#{filename}'."
  return true
rescue Aws::EC2::Errors::InvalidKeyPairDuplicate
  puts "Error creating key pair: a key pair named '#{key_pair_name}' " \
    "already exists."
  return false
rescue StandardError => e
  puts "Error creating key pair or saving private key file: #{e.message}"
```

```
    return false
  end

  # Displays information about available key pairs in
  # Amazon Elastic Compute Cloud (Amazon EC2).
  #
  # @param ec2_client [Aws::EC2::Client] An initialized EC2 client.
  # @example
  #   describe_key_pairs(Aws::EC2::Client.new(region: 'us-west-2'))
  def describe_key_pairs(ec2_client)
    result = ec2_client.describe_key_pairs
    if result.key_pairs.count.zero?
      puts "No key pairs found."
    else
      puts "Key pair names:"
      result.key_pairs.each do |key_pair|
        puts key_pair.key_name
      end
    end
  end
rescue StandardError => e
  puts "Error getting information about key pairs: #{e.message}"
end

# Deletes a key pair in Amazon Elastic Compute Cloud (Amazon EC2).
#
# Prerequisites:
#
# - The key pair to delete.
#
# @param ec2_client [Aws::EC2::Client] An initialized EC2 client.
# @param key_pair_name [String] The name of the key pair to delete.
# @return [Boolean] true if the key pair was deleted; otherwise, false.
# @example
#   exit 1 unless key_pair_deleted?(
#     Aws::EC2::Client.new(region: 'us-west-2'),
#     'my-key-pair'
#   )
def key_pair_deleted?(ec2_client, key_pair_name)
  ec2_client.delete_key_pair(key_name: key_pair_name)
  return true
rescue StandardError => e
  puts "Error deleting key pair: #{e.message}"
  return false
end
```



```
# Example usage:
def run_me
  key_pair_name = ""
  region = ""
  # Print usage information and then stop.
  if ARGV[0] == "--help" || ARGV[0] == "-h"
    puts "Usage: ruby ec2-ruby-example-key-pairs.rb KEY_PAIR_NAME REGION"
    puts "Example: ruby ec2-ruby-example-key-pairs.rb my-key-pair us-west-2"
    exit 1
  # If no values are specified at the command prompt, use these default values.
  # Replace us-west-2 with the AWS Region you're using for Amazon EC2.
  elsif ARGV.count.zero?
    key_pair_name = "my-key-pair"
    region = "us-west-2"
  # Otherwise, use the values as specified at the command prompt.
  else
    key_pair_name = ARGV[0]
    region = ARGV[1]
  end

  ec2_client = Aws::EC2::Client.new(region: region)

  puts "Displaying existing key pair names before creating this key pair..."
  describe_key_pairs(ec2_client)

  puts "-" * 10
  puts "Creating key pair..."
  unless key_pair_created?(ec2_client, key_pair_name)
    puts "Stopping program."
    exit 1
  end

  puts "-" * 10
  puts "Displaying existing key pair names after creating this key pair..."
  describe_key_pairs(ec2_client)

  puts "-" * 10
  puts "Deleting key pair..."
  unless key_pair_deleted?(ec2_client, key_pair_name)
    puts "Stopping program. You must delete the key pair yourself."
    exit 1
  end
  puts "Key pair deleted."
```

```

puts "-" * 10
puts "Now that the key pair is deleted, " \
     "also deleting the related private key pair file..."
filename = File.join(Dir.home, key_pair_name + ".pem")
File.delete(filename)
if File.exist?(filename)
  puts "Could not delete file at '#{filename}'. You must delete it yourself."
else
  puts "File deleted."
end

puts "-" * 10
puts "Displaying existing key pair names after deleting this key pair..."
describe_key_pairs(ec2_client)
end

run_me if $PROGRAM_NAME == __FILE__

```

- Untuk detail API, lihat [CreateKeyPair](#) di Referensi AWS SDK for Ruby API.

SAP ABAP

SDK untuk SAP ABAP

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```

TRY.
  oo_result = lo_ec2->createkeypair( iv_keyname = iv_key_name ).
    " oo_result is returned for testing purposes. "
  MESSAGE 'Amazon EC2 key pair created.' TYPE 'I'.
  CATCH /aws1/cx_rt_service_generic INTO DATA(lo_exception).
  DATA(lv_error) = |"{ lo_exception->av_err_code }" - { lo_exception-
>av_err_msg }|.
  MESSAGE lv_error TYPE 'E'.
ENDTRY.

```

- Untuk detail API, lihat [CreateKeyPair](#) di AWS SDK untuk referensi SAP ABAP API.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **CreateLaunchTemplate** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `CreateLaunchTemplate`.

Contoh tindakan adalah kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Anda dapat melihat tindakan ini dalam konteks dalam contoh kode berikut:

- [Membangun dan mengelola layanan yang tangguh](#)

.NET

AWS SDK for .NET

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
/// <summary>
/// Creates an Amazon EC2 launch template to use with Amazon EC2 Auto
Scaling.
/// The launch template specifies a Bash script in its user data field that
runs after
/// the instance is started. This script installs the Python packages and
starts a Python
/// web server on the instance.
/// </summary>
/// <param name="startupScriptPath">The path to a Bash script file that is
run.</param>
/// <param name="instancePolicyPath">The path to a permissions policy to
create and attach to the profile.</param>
```

```
/// <returns>The template object.</returns>
public async Task<Amazon.EC2.Model.LaunchTemplate> CreateTemplate(string
startupScriptPath, string instancePolicyPath)
{
    await CreateKeyPair(_keyPairName);
    await CreateInstanceProfileWithName(_instancePolicyName,
_instanceRoleName, _instanceProfileName, instancePolicyPath);

    var startServerText = await File.ReadAllTextAsync(startupScriptPath);
    var plainTextBytes = System.Text.Encoding.UTF8.GetBytes(startServerText);

    var amiLatest = await _amazonSsm.GetParameterAsync(
        new GetParameterRequest() { Name = _amiParam });
    var amiId = amiLatest.Parameter.Value;
    var launchTemplateResponse = await _amazonEc2.CreateLaunchTemplateAsync(
        new CreateLaunchTemplateRequest()
        {
            LaunchTemplateName = _launchTemplateName,
            LaunchTemplateData = new RequestLaunchTemplateData()
            {
                InstanceType = _instanceType,
                ImageId = amiId,
                IamInstanceProfile =
                    new
LaunchTemplateIamInstanceProfileSpecificationRequest()
                    {
                        Name = _instanceProfileName
                    },
                KeyName = _keyPairName,
                UserData = System.Convert.ToBase64String(plainTextBytes)
            }
        });
    return launchTemplateResponse.LaunchTemplate;
}
```

- Untuk detail API, lihat [CreateLaunchTemplate](#) di Referensi AWS SDK for .NET API.

CLI

AWS CLI

Contoh 1: Untuk membuat templat peluncuran

Contoh `create-launch-template` berikut membuat templat peluncuran yang menentukan subnet untuk meluncurkan instans, menetapkan alamat IP publik dan alamat IPv6 ke instans, lalu membuat tanda untuk instans.

```
aws ec2 create-launch-template \
  --launch-template-name TemplateForWebServer \
  --version-description WebVersion1 \
  --launch-template-data '{"NetworkInterfaces":
[{"AssociatePublicIpAddress":true,"DeviceIndex":0,"Ipv6AddressCount":1,"SubnetId":"subnet-
[{"ResourceType":"instance","Tags":[{"Key":"purpose","Value":"webserver"}]}]}'
```

Output:

```
{
  "LaunchTemplate": {
    "LatestVersionNumber": 1,
    "LaunchTemplateId": "lt-01238c059e3466abc",
    "LaunchTemplateName": "TemplateForWebServer",
    "DefaultVersionNumber": 1,
    "CreatedBy": "arn:aws:iam::123456789012:user/Bob",
    "CreateTime": "2019-01-27T09:13:24.000Z"
  }
}
```

Untuk informasi selengkapnya, lihat [Meluncurkan Instans dari Templat Peluncuran](#) di [Panduan Pengguna Amazon Elastic Compute Cloud](#). Untuk informasi tentang mengutip parameter berformat JSON, lihat [Mengutip String](#) di [Panduan Pengguna Antarmuka Baris Perintah AWS](#).

Contoh 2: Guna membuat templat peluncuran untuk Amazon EC2 Auto Scaling

Contoh `create-launch-template` berikut membuat templat peluncuran dengan banyak tanda dan pemetaan perangkat blok untuk menentukan volume EBS tambahan saat instans diluncurkan. Tentukan nilai untuk `Groups` yang sesuai dengan grup keamanan untuk VPC yang akan menjadi tujuan peluncuran instans oleh grup Auto Scaling Anda. Tentukan VPC dan subnet sebagai properti grup Auto Scaling.

```
aws ec2 create-launch-template \  
  --launch-template-name TemplateForAutoScaling \  
  --version-description AutoScalingVersion1 \  
  --launch-template-data '{"NetworkInterfaces":  
  [{"DeviceIndex":0,"AssociatePublicIpAddress":true,"Groups":  
  [{"sg-7c227019,sg-903004f8}], "DeleteOnTermination":true}], "ImageId":"ami-  
b42209de", "InstanceType":"m4.large", "TagSpecifications":  
  [{"ResourceType":"instance", "Tags":[{"Key":"environment", "Value":"production"},  
  {"Key":"purpose", "Value":"webserver"}]}, {"ResourceType":"volume", "Tags":  
  [{"Key":"environment", "Value":"production"}, {"Key":"cost-  
center", "Value":"cc123"}]}]', "BlockDeviceMappings":[{"DeviceName":"/dev/  
sda1", "Ebs":{"VolumeSize":100}}]}' --region us-east-1
```

Output:

```
{  
  "LaunchTemplate": {  
    "LatestVersionNumber": 1,  
    "LaunchTemplateId": "lt-0123c79c33a54e0abc",  
    "LaunchTemplateName": "TemplateForAutoScaling",  
    "DefaultVersionNumber": 1,  
    "CreatedBy": "arn:aws:iam::123456789012:user/Bob",  
    "CreateTime": "2019-04-30T18:16:06.000Z"  
  }  
}
```

Untuk informasi selengkapnya, lihat Membuat Templat Peluncuran untuk Grup Auto Scaling di Panduan Pengguna Amazon EC2 Auto Scaling. Untuk informasi tentang mengutip parameter berformat JSON, lihat Mengutip String di Panduan Pengguna Antarmuka Baris Perintah AWS .

Contoh 3: Untuk membuat templat peluncuran yang menentukan enkripsi volume EBS

Contoh `create-launch-template` berikut membuat templat peluncuran yang menyertakan volume EBS terenkripsi yang dibuat dari snapshot yang tidak terenkripsi. Contoh tersebut juga menandai volume selama pembuatan. Jika enkripsi secara default dinonaktifkan, Anda harus menentukan opsi `"Encrypted"` seperti yang ditunjukkan pada contoh berikut. Jika menggunakan opsi `"KmsKeyId"` untuk menentukan CMK yang dikelola pelanggan, Anda juga harus menentukan opsi `"Encrypted"` bahkan jika enkripsi secara default diaktifkan.

```
aws ec2 create-launch-template \  
  --launch-template-name TemplateForEncryption \  
  --launch-template-data '{"NetworkInterfaces":
```

```
--launch-template-data file://config.json
```

Isi dari config.json:

```
{
  "BlockDeviceMappings":[
    {
      "DeviceName":"/dev/sda1",
      "Ebs":{"
        "VolumeType":"gp2",
        "DeleteOnTermination":true,
        "SnapshotId":"snap-066877671789bd71b",
        "Encrypted":true,
        "KmsKeyId":"arn:aws:kms:us-east-1:012345678910:key/abcd1234-
a123-456a-a12b-a123b4cd56ef"
      }
    }
  ],
  "ImageId":"ami-00068cd7555f543d5",
  "InstanceType":"c5.large",
  "TagSpecifications":[
    {
      "ResourceType":"volume",
      "Tags":[
        {
          "Key":"encrypted",
          "Value":"yes"
        }
      ]
    }
  ]
}
```

Output:

```
{
  "LaunchTemplate": {
    "LatestVersionNumber": 1,
    "LaunchTemplateId": "lt-0d5bd51bcf8530abc",
    "LaunchTemplateName": "TemplateForEncryption",
    "DefaultVersionNumber": 1,
    "CreatedBy": "arn:aws:iam::123456789012:user/Bob",
    "CreateTime": "2020-01-07T19:08:36.000Z"
  }
}
```

```
}  
}
```

Untuk informasi selengkapnya, lihat [Memulihkan Volume Amazon EBS dari Snapshot dan Enkripsi secara Default](#) di Panduan Pengguna Amazon Elastic Compute Cloud.

- Untuk detail API, lihat [CreateLaunchTemplate](#) di Referensi AWS CLI Perintah.

JavaScript

SDK untuk JavaScript (v3)

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
const ssmClient = new SSMClient({});  
const { Parameter } = await ssmClient.send(  
  new GetParameterCommand({  
    Name: "/aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x86_64-gp2",  
  }),  
);  
const ec2Client = new EC2Client({});  
await ec2Client.send(  
  new CreateLaunchTemplateCommand({  
    LaunchTemplateName: NAMES.launchTemplateName,  
    LaunchTemplateData: {  
      InstanceType: "t3.micro",  
      ImageId: Parameter.Value,  
      IamInstanceProfile: { Name: NAMES.instanceProfileName },  
      UserData: readFileSync(  
        join(RESOURCES_PATH, "server_startup_script.sh"),  
      ).toString("base64"),  
      KeyName: NAMES.keyPairName,  
    },  
  }),  
);
```

- Untuk detail API, lihat [CreateLaunchTemplate](#) di Referensi AWS SDK for JavaScript API.

Python

SDK untuk Python (Boto3)

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturannya dan menjalankannya di [Repositori Contoh Kode AWS](#).

Contoh ini membuat templat peluncuran yang menyertakan profil instans yang memberikan izin khusus ke instans, dan skrip Bash data pengguna yang berjalan pada instans tersebut setelah dimulai.

```
class AutoScaler:
    """
    Encapsulates Amazon EC2 Auto Scaling and EC2 management actions.
    """

    def __init__(
        self,
        resource_prefix,
        inst_type,
        ami_param,
        autoscaling_client,
        ec2_client,
        ssm_client,
        iam_client,
    ):
        """
        :param resource_prefix: The prefix for naming AWS resources that are
        created by this class.
        :param inst_type: The type of EC2 instance to create, such as t3.micro.
        :param ami_param: The Systems Manager parameter used to look up the AMI
        that is
            created.
        :param autoscaling_client: A Boto3 EC2 Auto Scaling client.
        :param ec2_client: A Boto3 EC2 client.
        :param ssm_client: A Boto3 Systems Manager client.
        :param iam_client: A Boto3 IAM client.
        """
        self.inst_type = inst_type
```

```

self.ami_param = ami_param
self.autoscaling_client = autoscaling_client
self.ec2_client = ec2_client
self.ssm_client = ssm_client
self.iam_client = iam_client
self.launch_template_name = f"{resource_prefix}-template"
self.group_name = f"{resource_prefix}-group"
self.instance_policy_name = f"{resource_prefix}-pol"
self.instance_role_name = f"{resource_prefix}-role"
self.instance_profile_name = f"{resource_prefix}-prof"
self.bad_creds_policy_name = f"{resource_prefix}-bc-pol"
self.bad_creds_role_name = f"{resource_prefix}-bc-role"
self.bad_creds_profile_name = f"{resource_prefix}-bc-prof"
self.key_pair_name = f"{resource_prefix}-key-pair"

def create_template(self, server_startup_script_file, instance_policy_file):
    """
    Creates an Amazon EC2 launch template to use with Amazon EC2 Auto
    Scaling. The
    launch template specifies a Bash script in its user data field that runs
    after
    the instance is started. This script installs Python packages and starts
    a
    Python web server on the instance.

    :param server_startup_script_file: The path to a Bash script file that is
    run
                                     when an instance starts.
    :param instance_policy_file: The path to a file that defines a
    permissions policy
                                 to create and attach to the instance
    profile.
    :return: Information about the newly created template.
    """
    template = {}
    try:
        self.create_key_pair(self.key_pair_name)
        self.create_instance_profile(
            instance_policy_file,
            self.instance_policy_name,
            self.instance_role_name,
            self.instance_profile_name,
        )

```

```
with open(server_startup_script_file) as file:
    start_server_script = file.read()
ami_latest = self.ssm_client.get_parameter(Name=self.ami_param)
ami_id = ami_latest["Parameter"]["Value"]
lt_response = self.ec2_client.create_launch_template(
    LaunchTemplateName=self.launch_template_name,
    LaunchTemplateData={
        "InstanceType": self.inst_type,
        "ImageId": ami_id,
        "IamInstanceProfile": {"Name": self.instance_profile_name},
        "UserData": base64.b64encode(
            start_server_script.encode(encoding="utf-8")
        ).decode(encoding="utf-8"),
        "KeyName": self.key_pair_name,
    },
)
template = lt_response["LaunchTemplate"]
log.info(
    "Created launch template %s for AMI %s on %s.",
    self.launch_template_name,
    ami_id,
    self.inst_type,
)
except ClientError as err:
    if (
        err.response["Error"]["Code"]
        == "InvalidLaunchTemplateName.AlreadyExistsException"
    ):
        log.info(
            "Launch template %s already exists, nothing to do.",
            self.launch_template_name,
        )
    else:
        raise AutoScalerError(
            f"Couldn't create launch template
{self.launch_template_name}: {err}."
        )
    return template
```

- Untuk detail API, lihat [CreateLaunchTemplate](#) di AWS SDK for Python (Boto3) Referensi API.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan `CreateNetworkAcl` dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `CreateNetworkAcl`.

CLI

AWS CLI

Untuk membuat jaringan ACL

Contoh ini membuat ACL jaringan untuk VPC tertentu.

Perintah:

```
aws ec2 create-network-acl --vpc-id vpc-a01106c2
```

Output:

```
{
  "NetworkAcl": {
    "Associations": [],
    "NetworkAclId": "acl-5fb85d36",
    "VpcId": "vpc-a01106c2",
    "Tags": [],
    "Entries": [
      {
        "CidrBlock": "0.0.0.0/0",
        "RuleNumber": 32767,
        "Protocol": "-1",
        "Egress": true,
        "RuleAction": "deny"
      },
      {
        "CidrBlock": "0.0.0.0/0",
        "RuleNumber": 32767,
        "Protocol": "-1",
        "Egress": false,
        "RuleAction": "deny"
      }
    ]
  }
}
```

```
    ],  
    "IsDefault": false  
  }  
}
```

- Untuk detail API, lihat [CreateNetworkAcl](#) di Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini membuat ACL jaringan untuk VPC yang ditentukan.

```
New-EC2NetworkAcl -VpcId vpc-12345678
```

Output:

```
Associations : {}  
Entries      : {Amazon.EC2.Model.NetworkAclEntry,  
               Amazon.EC2.Model.NetworkAclEntry}  
IsDefault    : False  
NetworkAclId : acl-12345678  
Tags         : {}  
VpcId        : vpc-12345678
```

- Untuk detail API, lihat [CreateNetworkAcl](#) di Referensi AWS Tools for PowerShell Cmdlet.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **CreateNetworkAclEntry** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `CreateNetworkAclEntry`.

CLI

AWS CLI

Untuk membuat entri ACL jaringan

Contoh ini membuat entri untuk ACL jaringan tertentu. Aturan ini memungkinkan masuknya lalu lintas dari alamat IPv4 (0.0.0.0/0) pada port UDP 53 (DNS) ke subnet terkait. Jika perintah berhasil, tidak ada output yang akan ditampilkan.

Perintah:

```
aws ec2 create-network-acl-entry --network-acl-id acl-5fb85d36 --ingress --rule-number 100 --protocol udp --port-range From=53,To=53 --cidr-block 0.0.0.0/0 --rule-action allow
```

Contoh ini membuat aturan untuk ACL jaringan tertentu yang memungkinkan lalu lintas masuk dari alamat IPv6 (:: /0) pada port TCP 80 (HTTP).

Perintah:

```
aws ec2 create-network-acl-entry --network-acl-id acl-5fb85d36 --ingress --rule-number 120 --protocol tcp --port-range From=80,To=80 --ipv6-cidr-block ::/0 --rule-action allow
```

- Untuk detail API, lihat [CreateNetworkAclEntry](#) di Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini membuat entri untuk ACL jaringan tertentu. Aturan ini memungkinkan lalu lintas masuk dari mana saja (0.0.0.0/0) pada port UDP 53 (DNS) ke subnet terkait.

```
New-EC2NetworkAclEntry -NetworkAclId acl-12345678 -Egress $false -RuleNumber 100 -Protocol 17 -PortRange_From 53 -PortRange_To 53 -CidrBlock 0.0.0.0/0 -RuleAction allow
```

- Untuk detail API, lihat [CreateNetworkAclEntry](#) di Referensi AWS Tools for PowerShell Cmdlet.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan `CreateNetworkInterface` dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `CreateNetworkInterface`.

CLI

AWS CLI

Contoh 1: Untuk menentukan alamat IPv4 untuk antarmuka jaringan

`create-network-interface` Contoh berikut membuat antarmuka jaringan untuk subnet tertentu dengan alamat IPv4 primer yang ditentukan.

```
aws ec2 create-network-interface \  
  --subnet-id subnet-00a24d0d67acf6333 \  
  --description "my network interface" \  
  --groups sg-09dfba7ed20cda78b \  
  --private-ip-address 10.0.8.17
```

Output:

```
{  
  "NetworkInterface": {  
    "AvailabilityZone": "us-west-2a",  
    "Description": "my network interface",  
    "Groups": [  
      {  
        "GroupName": "my-security-group",  
        "GroupId": "sg-09dfba7ed20cda78b"  
      }  
    ],  
    "InterfaceType": "interface",  
    "Ipv6Addresses": [],  
    "MacAddress": "06:6a:0f:9a:49:37",  
    "NetworkInterfaceId": "eni-0492b355f0cf3b3f8",  
    "OwnerId": "123456789012",  
    "PrivateDnsName": "ip-10-0-8-18.us-west-2.compute.internal",  
    "PrivateIpAddress": "10.0.8.17",  
    "PrivateIpAddresses": [  
      {  
        "Primary": true,  
        "PrivateDnsName": "ip-10-0-8-17.us-west-2.compute.internal",
```

```

        "PrivateIpAddress": "10.0.8.17"
      }
    ],
    "RequesterId": "AIDA4Z3Y7GSXTMEXAMPLE",
    "RequesterManaged": false,
    "SourceDestCheck": true,
    "Status": "pending",
    "SubnetId": "subnet-00a24d0d67acf6333",
    "TagSet": [],
    "VpcId": "vpc-02723a0feeb9d57b"
  }
}

```

Contoh 2: Untuk membuat antarmuka jaringan dengan alamat IPv4 dan alamat IPv6

`create-network-interface` Contoh berikut membuat antarmuka jaringan untuk subnet yang ditentukan dengan alamat IPv4 dan alamat IPv6 yang dipilih oleh Amazon EC2.

```

aws ec2 create-network-interface \
  --subnet-id subnet-00a24d0d67acf6333 \
  --description "my dual stack network interface" \
  --ipv6-address-count 1 \
  --groups sg-09dfba7ed20cda78b

```

Output:

```

{
  "NetworkInterface": {
    "AvailabilityZone": "us-west-2a",
    "Description": "my dual stack network interface",
    "Groups": [
      {
        "GroupName": "my-security-group",
        "GroupId": "sg-09dfba7ed20cda78b"
      }
    ],
    "InterfaceType": "interface",
    "Ipv6Addresses": [
      {
        "Ipv6Address": "2600:1f13:cfe:3650:a1dc:237c:393a:4ba7",
        "IsPrimaryIpv6": false
      }
    ],
  },
}

```



```

    "MacAddress": "06:b8:68:d2:b2:2d",
    "NetworkInterfaceId": "eni-05da417453f9a84bf",
    "OwnerId": "123456789012",
    "PrivateDnsName": "ip-10-0-8-18.us-west-2.compute.internal",
    "PrivateIpAddress": "10.0.8.18",
    "PrivateIpAddresses": [
      {
        "Primary": true,
        "PrivateDnsName": "ip-10-0-8-18.us-west-2.compute.internal",
        "PrivateIpAddress": "10.0.8.18"
      }
    ],
    "RequesterId": "AIDA4Z3Y7GSXTMEXAMPLE",
    "RequesterManaged": false,
    "SourceDestCheck": true,
    "Status": "pending",
    "SubnetId": "subnet-00a24d0d67acf6333",
    "TagSet": [],
    "VpcId": "vpc-02723a0feeb9d57b",
    "Ipv6Address": "2600:1f13:cfe:3650:a1dc:237c:393a:4ba7"
  }
}

```

Contoh 3: Untuk membuat antarmuka jaringan dengan opsi konfigurasi pelacakan koneksi `create-network-interface` Contoh berikut membuat antarmuka jaringan dan mengkonfigurasi batas waktu pelacakan koneksi idle.

```

aws ec2 create-network-interface \
  --subnet-id subnet-00a24d0d67acf6333 \
  --groups sg-02e57dbcfe0331c1b \
  --connection-tracking-specification TcpEstablishedTimeout=86400,UdpTimeout=60

```

Output:

```

{
  "NetworkInterface": {
    "AvailabilityZone": "us-west-2a",
    "ConnectionTrackingConfiguration": {
      "TcpEstablishedTimeout": 86400,
      "UdpTimeout": 60
    },
    "Description": ""
  }
}

```

```

    "Groups": [
      {
        "GroupName": "my-security-group",
        "GroupId": "sg-02e57dbcfe0331c1b"
      }
    ],
    "InterfaceType": "interface",
    "Ipv6Addresses": [],
    "MacAddress": "06:4c:53:de:6d:91",
    "NetworkInterfaceId": "eni-0c133586e08903d0b",
    "OwnerId": "123456789012",
    "PrivateDnsName": "ip-10-0-8-94.us-west-2.compute.internal",
    "PrivateIpAddress": "10.0.8.94",
    "PrivateIpAddresses": [
      {
        "Primary": true,
        "PrivateDnsName": "ip-10-0-8-94.us-west-2.compute.internal",
        "PrivateIpAddress": "10.0.8.94"
      }
    ],
    "RequesterId": "AIDA4Z3Y7GSXTMEXAMPLE",
    "RequesterManaged": false,
    "SourceDestCheck": true,
    "Status": "pending",
    "SubnetId": "subnet-00a24d0d67acf6333",
    "TagSet": [],
    "VpcId": "vpc-02723a0feeb9d57b"
  }
}

```

Contoh 4: Untuk membuat Adaptor Kain Elastis

`create-network-interface` Contoh berikut menciptakan EFA.

```

aws ec2 create-network-interface \
  --interface-type efa \
  --subnet-id subnet-00a24d0d67acf6333 \
  --description "my efa" \
  --groups sg-02e57dbcfe0331c1b

```

Output:

```
{
```

```

    "NetworkInterface": {
      "AvailabilityZone": "us-west-2a",
      "Description": "my efa",
      "Groups": [
        {
          "GroupName": "my-efa-sg",
          "GroupId": "sg-02e57dbcfe0331c1b"
        }
      ],
      "InterfaceType": "efa",
      "Ipv6Addresses": [],
      "MacAddress": "06:d7:a4:f7:4d:57",
      "NetworkInterfaceId": "eni-034acc2885e862b65",
      "OwnerId": "123456789012",
      "PrivateDnsName": "ip-10-0-8-180.us-west-2.compute.internal",
      "PrivateIpAddress": "10.0.8.180",
      "PrivateIpAddresses": [
        {
          "Primary": true,
          "PrivateDnsName": "ip-10-0-8-180.us-west-2.compute.internal",
          "PrivateIpAddress": "10.0.8.180"
        }
      ],
      "RequesterId": "AIDA4Z3Y7GSXTMEXAMPLE",
      "RequesterManaged": false,
      "SourceDestCheck": true,
      "Status": "pending",
      "SubnetId": "subnet-00a24d0d67acf6333",
      "TagSet": [],
      "VpcId": "vpc-02723a0feeeb9d57b"
    }
  }
}

```

Untuk informasi selengkapnya, lihat [Antarmuka jaringan elastis](#) di Panduan Pengguna Amazon EC2.

- Untuk detail API, lihat [CreateNetworkInterface](#) di Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini menciptakan antarmuka jaringan yang ditentukan.

```
New-EC2NetworkInterface -SubnetId subnet-1a2b3c4d -Description "my network interface" -Group sg-12345678 -PrivateIpAddress 10.0.0.17
```

Output:

```
Association      :
Attachment      :
AvailabilityZone : us-west-2c
Description     : my network interface
Groups          : {my-security-group}
MacAddress      : 0a:72:bc:1a:cd:7f
NetworkInterfaceId : eni-12345678
OwnerId        : 123456789012
PrivateDnsName  : ip-10-0-0-17.us-west-2.compute.internal
PrivateIpAddress : 10.0.0.17
PrivateIpAddresses : {}
RequesterId     :
RequesterManaged : False
SourceDestCheck : True
Status         : pending
SubnetId       : subnet-1a2b3c4d
TagSet         : {}
VpcId         : vpc-12345678
```

- Untuk detail API, lihat [CreateNetworkInterface](#) di Referensi AWS Tools for PowerShell Cmdlet.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **CreatePlacementGroup** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `CreatePlacementGroup`.

CLI

AWS CLI

Untuk membuat grup penempatan

Perintah contoh ini membuat grup penempatan dengan nama yang ditentukan.

Perintah:

```
aws ec2 create-placement-group --group-name my-cluster --strategy cluster
```

Untuk membuat grup penempatan partisi

Perintah contoh ini membuat grup penempatan partisi bernama HDFS-Group-A dengan lima partisi.

Perintah:

```
aws ec2 create-placement-group --group-name HDFS-Group-A --strategy partition --partition-count 5
```

- Untuk detail API, lihat [CreatePlacementGroup](#) di Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini membuat grup penempatan dengan nama yang ditentukan.

```
New-EC2PlacementGroup -GroupName my-placement-group -Strategy cluster
```

- Untuk detail API, lihat [CreatePlacementGroup](#) di Referensi AWS Tools for PowerShell Cmdlet.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **CreateRoute** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `CreateRoute`.

CLI

AWS CLI

Untuk membuat rute

Contoh ini membuat rute untuk tabel rute yang ditentukan. Rute cocok dengan semua lalu lintas IPv4 ($0.0.0.0/0$) dan merutekannya ke gateway Internet yang ditentukan. Jika perintah berhasil, tidak ada output yang akan ditampilkan.

Perintah:

```
aws ec2 create-route --route-table-id rtb-22574640 --destination-cidr-block 0.0.0.0/0 --gateway-id igw-c0a643a9
```

Perintah contoh ini membuat rute dalam tabel rute `rtb-g8ff4ea2`. Rute ini cocok dengan lalu lintas untuk blok IPv4 CIDR `10.0.0.0/16` dan merutekannya ke koneksi peering VPC, `pcx-111aaa22`. Rute ini memungkinkan lalu lintas diarahkan ke VPC rekan dalam koneksi peering VPC. Jika perintah berhasil, tidak ada output yang akan ditampilkan.

Perintah:

```
aws ec2 create-route --route-table-id rtb-g8ff4ea2 --destination-cidr-block 10.0.0.0/16 --vpc-peering-connection-id pcx-1a2b3c4d
```

Contoh ini membuat rute dalam tabel rute yang ditentukan yang cocok dengan semua lalu lintas IPv6 ($:::/0$) dan merutekannya ke gateway Internet khusus egres yang ditentukan.

Perintah:

```
aws ec2 create-route --route-table-id rtb-dce620b8 --destination-ipv6-cidr-block :::/0 --egress-only-internet-gateway-id eigw-01eadbd45ecd7943f
```

- Untuk detail API, lihat [CreateRoute](#) di Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini menciptakan rute yang ditentukan untuk tabel rute yang ditentukan. Rute cocok dengan semua lalu lintas dan mengirimkannya ke gateway Internet yang ditentukan.

```
New-EC2Route -RouteTableId rtb-1a2b3c4d -DestinationCidrBlock 0.0.0.0/0 -  
GatewayId igw-1a2b3c4d
```

Output:

```
True
```

- Untuk detail API, lihat [CreateRoute](#) di Referensi AWS Tools for PowerShell Cmdlet.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **CreateRouteTable** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `CreateRouteTable`.

CLI

AWS CLI

Untuk membuat tabel rute

Contoh ini membuat tabel rute untuk VPC tertentu.

Perintah:

```
aws ec2 create-route-table --vpc-id vpc-a01106c2
```

Output:

```
{  
  "RouteTable": {  
    "Associations": [],  
    "RouteTableId": "rtb-22574640",  
    "VpcId": "vpc-a01106c2",  
    "PropagatingVgws": [],  
    "Tags": [],  
    "Routes": [  
      {  
        "GatewayId": "local",
```

```

        "DestinationCidrBlock": "10.0.0.0/16",
        "State": "active"
      }
    ]
  }
}

```

- Untuk detail API, lihat [CreateRouteTable](#) di Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini membuat tabel rute untuk VPC yang ditentukan.

```
New-EC2RouteTable -VpcId vpc-12345678
```

Output:

```

Associations      : {}
PropagatingVgws  : {}
Routes           : {}
RouteTableId     : rtb-1a2b3c4d
Tags             : {}
VpcId            : vpc-12345678

```

- Untuk detail API, lihat [CreateRouteTable](#) di Referensi AWS Tools for PowerShell Cmdlet.

Ruby

SDK untuk Ruby

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
require "aws-sdk-ec2"
```



```
# Prerequisites:
#
# - A VPC in Amazon VPC.
# - A subnet in that VPC.
# - A gateway attached to that subnet.
#
# @param ec2_resource [Aws::EC2::Resource] An initialized
#   Amazon Elastic Compute Cloud (Amazon EC2) resource object.
# @param vpc_id [String] The ID of the VPC for the route table.
# @param subnet_id [String] The ID of the subnet for the route table.
# @param gateway_id [String] The ID of the gateway for the route.
# @param destination_cidr_block [String] The destination CIDR block
#   for the route.
# @param tag_key [String] The key portion of the tag for the route table.
# @param tag_value [String] The value portion of the tag for the route table.
# @return [Boolean] true if the route table was created and associated;
#   otherwise, false.
# @example
#   exit 1 unless route_table_created_and_associated?(
#     Aws::EC2::Resource.new(region: 'us-west-2'),
#     'vpc-0b6f769731EXAMPLE',
#     'subnet-03d9303b57EXAMPLE',
#     'igw-06ca90c011EXAMPLE',
#     '0.0.0.0/0',
#     'my-key',
#     'my-value'
#   )
def route_table_created_and_associated?(
  ec2_resource,
  vpc_id,
  subnet_id,
  gateway_id,
  destination_cidr_block,
  tag_key,
  tag_value
)
  route_table = ec2_resource.create_route_table(vpc_id: vpc_id)
  puts "Created route table with ID '#{route_table.id}'."
  route_table.create_tags(
    tags: [
      {
        key: tag_key,
        value: tag_value
      }
    ]
  )
end
```

```

    ]
  )
  puts "Added tags to route table."
  route_table.create_route(
    destination_cidr_block: destination_cidr_block,
    gateway_id: gateway_id
  )
  puts "Created route with destination CIDR block " \
    "'#{destination_cidr_block}' and associated with gateway " \
    "with ID '#{gateway_id}'."
  route_table.associate_with_subnet(subnet_id: subnet_id)
  puts "Associated route table with subnet with ID '#{subnet_id}'."
  return true
rescue StandardError => e
  puts "Error creating or associating route table: #{e.message}"
  puts "If the route table was created but not associated, you should " \
    "clean up by deleting the route table."
  return false
end

# Example usage:
def run_me
  vpc_id = ""
  subnet_id = ""
  gateway_id = ""
  destination_cidr_block = ""
  tag_key = ""
  tag_value = ""
  region = ""
  # Print usage information and then stop.
  if ARGV[0] == "--help" || ARGV[0] == "-h"
    puts "Usage: ruby ec2-ruby-example-create-route-table.rb " \
      "VPC_ID SUBNET_ID GATEWAY_ID DESTINATION_CIDR_BLOCK " \
      "TAG_KEY TAG_VALUE REGION"
  # Replace us-west-2 with the AWS Region you're using for Amazon EC2.
  puts "Example: ruby ec2-ruby-example-create-route-table.rb " \
    "vpc-0b6f769731EXAMPLE subnet-03d9303b57EXAMPLE igw-06ca90c011EXAMPLE " \
    "'0.0.0.0/0' my-key my-value us-west-2"
  exit 1
  # If no values are specified at the command prompt, use these default values.
  elsif ARGV.count.zero?
    vpc_id = "vpc-0b6f769731EXAMPLE"
    subnet_id = "subnet-03d9303b57EXAMPLE"
    gateway_id = "igw-06ca90c011EXAMPLE"

```

```
destination_cidr_block = "0.0.0.0/0"
tag_key = "my-key"
tag_value = "my-value"
# Replace us-west-2 with the AWS Region you're using for Amazon EC2.
region = "us-west-2"
# Otherwise, use the values as specified at the command prompt.
else
  vpc_id = ARGV[0]
  subnet_id = ARGV[1]
  gateway_id = ARGV[2]
  destination_cidr_block = ARGV[3]
  tag_key = ARGV[4]
  tag_value = ARGV[5]
  region = ARGV[6]
end

ec2_resource = Aws::EC2::Resource.new(region: region)

if route_table_created_and_associated?(
  ec2_resource,
  vpc_id,
  subnet_id,
  gateway_id,
  destination_cidr_block,
  tag_key,
  tag_value
)
  puts "Route table created and associated."
else
  puts "Route table not created or not associated."
end
end

run_me if $PROGRAM_NAME == __FILE__
```

- Untuk detail API, lihat [CreateRouteTable](#) di Referensi AWS SDK for Ruby API.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan `CreateSecurityGroup` dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `CreateSecurityGroup`.

Contoh tindakan adalah kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Anda dapat melihat tindakan ini dalam konteks dalam contoh kode berikut:

- [Memulai instans](#)

.NET

AWS SDK for .NET

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).


```
/// <summary>
/// Create an Amazon EC2 security group.
/// </summary>
/// <param name="groupName">The name for the new security group.</param>
/// <param name="groupDescription">A description of the new security group.</
param>
/// <returns>The group Id of the new security group.</returns>
public async Task<string> CreateSecurityGroup(string groupName, string
groupDescription)
{
    var response = await _amazonEC2.CreateSecurityGroupAsync(
        new CreateSecurityGroupRequest(groupName, groupDescription));

    return response.GroupId;
}
```

- Untuk detail API, lihat [CreateSecurityGroup](#) di Referensi AWS SDK for .NET API.

Bash

AWS CLI dengan skrip Bash

 Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
#####
# function ec2_create_security_group
#
# This function creates an Amazon Elastic Compute Cloud (Amazon EC2) security
# group.
#
# Parameters:
#     -n security_group_name - The name of the security group.
#     -d security_group_description - The description of the security group.
#
# Returns:
#     The ID of the created security group, or an error message if the
#     operation fails.
# And:
#     0 - If successful.
#     1 - If it fails.
#
#####
function ec2_create_security_group() {
    local security_group_name security_group_description response

    # Function to display usage information
    function usage() {
        echo "function ec2_create_security_group"
        echo "Creates an Amazon Elastic Compute Cloud (Amazon EC2) security group."
        echo "  -n security_group_name - The name of the security group."
        echo "  -d security_group_description - The description of the security
group."
        echo ""
    }

    # Parse the command-line arguments
```

```
while getopts "n:d:h" option; do
  case "${option}" in
    n) security_group_name="${OPTARG}" ;;
    d) security_group_description="${OPTARG}" ;;
    h)
      usage
      return 0
      ;;
    \?)
      echo "Invalid parameter"
      usage
      return 1
      ;;
  esac
done
export OPTIND=1

# Validate the input parameters
if [[ -z "$security_group_name" ]]; then
  errecho "ERROR: You must provide a security group name with the -n
parameter."
  return 1
fi

if [[ -z "$security_group_description" ]]; then
  errecho "ERROR: You must provide a security group description with the -d
parameter."
  return 1
fi

# Create the security group
response=$(aws ec2 create-security-group \
  --group-name "$security_group_name" \
  --description "$security_group_description" \
  --query "GroupId" \
  --output text) || {
  aws_cli_error_log ${?}
  errecho "ERROR: AWS reports create-security-group operation failed."
  errecho "$response"
  return 1
}

echo "$response"
return 0
```

```
}

```

Fungsi utilitas yang digunakan dalam contoh ini.

```
#####
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
#####
function errecho() {
    printf "%s\n" "$*" 1>&2
}

#####
# function aws_cli_error_log()
#
# This function is used to log the error messages from the AWS CLI.
#
# The function expects the following argument:
#     $1 - The error code returned by the AWS CLI.
#
# Returns:
#     0: - Success.
#
#####
function aws_cli_error_log() {
    local err_code=$1
    errecho "Error code : $err_code"
    if [ "$err_code" == 1 ]; then
        errecho " One or more S3 transfers failed."
    elif [ "$err_code" == 2 ]; then
        errecho " Command line failed to parse."
    elif [ "$err_code" == 130 ]; then
        errecho " Process received SIGINT."
    elif [ "$err_code" == 252 ]; then
        errecho " Command syntax invalid."
    elif [ "$err_code" == 253 ]; then
        errecho " The system environment or configuration was invalid."
    elif [ "$err_code" == 254 ]; then
        errecho " The service returned an error."
    elif [ "$err_code" == 255 ]; then
        errecho " 255 is a catch-all error."
    }
}

```

```
fi

return 0
}
```

- Untuk detail API, lihat [CreateSecurityGroup](#) di Referensi AWS CLI Perintah.

C++

SDK untuk C++

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
Aws::EC2::EC2Client ec2Client(clientConfiguration);

Aws::EC2::Model::CreateSecurityGroupRequest request;

request.SetGroupName(groupName);
request.SetDescription(description);
request.SetVpcId(vpcID);

const Aws::EC2::Model::CreateSecurityGroupOutcome outcome =
    ec2Client.CreateSecurityGroup(request);

if (!outcome.IsSuccess()) {
    std::cerr << "Failed to create security group:" <<
        outcome.GetError().GetMessage() << std::endl;
    return false;
}

std::cout << "Successfully created security group named " << groupName <<
    std::endl;
```

- Untuk detail API, lihat [CreateSecurityGroup](#) di Referensi AWS SDK for C++ API.

CLI

AWS CLI

Guna membuat grup keamanan untuk EC2-Classic

Contoh ini membuat grup keamanan bernama MySecurityGroup.

Perintah:

```
aws ec2 create-security-group --group-name MySecurityGroup --description "My security group"
```

Output:

```
{
  "GroupId": "sg-903004f8"
}
```

Guna membuat grup keamanan untuk EC2-VPC

Contoh ini membuat grup keamanan bernama MySecurityGroup untuk VPC tertentu.

Perintah:

```
aws ec2 create-security-group --group-name MySecurityGroup --description "My security group" --vpc-id vpc-1a2b3c4d
```

Output:

```
{
  "GroupId": "sg-903004f8"
}
```

Untuk informasi selengkapnya, lihat Menggunakan Grup Keamanan di Panduan Pengguna Antarmuka Baris Perintah AWS .

- Untuk detail API, lihat [CreateSecurityGroup](#) di Referensi AWS CLI Perintah.

Java

SDK untuk Java 2.x

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
public static String createSecurityGroup(Ec2Client ec2, String groupName,
String groupDesc, String vpcId,
    String myIpAddress) {
    try {
        CreateSecurityGroupRequest createRequest =
CreateSecurityGroupRequest.builder()
            .groupName(groupName)
            .description(groupDesc)
            .vpcId(vpcId)
            .build();

        CreateSecurityGroupResponse resp =
ec2.createSecurityGroup(createRequest);
        IpRange ipRange = IpRange.builder()
            .cidrIp(myIpAddress + "/0")
            .build();

        IpPermission ipPerm = IpPermission.builder()
            .ipProtocol("tcp")
            .toPort(80)
            .fromPort(80)
            .ipRanges(ipRange)
            .build();

        IpPermission ipPerm2 = IpPermission.builder()
            .ipProtocol("tcp")
            .toPort(22)
            .fromPort(22)
            .ipRanges(ipRange)
            .build();
```

```

        AuthorizeSecurityGroupIngressRequest authRequest =
AuthorizeSecurityGroupIngressRequest.builder()
    .groupName(groupName)
    .ipPermissions(ipPerm, ipPerm2)
    .build();

        ec2.authorizeSecurityGroupIngress(authRequest);
        System.out.println("Successfully added ingress policy to security
group " + groupName);
        return resp.groupId();

    } catch (Ec2Exception e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
    return "";
}
}

```

- Untuk detail API, lihat [CreateSecurityGroup](#) di Referensi AWS SDK for Java 2.x API.

JavaScript

SDK untuk JavaScript (v3)

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```

import { CreateSecurityGroupCommand } from "@aws-sdk/client-ec2";

import { client } from "../libs/client.js";

export const main = async () => {
    const command = new CreateSecurityGroupCommand({
        // Up to 255 characters in length. Cannot start with sg-.
        GroupName: "SECURITY_GROUP_NAME",
        // Up to 255 characters in length.
        Description: "DESCRIPTION",
    });
}

```

```
try {
  const { GroupId } = await client.send(command);
  console.log(GroupId);
} catch (err) {
  console.error(err);
}
};
```

- Untuk detail API, lihat [CreateSecurityGroup](#) di Referensi AWS SDK for JavaScript API.

Kotlin

SDK untuk Kotlin

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
suspend fun createEC2SecurityGroup(
  groupNameVal: String?,
  groupDescVal: String?,
  vpcIdVal: String?,
): String? {
  val request =
    CreateSecurityGroupRequest {
      groupName = groupNameVal
      description = groupDescVal
      vpcId = vpcIdVal
    }

  Ec2Client { region = "us-west-2" }.use { ec2 ->
    val resp = ec2.createSecurityGroup(request)
    val ipRange =
      IpRange {
        cidrIp = "0.0.0.0/0"
      }

    val ipPerm =
```

```
        IpPermission {
            ipProtocol = "tcp"
            toPort = 80
            fromPort = 80
            ipRanges = listOf(ipRange)
        }

    val ipPerm2 =
        IpPermission {
            ipProtocol = "tcp"
            toPort = 22
            fromPort = 22
            ipRanges = listOf(ipRange)
        }

    val authRequest =
        AuthorizeSecurityGroupIngressRequest {
            groupName = groupNameVal
            ipPermissions = listOf(ipPerm, ipPerm2)
        }
    ec2.authorizeSecurityGroupIngress(authRequest)
    println("Successfully added ingress policy to Security Group
    $groupNameVal")
    return resp.groupId
}
}
```

- Untuk detail API, lihat [CreateSecurityGroup](#) di AWS SDK untuk referensi API Kotlin.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini membuat grup keamanan untuk VPC yang ditentukan.

```
New-EC2SecurityGroup -GroupName my-security-group -Description "my security
group" -VpcId vpc-12345678
```

Output:

```
sg-12345678
```

Contoh 2: Contoh ini membuat grup keamanan untuk EC2-Classic.

```
New-EC2SecurityGroup -GroupName my-security-group -Description "my security group"
```

Output:

```
sg-45678901
```

- Untuk detail API, lihat [CreateSecurityGroup](#) di Referensi AWS Tools for PowerShell Cmdlet.

Python

SDK untuk Python (Boto3)

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
class SecurityGroupWrapper:
    """Encapsulates Amazon Elastic Compute Cloud (Amazon EC2) security group
    actions."""

    def __init__(self, ec2_resource, security_group=None):
        """
        :param ec2_resource: A Boto3 Amazon EC2 resource. This high-level
        resource
                               is used to create additional high-level objects
                               that wrap low-level Amazon EC2 service actions.
        :param security_group: A Boto3 SecurityGroup object. This is a high-level
        object
                               that wraps security group actions.
        """
        self.ec2_resource = ec2_resource
        self.security_group = security_group

    @classmethod
    def from_resource(cls):
        ec2_resource = boto3.resource("ec2")
```

```
        return cls(ec2_resource)

    def create(self, group_name, group_description):
        """
        Creates a security group in the default virtual private cloud (VPC) of
the
        current account.

        :param group_name: The name of the security group to create.
        :param group_description: The description of the security group to
create.
        :return: A Boto3 SecurityGroup object that represents the newly created
security group.
        """
        try:
            self.security_group = self.ec2_resource.create_security_group(
                GroupName=group_name, Description=group_description
            )
        except ClientError as err:
            logger.error(
                "Couldn't create security group %s. Here's why: %s: %s",
                group_name,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
        else:
            return self.security_group
```

- Untuk detail API, lihat [CreateSecurityGroup](#) di AWS SDK for Python (Boto3) Referensi API.

Ruby

SDK untuk Ruby

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
# This code example does the following:
# 1. Creates an Amazon Elastic Compute Cloud (Amazon EC2) security group.
# 2. Adds inbound rules to the security group.
# 3. Displays information about available security groups.
# 4. Deletes the security group.

require "aws-sdk-ec2"

# Creates an Amazon Elastic Compute Cloud (Amazon EC2) security group.
#
# Prerequisites:
#
# - A VPC in Amazon Virtual Private Cloud (Amazon VPC).
#
# @param ec2_client [Aws::EC2::Client] An initialized
#   Amazon EC2 client.
# @param group_name [String] A name for the security group.
# @param description [String] A description for the security group.
# @param vpc_id [String] The ID of the VPC for the security group.
# @return [String] The ID of security group that was created.
# @example
#   puts create_security_group(
#     Aws::EC2::Client.new(region: 'us-west-2'),
#     'my-security-group',
#     'This is my security group.',
#     'vpc-6713dfEX'
#   )
def create_security_group(
  ec2_client,
  group_name,
  description,
  vpc_id
)
  security_group = ec2_client.create_security_group(
    group_name: group_name,
    description: description,
    vpc_id: vpc_id
  )
  puts "Created security group '#{group_name}' with ID " \
    "'#{security_group.group_id}' in VPC with ID '#{vpc_id}'."
  return security_group.group_id
rescue StandardError => e
```



```
puts "Error creating security group: #{e.message}"
return "Error"
end

# Adds an inbound rule to an Amazon Elastic Compute Cloud (Amazon EC2)
# security group.
#
# Prerequisites:
#
# - The security group.
#
# @param ec2_client [Aws::EC2::Client] An initialized Amazon EC2 client.
# @param security_group_id [String] The ID of the security group.
# @param ip_protocol [String] The network protocol for the inbound rule.
# @param from_port [String] The originating port for the inbound rule.
# @param to_port [String] The destination port for the inbound rule.
# @param cidr_ip_range [String] The CIDR IP range for the inbound rule.
# @return
# @example
#   exit 1 unless security_group_ingress_authorized?(
#     Aws::EC2::Client.new(region: 'us-west-2'),
#     'sg-030a858e078f1b9EX',
#     'tcp',
#     '80',
#     '80',
#     '0.0.0.0/0'
#   )
def security_group_ingress_authorized?(
  ec2_client,
  security_group_id,
  ip_protocol,
  from_port,
  to_port,
  cidr_ip_range
)
  ec2_client.authorize_security_group_ingress(
    group_id: security_group_id,
    ip_permissions: [
      {
        ip_protocol: ip_protocol,
        from_port: from_port,
        to_port: to_port,
        ip_ranges: [
          {
```

```

        cidr_ip: cidr_ip_range
      }
    ]
  }
]
)
puts "Added inbound rule to security group '#{security_group_id}' for protocol
" \
  "'#{ip_protocol}' from port '#{from_port}' to port '#{to_port}' " \
  "with CIDR IP range '#{cidr_ip_range}'."
return true
rescue StandardError => e
  puts "Error adding inbound rule to security group: #{e.message}"
  return false
end

# Displays information about a security group's IP permissions set in
# Amazon Elastic Compute Cloud (Amazon EC2).
#
# Prerequisites:
#
# - A security group with inbound rules, outbound rules, or both.
#
# @param p [Aws::EC2::Types::IpPermission] The IP permissions set.
# @example
#   ec2_client = Aws::EC2::Client.new(region: 'us-west-2')
#   response = ec2_client.describe_security_groups
#   unless sg.ip_permissions.empty?
#     describe_security_group_permissions(
#       response.security_groups[0].ip_permissions[0]
#     )
#   end
def describe_security_group_permissions(perm)
  print " Protocol: #{perm.ip_protocol == '-1' ? 'All' : perm.ip_protocol}"

  unless perm.from_port.nil?
    if perm.from_port == "-1" || perm.from_port == -1
      print ", From: All"
    else
      print ", From: #{perm.from_port}"
    end
  end
end

unless perm.to_port.nil?

```

```
    if perm.to_port == "-1" || perm.to_port == -1
      print ", To: All"
    else
      print ", To: #{perm.to_port}"
    end
  end
end

if perm.key?(:ipv_6_ranges) && perm.ipv_6_ranges.count.positive?
  print ", CIDR IPv6: #{perm.ipv_6_ranges[0].cidr_ipv_6}"
end

if perm.key?(:ip_ranges) && perm.ip_ranges.count.positive?
  print ", CIDR IPv4: #{perm.ip_ranges[0].cidr_ip}"
end

print "\n"
end

# Displays information about available security groups in
# Amazon Elastic Compute Cloud (Amazon EC2).
#
# @param ec2_client [Aws::EC2::Client] An initialized Amazon EC2 client.
# @example
#   describe_security_groups(Aws::EC2::Client.new(region: 'us-west-2'))
def describe_security_groups(ec2_client)
  response = ec2_client.describe_security_groups

  if response.security_groups.count.positive?
    response.security_groups.each do |sg|
      puts "-" * (sg.group_name.length + 13)
      puts "Name:          #{sg.group_name}"
      puts "Description:  #{sg.description}"
      puts "Group ID:     #{sg.group_id}"
      puts "Owner ID:     #{sg.owner_id}"
      puts "VPC ID:       #{sg.vpc_id}"

      if sg.tags.count.positive?
        puts "Tags:"
        sg.tags.each do |tag|
          puts "  Key: #{tag.key}, Value: #{tag.value}"
        end
      end
    end

    unless sg.ip_permissions.empty?
```

```
    puts "Inbound rules:" if sg.ip_permissions.count.positive?
    sg.ip_permissions.each do |p|
      describe_security_group_permissions(p)
    end
  end
end

unless sg.ip_permissions_egress.empty?
  puts "Outbound rules:" if sg.ip_permissions.count.positive?
  sg.ip_permissions_egress.each do |p|
    describe_security_group_permissions(p)
  end
end
end
else
  puts "No security groups found."
end
rescue StandardError => e
  puts "Error getting information about security groups: #{e.message}"
end

# Deletes an Amazon Elastic Compute Cloud (Amazon EC2)
# security group.
#
# Prerequisites:
#
# - The security group.
#
# @param ec2_client [Aws::EC2::Client] An initialized
#   Amazon EC2 client.
# @param security_group_id [String] The ID of the security group to delete.
# @return [Boolean] true if the security group was deleted; otherwise, false.
# @example
#   exit 1 unless security_group_deleted?(
#     Aws::EC2::Client.new(region: 'us-west-2'),
#     'sg-030a858e078f1b9EX'
#   )
def security_group_deleted?(ec2_client, security_group_id)
  ec2_client.delete_security_group(group_id: security_group_id)
  puts "Deleted security group '#{security_group_id}'."
  return true
rescue StandardError => e
  puts "Error deleting security group: #{e.message}"
  return false
end
end
```

```
# Example usage:
def run_me
  group_name = ""
  description = ""
  vpc_id = ""
  ip_protocol_http = ""
  from_port_http = ""
  to_port_http = ""
  cidr_ip_range_http = ""
  ip_protocol_ssh = ""
  from_port_ssh = ""
  to_port_ssh = ""
  cidr_ip_range_ssh = ""
  region = ""
  # Print usage information and then stop.
  if ARGV[0] == "--help" || ARGV[0] == "-h"
    puts "Usage:  ruby ec2-ruby-example-security-group.rb " \
      "GROUP_NAME DESCRIPTION VPC_ID IP_PROTOCOL_1 FROM_PORT_1 TO_PORT_1 " \
      "CIDR_IP_RANGE_1 IP_PROTOCOL_2 FROM_PORT_2 TO_PORT_2 " \
      "CIDR_IP_RANGE_2 REGION"
    puts "Example: ruby ec2-ruby-example-security-group.rb " \
      "my-security-group 'This is my security group.' vpc-6713dfEX " \
      "tcp 80 80 '0.0.0.0/0' tcp 22 22 '0.0.0.0/0' us-west-2"
    exit 1
  # If no values are specified at the command prompt, use these default values.
  elsif ARGV.count.zero?
    group_name = "my-security-group"
    description = "This is my security group."
    vpc_id = "vpc-6713dfEX"
    ip_protocol_http = "tcp"
    from_port_http = "80"
    to_port_http = "80"
    cidr_ip_range_http = "0.0.0.0/0"
    ip_protocol_ssh = "tcp"
    from_port_ssh = "22"
    to_port_ssh = "22"
    cidr_ip_range_ssh = "0.0.0.0/0"
    # Replace us-west-2 with the AWS Region you're using for Amazon EC2.
    region = "us-west-2"
  # Otherwise, use the values as specified at the command prompt.
  else
    group_name = ARGV[0]
    description = ARGV[1]
```

```
vpc_id = ARGV[2]
ip_protocol_http = ARGV[3]
from_port_http = ARGV[4]
to_port_http = ARGV[5]
cidr_ip_range_http = ARGV[6]
ip_protocol_ssh = ARGV[7]
from_port_ssh = ARGV[8]
to_port_ssh = ARGV[9]
cidr_ip_range_ssh = ARGV[10]
region = ARGV[11]
end

security_group_id = ""
security_group_exists = false
ec2_client = Aws::EC2::Client.new(region: region)

puts "Attempting to create security group..."
security_group_id = create_security_group(
  ec2_client,
  group_name,
  description,
  vpc_id
)
if security_group_id == "Error"
  puts "Could not create security group. Skipping this step."
else
  security_group_exists = true
end

if security_group_exists
  puts "Attempting to add inbound rules to security group..."
  unless security_group_ingress_authorized?(
    ec2_client,
    security_group_id,
    ip_protocol_http,
    from_port_http,
    to_port_http,
    cidr_ip_range_http
  )
    puts "Could not add inbound HTTP rule to security group. " \
      "Skipping this step."
  end

  unless security_group_ingress_authorized?(
```

```

    ec2_client,
    security_group_id,
    ip_protocol_ssh,
    from_port_ssh,
    to_port_ssh,
    cidr_ip_range_ssh
  )
  puts "Could not add inbound SSH rule to security group. " \
    "Skipping this step."
end
end

puts "\nInformation about available security groups:"
describe_security_groups(ec2_client)

if security_group_exists
  puts "\nAttempting to delete security group..."
  unless security_group_deleted?(ec2_client, security_group_id)
    puts "Could not delete security group. You must delete it yourself."
  end
end
end
end

run_me if $PROGRAM_NAME == __FILE__

```

- Untuk detail API, lihat [CreateSecurityGroup](#) di Referensi AWS SDK for Ruby API.

SAP ABAP

SDK untuk SAP ABAP

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```

TRY.
  oo_result = lo_ec2->createsecuritygroup(
    iv_description = 'Security group example'
  )
  " oo_result is
returned for testing purposes. "

```

```
        iv_groupname = iv_security_group_name
        iv_vpcid = iv_vpc_id
    ).
    MESSAGE 'Security group created.' TYPE 'I'.
    CATCH /aws1/cx_rt_service_generic INTO DATA(lo_exception).
    DATA(lv_error) = |"{ lo_exception->av_err_code }" - { lo_exception-
>av_err_msg }|.
    MESSAGE lv_error TYPE 'E'.
ENDTRY.
```

- Untuk detail API, lihat [CreateSecurityGroup](#) di AWS SDK untuk referensi SAP ABAP API.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **CreateSnapshot** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `CreateSnapshot`.

CLI

AWS CLI

Untuk membuat snapshot

Perintah contoh ini membuat snapshot volume dengan ID volume `vol-1234567890abcdef0` dan deskripsi singkat untuk mengidentifikasi snapshot.

Perintah:

```
aws ec2 create-snapshot --volume-id vol-1234567890abcdef0 --description "This is my root volume snapshot"
```

Output:

```
{
  "Description": "This is my root volume snapshot",
  "Tags": [],
  "Encrypted": false,
```



```
"VolumeId": "vol-1234567890abcdef0",
"State": "pending",
"VolumeSize": 8,
"StartTime": "2018-02-28T21:06:01.000Z",
"Progress": "",
"OwnerId": "012345678910",
"SnapshotId": "snap-066877671789bd71b"
}
```

Untuk membuat snapshot dengan tag

Perintah contoh ini membuat snapshot dan menerapkan dua tag: `purpose=prod` dan `costcenter=123`.

Perintah:

```
aws ec2 create-snapshot --volume-id vol-1234567890abcdef0
--description 'Prod backup' --tag-specifications
'ResourceType=snapshot,Tags=[{Key=purpose,Value=prod},
{Key=costcenter,Value=123}]'
```

Output:

```
{
  "Description": "Prod backup",
  "Tags": [
    {
      "Value": "prod",
      "Key": "purpose"
    },
    {
      "Value": "123",
      "Key": "costcenter"
    }
  ],
  "Encrypted": false,
  "VolumeId": "vol-1234567890abcdef0",
  "State": "pending",
  "VolumeSize": 8,
  "StartTime": "2018-02-28T21:06:06.000Z",
  "Progress": "",
  "OwnerId": "012345678910",
  "SnapshotId": "snap-09ed24a70bc19bbe4"
```

```
}
```

- Untuk detail API, lihat [CreateSnapshot](#) di Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini membuat snapshot dari volume yang ditentukan.

```
New-EC2Snapshot -VolumeId vol-12345678 -Description "This is a test"
```

Output:

```
DataEncryptionKeyId :  
Description          : This is a test  
Encrypted            : False  
KmsKeyId             :  
OwnerAlias           :  
OwnerId              : 123456789012  
Progress             :  
SnapshotId           : snap-12345678  
StartTime            : 12/22/2015 1:28:42 AM  
State                : pending  
StateMessage         :  
Tags                 : {}  
VolumeId             : vol-12345678  
VolumeSize           : 20
```

- Untuk detail API, lihat [CreateSnapshot](#) di Referensi AWS Tools for PowerShell Cmdlet.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **CreateSpotDatafeedSubscription** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `CreateSpotDatafeedSubscription`.

CLI

AWS CLI

Untuk membuat umpan data Instance Spot

`create-spot-datafeed-subscription` Contoh berikut membuat umpan data Spot Instance.

```
aws ec2 create-spot-datafeed-subscription \  
  --bucket my-bucket \  
  --prefix spot-data-feed
```

Output:

```
{  
  "SpotDatafeedSubscription": {  
    "Bucket": "my-bucket",  
    "OwnerId": "123456789012",  
    "Prefix": "spot-data-feed",  
    "State": "Active"  
  }  
}
```

Umpan data disimpan di bucket Amazon S3 yang Anda tentukan. Nama file untuk umpan data ini memiliki format berikut.

```
my-bucket.s3.amazonaws.com/spot-data-feed/123456789012.YYYY-MM-DD-  
HH.n.abcd1234.gz
```

Untuk informasi selengkapnya, lihat [umpan data Instans Spot](#) di Panduan Pengguna Amazon Elastic Compute Cloud untuk Instans Linux.

- Untuk detail API, lihat [CreateSpotDatafeedSubscription](#) di Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini membuat umpan data instance Spot.

```
New-EC2SpotDatafeedSubscription -Bucket my-s3-bucket -Prefix spotdata
```

Output:

```
Bucket   : my-s3-bucket
Fault    :
OwnerId  : 123456789012
Prefix   : spotdata
State    : Active
```

- Untuk detail API, lihat [CreateSpotDatafeedSubscription](#) di Referensi AWS Tools for PowerShell Cmdlet.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **CreateSubnet** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `CreateSubnet`.

CLI

AWS CLI

Contoh 1: Untuk membuat subnet dengan blok CIDR IPv4 saja

Contoh `create-subnet` berikut membuat subnet di VPC tertentu dengan blok CIDR IPv4 tertentu.

```
aws ec2 create-subnet \  
  --vpc-id vpc-081ec835f3EXAMPLE \  
  --cidr-block 10.0.0.0/24 \  
  --tag-specifications ResourceType=subnet,Tags=[{Key=Name,Value=my-ipv4-only-  
subnet}]
```

Output:

```
{  
  "Subnet": {
```

```

    "AvailabilityZone": "us-west-2a",
    "AvailabilityZoneId": "usw2-az2",
    "AvailableIpAddressCount": 251,
    "CidrBlock": "10.0.0.0/24",
    "DefaultForAz": false,
    "MapPublicIpOnLaunch": false,
    "State": "available",
    "SubnetId": "subnet-0e99b93155EXAMPLE",
    "VpcId": "vpc-081ec835f3EXAMPLE",
    "OwnerId": "123456789012",
    "AssignIpv6AddressOnCreation": false,
    "Ipv6CidrBlockAssociationSet": [],
    "Tags": [
      {
        "Key": "Name",
        "Value": "my-ipv4-only-subnet"
      }
    ],
    "SubnetArn": "arn:aws:ec2:us-west-2:123456789012:subnet/
subnet-0e99b93155EXAMPLE"
  }
}

```

Contoh 2: Untuk membuat subnet dengan blok CIDR IPv4 dan IPv6

Contoh `create-subnet` berikut membuat subnet di VPC tertentu dengan blok CIDR IPv4 dan IPv6 tertentu.

```

aws ec2 create-subnet \
  --vpc-id vpc-081ec835f3EXAMPLE \
  --cidr-block 10.0.0.0/24 \
  --ipv6-cidr-block 2600:1f16:cfe:3660::/64 \
  --tag-specifications ResourceType=subnet,Tags=[{Key=Name,Value=my-ipv4-ipv6-
subnet}]

```

Output:

```

{
  "Subnet": {
    "AvailabilityZone": "us-west-2a",
    "AvailabilityZoneId": "usw2-az2",
    "AvailableIpAddressCount": 251,
    "CidrBlock": "10.0.0.0/24",

```

```

    "DefaultForAz": false,
    "MapPublicIpOnLaunch": false,
    "State": "available",
    "SubnetId": "subnet-0736441d38EXAMPLE",
    "VpcId": "vpc-081ec835f3EXAMPLE",
    "OwnerId": "123456789012",
    "AssignIpv6AddressOnCreation": false,
    "Ipv6CidrBlockAssociationSet": [
      {
        "AssociationId": "subnet-cidr-assoc-06c5f904499fcc623",
        "Ipv6CidrBlock": "2600:1f13:cfe:3660::/64",
        "Ipv6CidrBlockState": {
          "State": "associating"
        }
      }
    ],
    "Tags": [
      {
        "Key": "Name",
        "Value": "my-ipv4-ipv6-subnet"
      }
    ],
    "SubnetArn": "arn:aws:ec2:us-west-2:123456789012:subnet/
subnet-0736441d38EXAMPLE"
  }
}

```

Contoh 3: Untuk membuat subnet dengan blok CIDR IPv6 saja

Contoh `create-subnet` berikut membuat subnet di VPC tertentu dengan blok CIDR IPv6 tertentu.

```

aws ec2 create-subnet \
  --vpc-id vpc-081ec835f3EXAMPLE \
  --ipv6-native \
  --ipv6-cidr-block 2600:1f16:115:200::/64 \
  --tag-specifications ResourceType=subnet,Tags=[{Key=Name,Value=my-ipv6-only-
subnet}]

```

Output:

```

{
  "Subnet": {

```

```

    "AvailabilityZone": "us-west-2a",
    "AvailabilityZoneId": "usw2-az2",
    "AvailableIpAddressCount": 0,
    "DefaultForAz": false,
    "MapPublicIpOnLaunch": false,
    "State": "available",
    "SubnetId": "subnet-03f720e7deEXAMPLE",
    "VpcId": "vpc-081ec835f3EXAMPLE",
    "OwnerId": "123456789012",
    "AssignIpv6AddressOnCreation": true,
    "Ipv6CidrBlockAssociationSet": [
      {
        "AssociationId": "subnet-cidr-assoc-01ef639edde556709",
        "Ipv6CidrBlock": "2600:1f13:cfe:3660::/64",
        "Ipv6CidrBlockState": {
          "State": "associating"
        }
      }
    ],
    "Tags": [
      {
        "Key": "Name",
        "Value": "my-ipv6-only-subnet"
      }
    ],
    "SubnetArn": "arn:aws:ec2:us-west-2:123456789012:subnet/
subnet-03f720e7deEXAMPLE"
  }
}

```

Untuk informasi selengkapnya, lihat [VPC dan subnet](#) di Panduan Pengguna Amazon VPC.

- Untuk detail API, lihat [CreateSubnet](#) di Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini membuat subnet dengan CIDR yang ditentukan.

```
New-EC2Subnet -VpcId vpc-12345678 -CidrBlock 10.0.0.0/24
```

Output:

```
AvailabilityZone      : us-west-2c
AvailableIpAddressCount : 251
CidrBlock             : 10.0.0.0/24
DefaultForAz         : False
MapPublicIpOnLaunch  : False
State                 : pending
SubnetId              : subnet-1a2b3c4d
Tag                   : {}
VpcId                 : vpc-12345678
```

- Untuk detail API, lihat [CreateSubnet](#) di Referensi AWS Tools for PowerShell Cmdlet.

Ruby

SDK untuk Ruby

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
require "aws-sdk-ec2"

# Creates a subnet within a virtual private cloud (VPC) in
# Amazon Virtual Private Cloud (Amazon VPC) and then tags
# the subnet.
#
# Prerequisites:
#
# - A VPC in Amazon VPC.
#
# @param ec2_resource [Aws::EC2::Resource] An initialized
#   Amazon Elastic Compute Cloud (Amazon EC2) resource object.
# @param vpc_id [String] The ID of the VPC for the subnet.
# @param cidr_block [String] The IPv4 CIDR block for the subnet.
# @param availability_zone [String] The ID of the Availability Zone
#   for the subnet.
# @param tag_key [String] The key portion of the tag for the subnet.
# @param tag_value [String] The value portion of the tag for the subnet.
```



```
# @return [Boolean] true if the subnet was created and tagged;
# otherwise, false.
# @example
# exit 1 unless subnet_created_and_tagged?(
#   Aws::EC2::Resource.new(region: 'us-west-2'),
#   'vpc-6713dfEX',
#   '10.0.0.0/24',
#   'us-west-2a',
#   'my-key',
#   'my-value'
# )
def subnet_created_and_tagged?(
  ec2_resource,
  vpc_id,
  cidr_block,
  availability_zone,
  tag_key,
  tag_value
)
  subnet = ec2_resource.create_subnet(
    vpc_id: vpc_id,
    cidr_block: cidr_block,
    availability_zone: availability_zone
  )
  subnet.create_tags(
    tags: [
      {
        key: tag_key,
        value: tag_value
      }
    ]
  )
  puts "Subnet created with ID '#{subnet.id}' in VPC with ID '#{vpc_id}' " \
    "and CIDR block '#{cidr_block}' in availability zone " \
    "'#{availability_zone}' and tagged with key '#{tag_key}' and " \
    "value '#{tag_value}'."
  return true
rescue StandardError => e
  puts "Error creating or tagging subnet: #{e.message}"
  return false
end

# Example usage:
def run_me
```

```
vpc_id = ""
cidr_block = ""
availability_zone = ""
tag_key = ""
tag_value = ""
region = ""
# Print usage information and then stop.
if ARGV[0] == "--help" || ARGV[0] == "-h"
  puts "Usage:  ruby ec2-ruby-example-create-subnet.rb " \
    "VPC_ID CIDR_BLOCK AVAILABILITY_ZONE TAG_KEY TAG_VALUE REGION"
  # Replace us-west-2 with the AWS Region you're using for Amazon EC2.
  puts "Example: ruby ec2-ruby-example-create-subnet.rb " \
    "vpc-6713dfEX 10.0.0.0/24 us-west-2a my-key my-value us-west-2"
  exit 1
# If no values are specified at the command prompt, use these default values.
elsif ARGV.count.zero?
  vpc_id = "vpc-6713dfEX"
  cidr_block = "10.0.0.0/24"
  availability_zone = "us-west-2a"
  tag_key = "my-key"
  tag_value = "my-value"
  # Replace us-west-2 with the AWS Region you're using for Amazon EC2.
  region = "us-west-2"
# Otherwise, use the values as specified at the command prompt.
else
  vpc_id = ARGV[0]
  cidr_block = ARGV[1]
  availability_zone = ARGV[2]
  tag_key = ARGV[3]
  tag_value = ARGV[4]
  region = ARGV[5]
end

ec2_resource = Aws::EC2::Resource.new(region: region)

if subnet_created_and_tagged?(
  ec2_resource,
  vpc_id,
  cidr_block,
  availability_zone,
  tag_key,
  tag_value
)
  puts "Subnet created and tagged."
```

```
else
  puts "Subnet not created or not tagged."
end
end

run_me if $PROGRAM_NAME == __FILE__
```

- Untuk detail API, lihat [CreateSubnet](#) di Referensi AWS SDK for Ruby API.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **CreateTags** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `CreateTags`.

C++

SDK untuk C++

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
Aws::EC2::EC2Client ec2Client(clientConfiguration);

Aws::EC2::Model::Tag nameTag;
nameTag.SetKey("Name");
nameTag.SetValue(instanceName);

Aws::EC2::Model::CreateTagsRequest createRequest;
createRequest.AddResources(instanceID);
createRequest.AddTags(nameTag);

Aws::EC2::Model::CreateTagsOutcome createOutcome = ec2Client.CreateTags(
    createRequest);
```

```
if (!createOutcome.IsSuccess()) {
    std::cerr << "Failed to tag ec2 instance " << instanceID <<
        " with name " << instanceName << ":" <<
        createOutcome.GetError().GetMessage() << std::endl;
    return false;
}
```

- Untuk detail API, lihat [CreateTags](#) di Referensi AWS SDK for C++ API.

CLI

AWS CLI

Contoh 1: Untuk menambahkan tag ke sumber daya

Contoh `create-tags` berikut menambahkan tanda `Stack=production` ke gambar tertentu, atau menimpa tanda yang sudah ada untuk AMI di mana kunci tandanya adalah `Stack`.

```
aws ec2 create-tags \
  --resources ami-1234567890abcdef0 \
  --tags Key=Stack,Value=production
```

Untuk informasi selengkapnya, lihat [Ini adalah judul topik](#) di Panduan Pengguna Amazon Elastic Compute Cloud untuk Instans Linux.

Contoh 2: Untuk menambahkan tag ke beberapa sumber daya

Contoh `create-tags` berikut menambahkan (atau menimpa) dua tanda untuk AMI dan instans. Salah satu tanda memiliki kunci (`webserver`) tetapi tidak memiliki nilai (nilai diatur ke string kosong). Tanda lainnya memiliki kunci (`stack`) dan nilai (`Production`).

```
aws ec2 create-tags \
  --resources ami-1a2b3c4d i-1234567890abcdef0 \
  --tags Key=webserver,Value=   Key=stack,Value=Production
```

Untuk informasi selengkapnya, lihat [Ini adalah judul topik](#) di Panduan Pengguna Amazon Elastic Compute Cloud untuk Instans Linux.

Contoh 3: Untuk menambahkan tag yang berisi karakter khusus

Contoh `create-tags` berikut menambahkan tanda `[Group]=test` untuk instans. Tanda kurung siku (`[` dan `]`) adalah karakter khusus dan harus di-escape. Contoh berikut juga menggunakan karakter lanjutan baris yang sesuai untuk setiap lingkungan.

Jika Anda menggunakan Windows, kurung elemen yang memiliki karakter khusus dengan petik ganda (`"`), lalu di depan setiap karakter bertanda petik ganda, tambahkan garis miring terbalik (`\`) sebagai berikut:

```
aws ec2 create-tags ^
  --resources i-1234567890abcdef0 ^
  --tags Key=\"[Group]\",Value=test
```

Jika Anda menggunakan Windows PowerShell, kelilingi elemen nilai yang memiliki karakter khusus dengan tanda kutip ganda (`"`), mendahului setiap karakter kutipan ganda dengan garis miring terbalik (`\`), dan kemudian kelilingi seluruh kunci dan struktur nilai dengan tanda kutip tunggal (`'`) sebagai berikut:

```
aws ec2 create-tags `
  --resources i-1234567890abcdef0 `
  --tags 'Key=\"[Group]\",Value=test'
```

Jika Anda menggunakan Linux atau OS X, apit elemen yang memiliki karakter khusus dengan petik ganda (`"`), lalu apit keseluruhan kunci dan struktur nilai dengan tanda petik tunggal (`'`) sebagai berikut:

```
aws ec2 create-tags \
  --resources i-1234567890abcdef0 \
  --tags 'Key="[Group]",Value=test'
```

Untuk informasi selengkapnya, lihat [Ini adalah judul topik](#) di Panduan Pengguna Amazon Elastic Compute Cloud untuk Instans Linux.

- Untuk detail API, lihat [CreateTags](#) di Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini menambahkan satu tag ke sumber daya yang ditentukan. Kunci tag adalah 'MyTag' dan nilai tag adalah 'myTagValue'. Sintaks yang digunakan oleh contoh ini membutuhkan PowerShell versi 3 atau lebih tinggi.

```
New-EC2Tag -Resource i-12345678 -Tag @{ Key="myTag"; Value="myTagValue" }
```

Contoh 2: Contoh ini memperbarui atau menambahkan tag yang ditentukan ke sumber daya yang ditentukan. Sintaks yang digunakan oleh contoh ini membutuhkan PowerShell versi 3 atau lebih tinggi.

```
New-EC2Tag -Resource i-12345678 -Tag @( @{ Key="myTag"; Value="newTagValue" },  
    @{ Key="test"; Value="anotherTagValue" } )
```

Contoh 3: Dengan PowerShell versi 2, Anda harus menggunakan New-Object untuk membuat tag untuk parameter Tag.

```
$tag = New-Object Amazon.EC2.Model.Tag  
$tag.Key = "myTag"  
$tag.Value = "myTagValue"  
  
New-EC2Tag -Resource i-12345678 -Tag $tag
```

- Untuk detail API, lihat [CreateTags](#) di Referensi AWS Tools for PowerShell Cmdlet.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **CreateVolume** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `CreateVolume`.

CLI

AWS CLI

Untuk membuat volume General Purpose SSD (gp2) kosong

`create-volume` Contoh berikut membuat volume 80 GiB General Purpose SSD (gp2) di Availability Zone yang ditentukan. Perhatikan bahwa Wilayah saat ini harus `us-east-1`, atau Anda dapat menambahkan `--region` parameter untuk menentukan Wilayah untuk perintah.

```
aws ec2 create-volume \  
  --volume-type gp2 \  
  --size 80 \  
  --availability-zone us-east-1a
```

Output:

```
{  
  "AvailabilityZone": "us-east-1a",  
  "Tags": [],  
  "Encrypted": false,  
  "VolumeType": "gp2",  
  "VolumeId": "vol-1234567890abcdef0",  
  "State": "creating",  
  "Iops": 240,  
  "SnapshotId": "",  
  "CreateTime": "YYYY-MM-DDTHH:MM:SS.000Z",  
  "Size": 80  
}
```

Jika Anda tidak menentukan jenis volume, tipe volume default adalah `gp2`.

```
aws ec2 create-volume \  
  --size 80 \  
  --availability-zone us-east-1a
```

Contoh 2: Untuk membuat volume IOPS SSD (io1) yang Disediakan dari snapshot

`create-volume` Contoh berikut membuat volume Provisioned IOPS SSD (io1) dengan 1000 IOPS yang disediakan di Availability Zone yang ditentukan menggunakan snapshot yang ditentukan.

```
aws ec2 create-volume \  
  --volume-type io1 \  
  --iops 1000 \  
  --snapshot-id snap-066877671789bd71b \  
  --availability-zone us-east-1a
```

Output:

```
{  
  "AvailabilityZone": "us-east-1a",  
  "Tags": [],  
  "Encrypted": false,  
  "VolumeType": "io1",  
  "VolumeId": "vol-1234567890abcdef0",  
  "State": "creating",  
  "Iops": 1000,  
  "SnapshotId": "snap-066877671789bd71b",  
  "CreateTime": "YYYY-MM-DDTHH:MM:SS.000Z",  
  "Size": 500  
}
```

Contoh 3: Untuk membuat volume terenkripsi

`create-volume` Contoh berikut membuat volume terenkripsi menggunakan CMK default untuk enkripsi EBS. Jika enkripsi secara default dinonaktifkan, Anda harus menentukan `--encrypted` parameter sebagai berikut.

```
aws ec2 create-volume \  
  --size 80 \  
  --encrypted \  
  --availability-zone us-east-1a
```

Output:

```
{  
  "AvailabilityZone": "us-east-1a",  
  "Tags": [],  
  "Encrypted": true,  
  "VolumeType": "gp2",  
  "VolumeId": "vol-1234567890abcdef0",  
  "State": "creating",  
}
```



```
"Iops": 240,  
"SnapshotId": "",  
"CreateTime": "YYYY-MM-DDTHH:MM:SS.000Z",  
"Size": 80  
}
```

Jika enkripsi secara default diaktifkan, perintah contoh berikut membuat volume terenkripsi, bahkan tanpa parameter. `--encrypted`

```
aws ec2 create-volume \  
  --size 80 \  
  --availability-zone us-east-1a
```

Jika Anda menggunakan `--kms-key-id` parameter untuk menentukan CMK yang dikelola pelanggan, Anda harus menentukan `--encrypted` parameter bahkan jika enkripsi secara default diaktifkan.

```
aws ec2 create-volume \  
  --volume-type gp2 \  
  --size 80 \  
  --encrypted \  
  --kms-key-id 0ea3fef3-80a7-4778-9d8c-1c0c6EXAMPLE \  
  --availability-zone us-east-1a
```

Contoh 4: Untuk membuat volume dengan tag

`create-volume` Contoh berikut menciptakan volume dan menambahkan dua tag.

```
aws ec2 create-volume \  
  --availability-zone us-east-1a \  
  --volume-type gp2 \  
  --size 80 \  
  --tag-specifications  
  'ResourceType=volume,Tags=[{Key=purpose,Value=production},{Key=cost-  
center,Value=cc123}]'
```

- Untuk detail API, lihat [CreateVolume](#) di Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini menciptakan volume yang ditentukan.

```
New-EC2Volume -Size 50 -AvailabilityZone us-west-2a -VolumeType gp2
```

Output:

```
Attachments      : {}
AvailabilityZone  : us-west-2a
CreateTime       : 12/22/2015 1:42:07 AM
Encrypted        : False
Iops             : 150
KmsKeyId         :
Size            : 50
SnapshotId       :
State            : creating
Tags             : {}
VolumeId         : vol-12345678
VolumeType       : gp2
```

Contoh 2: Permintaan contoh ini membuat volume dan menerapkan tag dengan kunci tumpukan dan nilai produksi.

```
$tag = @{ Key="stack"; Value="production" }

$tagspec = new-object Amazon.EC2.Model.TagSpecification
$tagspec.ResourceType = "volume"
$tagspec.Tags.Add($tag)

New-EC2Volume -Size 80 -AvailabilityZone "us-west-2a" -TagSpecification $tagspec
```

- Untuk detail API, lihat [CreateVolume](#) di Referensi AWS Tools for PowerShell Cmdlet.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan `CreateVpc` dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `CreateVpc`.

CLI

AWS CLI

Contoh 1: Untuk membuat VPC

Contoh `create-vpc` berikut membuat VPC dengan blok CIDR IPv4 dan tanda Nama tertentu.

```
aws ec2 create-vpc \  
  --cidr-block 10.0.0.0/16 \  
  --tag-specifications ResourceType=vpc,Tags=[{Key=Name,Value=MyVpc}]
```

Output:

```
{  
  "Vpc": {  
    "CidrBlock": "10.0.0.0/16",  
    "DhcpOptionsId": "dopt-5EXAMPLE",  
    "State": "pending",  
    "VpcId": "vpc-0a60eb65b4EXAMPLE",  
    "OwnerId": "123456789012",  
    "InstanceTenancy": "default",  
    "Ipv6CidrBlockAssociationSet": [],  
    "CidrBlockAssociationSet": [  
      {  
        "AssociationId": "vpc-cidr-assoc-07501b79ecEXAMPLE",  
        "CidrBlock": "10.0.0.0/16",  
        "CidrBlockState": {  
          "State": "associated"  
        }  
      }  
    ],  
    "IsDefault": false,  
    "Tags": [  
      {  
        "Key": "Name",  
        "Value": "MyVpc"  
      }  
    ]  
  }  
}
```

```
}  
}
```

Contoh 2: Untuk membuat VPC dengan penghunian khusus

Contoh `create-vpc` berikut membuat VPC dengan blok CIDR IPv4 dan penghunian khusus tertentu.

```
aws ec2 create-vpc \  
  --cidr-block 10.0.0.0/16 \  
  --instance-tenancy dedicated
```

Output:

```
{  
  "Vpc": {  
    "CidrBlock": "10.0.0.0/16",  
    "DhcpOptionsId": "dopt-19edf471",  
    "State": "pending",  
    "VpcId": "vpc-0a53287fa4EXAMPLE",  
    "OwnerId": "111122223333",  
    "InstanceTenancy": "dedicated",  
    "Ipv6CidrBlockAssociationSet": [],  
    "CidrBlockAssociationSet": [  
      {  
        "AssociationId": "vpc-cidr-assoc-00b24cc1c2EXAMPLE",  
        "CidrBlock": "10.0.0.0/16",  
        "CidrBlockState": {  
          "State": "associated"  
        }  
      }  
    ],  
    "IsDefault": false  
  }  
}
```

Contoh 3: Untuk membuat VPC dengan blok CIDR IPv6

Contoh `create-vpc` berikut membuat VPC dengan blok CIDR IPv6 yang disediakan Amazon.

```
aws ec2 create-vpc \  
  --instance-tenancy dedicated
```

```
--cidr-block 10.0.0.0/16 \
--amazon-provided-ipv6-cidr-block
```

Output:

```
{
  "Vpc": {
    "CidrBlock": "10.0.0.0/16",
    "DhcpOptionsId": "dopt-dEXAMPLE",
    "State": "pending",
    "VpcId": "vpc-0fc5e3406bEXAMPLE",
    "OwnerId": "123456789012",
    "InstanceTenancy": "default",
    "Ipv6CidrBlockAssociationSet": [
      {
        "AssociationId": "vpc-cidr-assoc-068432c60bEXAMPLE",
        "Ipv6CidrBlock": "",
        "Ipv6CidrBlockState": {
          "State": "associating"
        },
        "Ipv6Pool": "Amazon",
        "NetworkBorderGroup": "us-west-2"
      }
    ],
    "CidrBlockAssociationSet": [
      {
        "AssociationId": "vpc-cidr-assoc-0669f8f9f5EXAMPLE",
        "CidrBlock": "10.0.0.0/16",
        "CidrBlockState": {
          "State": "associated"
        }
      }
    ],
    "IsDefault": false
  }
}
```

Contoh 4: Untuk membuat VPC dengan CIDR dari kolom IPAM

Contoh `create-vpc` berikut membuat VPC dengan CIDR dari kolom Manajer Alamat IP (IPAM) Amazon VPC.

Linux dan macOS:

```
aws ec2 create-vpc \  
  --ipv4-ipam-pool-id ipam-pool-0533048da7d823723 \  
  --tag-specifications  
  ResourceType=vpc,Tags='[{"Key=Environment,Value="Preprod"},  
{"Key=Owner,Value="Build Team"}]'
```

Windows:

```
aws ec2 create-vpc ^  
  --ipv4-ipam-pool-id ipam-pool-0533048da7d823723 ^  
  --tag-specifications  
  ResourceType=vpc,Tags=[{"Key=Environment,Value="Preprod"}, {"Key=Owner,Value="Build  
Team"}]
```

Output:

```
{  
  "Vpc": {  
    "CidrBlock": "10.0.1.0/24",  
    "DhcpOptionsId": "dopt-2afccf50",  
    "State": "pending",  
    "VpcId": "vpc-010e1791024eb0af9",  
    "OwnerId": "123456789012",  
    "InstanceTenancy": "default",  
    "Ipv6CidrBlockAssociationSet": [],  
    "CidrBlockAssociationSet": [  
      {  
        "AssociationId": "vpc-cidr-assoc-0a77de1d803226d4b",  
        "CidrBlock": "10.0.1.0/24",  
        "CidrBlockState": {  
          "State": "associated"  
        }  
      }  
    ],  
    "IsDefault": false,  
    "Tags": [  
      {  
        "Key": "Environment",  
        "Value": "Preprod"  
      },  
      {  
        "Key": "Owner",
```

```
    "Value": "Build Team"
  }
]
}
```

Untuk informasi selengkapnya, lihat [Membuat VPC yang menggunakan CIDR kolom IPAM](#) di Panduan Pengguna IPAM Amazon VPC.

- Untuk detail API, lihat [CreateVpc](#) di Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini membuat VPC dengan CIDR yang ditentukan. Amazon VPC juga membuat yang berikut untuk VPC: set opsi DHCP default, tabel rute utama, dan ACL jaringan default.

```
New-EC2VPC -CidrBlock 10.0.0.0/16
```

Output:

```
CidrBlock      : 10.0.0.0/16
DhcpOptionsId  : dopt-1a2b3c4d
InstanceTenancy : default
IsDefault      : False
State          : pending
Tags           : {}
VpcId          : vpc-12345678
```

- Untuk detail API, lihat [CreateVpc](#) di Referensi AWS Tools for PowerShell Cmdlet.

Ruby

SDK untuk Ruby

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
require "aws-sdk-ec2"

# Creates a virtual private cloud (VPC) in
# Amazon Virtual Private Cloud (Amazon VPC) and then tags
# the VPC.
#
# @param ec2_resource [Aws::EC2::Resource] An initialized
#   Amazon Elastic Compute Cloud (Amazon EC2) resource object.
# @param cidr_block [String] The IPv4 CIDR block for the subnet.
# @param tag_key [String] The key portion of the tag for the VPC.
# @param tag_value [String] The value portion of the tag for the VPC.
# @return [Boolean] true if the VPC was created and tagged;
#   otherwise, false.
# @example
#   exit 1 unless vpc_created_and_tagged?(
#     Aws::EC2::Resource.new(region: 'us-west-2'),
#     '10.0.0.0/24',
#     'my-key',
#     'my-value'
#   )
def vpc_created_and_tagged?(
  ec2_resource,
  cidr_block,
  tag_key,
  tag_value
)
  vpc = ec2_resource.create_vpc(cidr_block: cidr_block)

  # Create a public DNS by enabling DNS support and DNS hostnames.
  vpc.modify_attribute(enable_dns_support: { value: true })
  vpc.modify_attribute(enable_dns_hostnames: { value: true })
end
```



```
vpc.create_tags(tags: [{ key: tag_key, value: tag_value }])

puts "Created VPC with ID '#{vpc.id}' and tagged with key " \
    "'#{tag_key}' and value '#{tag_value}'."
return true
rescue StandardError => e
  puts "#{e.message}"
  return false
end

# Example usage:
def run_me
  cidr_block = ""
  tag_key = ""
  tag_value = ""
  region = ""
  # Print usage information and then stop.
  if ARGV[0] == "--help" || ARGV[0] == "-h"
    puts "Usage:  ruby ec2-ruby-example-create-vpc.rb " \
        "CIDR_BLOCK TAG_KEY TAG_VALUE REGION"
    # Replace us-west-2 with the AWS Region you're using for Amazon EC2.
    puts "Example: ruby ec2-ruby-example-create-vpc.rb " \
        "10.0.0.0/24 my-key my-value us-west-2"
    exit 1
  # If no values are specified at the command prompt, use these default values.
  elsif ARGV.count.zero?
    cidr_block = "10.0.0.0/24"
    tag_key = "my-key"
    tag_value = "my-value"
    # Replace us-west-2 with the AWS Region you're using for Amazon EC2.
    region = "us-west-2"
  # Otherwise, use the values as specified at the command prompt.
  else
    cidr_block = ARGV[0]
    tag_key = ARGV[1]
    tag_value = ARGV[2]
    region = ARGV[3]
  end

  ec2_resource = Aws::EC2::Resource.new(region: region)

  if vpc_created_and_tagged?(
    ec2_resource,
```

```

    cidr_block,
    tag_key,
    tag_value
  )
  puts "VPC created and tagged."
else
  puts "VPC not created or not tagged."
end
end
end

run_me if $PROGRAM_NAME == __FILE__

```

- Untuk detail API, lihat [CreateVpc](#) di Referensi AWS SDK for Ruby API.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **CreateVpcEndpoint** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `CreateVpcEndpoint`.

CLI

AWS CLI

Contoh 1: Untuk membuat titik akhir gateway

`create-vpc-endpoint` Contoh berikut membuat titik akhir VPC gateway antara VPC dan `vpc-1a2b3c4d` Amazon S3 di wilayah tersebut, dan mengaitkan tabel rute `us-east-1` dengan titik akhir. `rtb-11aa22bb`

```

aws ec2 create-vpc-endpoint \
  --vpc-id vpc-1a2b3c4d \
  --service-name com.amazonaws.us-east-1.s3 \
  --route-table-ids rtb-11aa22bb

```

Output:

```

{
  "VpcEndpoint": {

```

```

    "PolicyDocument": "{\"Version\":\"2008-10-17\",\"Statement\":[{\"Sid\":\"\\\",\\\"Effect\\\":\\\"Allow\\\",\\\"Principal\\\":\\\"*\",\\\"Action\\\":\\\"*\",\\\"Resource\\\":\\\"*\"]}]\",
    \"VpcId\": \"vpc-1a2b3c4d\",
    \"State\": \"available\",
    \"ServiceName\": \"com.amazonaws.us-east-1.s3\",
    \"RouteTableIds\": [
      \"rtb-11aa22bb\"
    ],
    \"VpcEndpointId\": \"vpc-1a2b3c4d\",
    \"CreationTimestamp\": \"2015-05-15T09:40:50Z\"
  }
}

```

Untuk informasi selengkapnya, lihat [Membuat titik akhir gateway](#) di AWS PrivateLink Panduan.

Contoh 2: Untuk membuat titik akhir antarmuka

create-vpc-endpoint Contoh berikut membuat antarmuka VPC endpoint antara VPC dan vpc-1a2b3c4d Amazon S3 di wilayah tersebut. us-east-1 Perintah membuat titik akhir di subnet subnet-1a2b3c4d, mengaitkannya dengan grup keamanan sg-1a2b3c4d, dan menambahkan tag dengan kunci “Layanan” dan Nilai “S3”.

```

aws ec2 create-vpc-endpoint \
  --vpc-id vpc-1a2b3c4d \
  --vpc-endpoint-type Interface \
  --service-name com.amazonaws.us-east-1.s3 \
  --subnet-ids subnet-7b16de0c \
  --security-group-id sg-1a2b3c4d \
  --tag-specifications ResourceType=vpc-endpoint,Tags=[{Key=service,Value=S3}]

```

Output:

```

{
  "VpcEndpoint": {
    "VpcEndpointId": "vpce-1a2b3c4d5e6f1a2b3",
    "VpcEndpointType": "Interface",
    "VpcId": "vpc-1a2b3c4d",
    "ServiceName": "com.amazonaws.us-east-1.s3",
    "State": "pending",
    "RouteTableIds": [],
    "SubnetIds": [
      "subnet-1a2b3c4d"
    ]
  }
}

```

```

    ],
    "Groups": [
      {
        "GroupId": "sg-1a2b3c4d",
        "GroupName": "default"
      }
    ],
    "PrivateDnsEnabled": false,
    "RequesterManaged": false,
    "NetworkInterfaceIds": [
      "eni-0b16f0581c8ac6877"
    ],
    "DnsEntries": [
      {
        "DnsName": "*.vpce-1a2b3c4d5e6f1a2b3-9hnenorg.s3.us-
east-1.vpce.amazonaws.com",
        "HostedZoneId": "Z7HUB22UULQXV"
      },
      {
        "DnsName": "*.vpce-1a2b3c4d5e6f1a2b3-9hnenorg-us-east-1c.s3.us-
east-1.vpce.amazonaws.com",
        "HostedZoneId": "Z7HUB22UULQXV"
      }
    ],
    "CreationTimestamp": "2021-03-05T14:46:16.030000+00:00",
    "Tags": [
      {
        "Key": "service",
        "Value": "S3"
      }
    ],
    "OwnerId": "123456789012"
  }
}

```

Untuk informasi selengkapnya, lihat [Membuat titik akhir antarmuka](#) di Panduan Pengguna untuk AWS PrivateLink.

Contoh 3: Untuk membuat titik akhir Load Balancer Gateway

`create-vpc-endpoint` Contoh berikut membuat titik akhir Load Balancer Gateway antara VPC `vpc-111122223333aabb` dan layanan yang dikonfigurasi menggunakan Load Balancer Gateway.

```
aws ec2 create-vpc-endpoint \  
  --service-name com.amazonaws.vpce.us-east-1.vpce-svc-123123a1c43abc123 \  
  --vpc-endpoint-type GatewayLoadBalancer \  
  --vpc-id vpc-111122223333aabbcc \  
  --subnet-ids subnet-0011aabbcc2233445
```

Output:

```
{  
  "VpcEndpoint": {  
    "VpcEndpointId": "vpce-aabbaabbaabbaabba",  
    "VpcEndpointType": "GatewayLoadBalancer",  
    "VpcId": "vpc-111122223333aabbcc",  
    "ServiceName": "com.amazonaws.vpce.us-east-1.vpce-svc-123123a1c43abc123",  
    "State": "pending",  
    "SubnetIds": [  
      "subnet-0011aabbcc2233445"  
    ],  
    "RequesterManaged": false,  
    "NetworkInterfaceIds": [  
      "eni-01010120203030405"  
    ],  
    "CreationTimestamp": "2020-11-11T08:06:03.522Z",  
    "OwnerId": "123456789012"  
  }  
}
```

Untuk informasi selengkapnya, lihat [titik akhir Load Balancer Gateway](#) di Panduan Pengguna untuk AWS PrivateLink

- Untuk detail API, lihat [CreateVpcEndpoint](#) di Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini membuat Endpoint VPC baru untuk layanan com.amazonaws.eu-west-1.s3 di VPC vpc-0fc1ff23f45b678eb

```
New-EC2VpcEndpoint -ServiceName com.amazonaws.eu-west-1.s3 -VpcId  
vpc-0fc1ff23f45b678eb
```

Output:

```
ClientToken VpcEndpoint
-----
                Amazon.EC2.Model.VpcEndpoint
```

- Untuk detail API, lihat [CreateVpcEndpoint](#) di Referensi AWS Tools for PowerShell Cmdlet.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **CreateVpnConnection** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `CreateVpnConnection`.

CLI**AWS CLI**

Contoh 1: Untuk membuat koneksi VPN dengan perutean dinamis

`create-vpn-connection` Contoh berikut membuat koneksi VPN antara gateway pribadi virtual yang ditentukan dan gateway pelanggan yang ditentukan, dan menerapkan tag ke koneksi VPN. Outputnya mencakup informasi konfigurasi untuk perangkat gateway pelanggan Anda, dalam format XHTML.

```
aws ec2 create-vpn-connection \
  --type ipsec.1 \
  --customer-gateway-id cgw-001122334455aabbcc \
  --vpn-gateway-id vgw-1a1a1a1a1a1a2b2b2 \
  --tag-specification 'ResourceType=vpn-connection,Tags=[{Key=Name,Value=BGP-VPN}]'
```

Output:

```
{
  "VpnConnection": {
    "CustomerGatewayConfiguration": "...configuration information...",
    "CustomerGatewayId": "cgw-001122334455aabbcc",
```

```

    "Category": "VPN",
    "State": "pending",
    "VpnConnectionId": "vpn-123123123123abcab",
    "VpnGatewayId": "vgw-1a1a1a1a1a1a2b2b2",
    "Options": {
      "EnableAcceleration": false,
      "StaticRoutesOnly": false,
      "LocalIpv4NetworkCidr": "0.0.0.0/0",
      "RemoteIpv4NetworkCidr": "0.0.0.0/0",
      "TunnelInsideIpVersion": "ipv4",
      "TunnelOptions": [
        {},
        {}
      ]
    },
    "Routes": [],
    "Tags": [
      {
        "Key": "Name",
        "Value": "BGP-VPN"
      }
    ]
  }
}

```

Untuk informasi selengkapnya, lihat [Cara kerja AWS VPN Site-to-Site di Panduan Pengguna VPN AWS Site-to-Site](#).

Contoh 2: Untuk membuat koneksi VPN dengan perutean statis

`create-vpn-connection` Contoh berikut membuat koneksi VPN antara gateway pribadi virtual yang ditentukan dan gateway pelanggan yang ditentukan. Opsi menentukan perutean statis. Outputnya mencakup informasi konfigurasi untuk perangkat gateway pelanggan Anda, dalam format XHTML.

```

aws ec2 create-vpn-connection \
  --type ipsec.1 \
  --customer-gateway-id cgw-001122334455aabbcc \
  --vpn-gateway-id vgw-1a1a1a1a1a1a2b2b2 \
  --options "{\"StaticRoutesOnly\":true}"

```

Output:

```
{
  "VpnConnection": {
    "CustomerGatewayConfiguration": "..configuration information...",
    "CustomerGatewayId": "cgw-001122334455aabbcc",
    "Category": "VPN",
    "State": "pending",
    "VpnConnectionId": "vpn-123123123123abcab",
    "VpnGatewayId": "vgw-1a1a1a1a1a1a2b2b2",
    "Options": {
      "EnableAcceleration": false,
      "StaticRoutesOnly": true,
      "LocalIpv4NetworkCidr": "0.0.0.0/0",
      "RemoteIpv4NetworkCidr": "0.0.0.0/0",
      "TunnelInsideIpVersion": "ipv4",
      "TunnelOptions": [
        {},
        {}
      ]
    },
    "Routes": [],
    "Tags": []
  }
}
```

Untuk informasi selengkapnya, lihat [Cara kerja AWS VPN Site-to-Site di Panduan Pengguna VPN AWS Site-to-Site](#).

Contoh 3: Untuk membuat koneksi VPN dan menentukan sendiri di dalam CIDR dan kunci yang telah dibagikan sebelumnya

create-`vpn-connection` Contoh berikut membuat koneksi VPN dan menentukan blok CIDR alamat IP bagian dalam dan kunci pra-bersama khusus untuk setiap terowongan. Nilai yang ditentukan dikembalikan dalam `CustomerGatewayConfiguration` informasi.

```
aws ec2 create-vpn-connection \
  --type ipsec.1 \
  --customer-gateway-id cgw-001122334455aabbcc \
  --vpn-gateway-id vgw-1a1a1a1a1a1a2b2b2 \
  --options
  TunnelOptions='[{"TunnelInsideCidr": "169.254.12.0/30", "PreSharedKey": "ExamplePreSharedKey1"},
{"TunnelInsideCidr": "169.254.13.0/30", "PreSharedKey": "ExamplePreSharedKey2"}]'
```


Output:

```
{
  "VpnConnection": {
    "CustomerGatewayConfiguration": "..configuration information...",
    "CustomerGatewayId": "cgw-001122334455aabbcc",
    "Category": "VPN",
    "State": "pending",
    "VpnConnectionId": "vpn-123123123123abcab",
    "VpnGatewayId": "vgw-1a1a1a1a1a1a2b2b2",
    "Options": {
      "EnableAcceleration": false,
      "StaticRoutesOnly": false,
      "LocalIpv4NetworkCidr": "0.0.0.0/0",
      "RemoteIpv4NetworkCidr": "0.0.0.0/0",
      "TunnelInsideIpVersion": "ipv4",
      "TunnelOptions": [
        {
          "OutsideIpAddress": "203.0.113.3",
          "TunnelInsideCidr": "169.254.12.0/30",
          "PreSharedKey": "ExamplePreSharedKey1"
        },
        {
          "OutsideIpAddress": "203.0.113.5",
          "TunnelInsideCidr": "169.254.13.0/30",
          "PreSharedKey": "ExamplePreSharedKey2"
        }
      ]
    },
    "Routes": [],
    "Tags": []
  }
}
```

Untuk informasi selengkapnya, lihat [Cara kerja AWS VPN Site-to-Site di Panduan Pengguna VPN AWS Site-to-Site](#).

Contoh 4: Untuk membuat koneksi VPN yang mendukung lalu lintas IPv6

`create-vpn-connection` Contoh berikut membuat koneksi VPN yang mendukung lalu lintas IPv6 antara gateway transit yang ditentukan dan gateway pelanggan yang ditentukan. Opsi terowongan untuk kedua terowongan menentukan yang AWS harus memulai negosiasi IKE.

```
aws ec2 create-vpn-connection \  
  --type ipsec.1 \  
  --transit-gateway-id tgw-12312312312312312 \  
  --customer-gateway-id cgw-001122334455aabbc \  
  --options TunnelInsideIpVersion=ipv6,TunnelOptions=[{StartupAction=start},  
{StartupAction=start}]
```

Output:

```
{  
  "VpnConnection": {  
    "CustomerGatewayConfiguration": "..configuration information...",  
    "CustomerGatewayId": "cgw-001122334455aabbc",  
    "Category": "VPN",  
    "State": "pending",  
    "VpnConnectionId": "vpn-11111111122222222",  
    "TransitGatewayId": "tgw-12312312312312312",  
    "Options": {  
      "EnableAcceleration": false,  
      "StaticRoutesOnly": false,  
      "LocalIpv6NetworkCidr": "::/0",  
      "RemoteIpv6NetworkCidr": "::/0",  
      "TunnelInsideIpVersion": "ipv6",  
      "TunnelOptions": [  
        {  
          "OutsideIpAddress": "203.0.113.3",  
          "StartupAction": "start"  
        },  
        {  
          "OutsideIpAddress": "203.0.113.5",  
          "StartupAction": "start"  
        }  
      ]  
    },  
    "Routes": [],  
    "Tags": []  
  }  
}
```

Untuk informasi selengkapnya, lihat [Cara kerja AWS VPN Site-to-Site di Panduan Pengguna VPN AWS Site-to-Site](#).

- Untuk detail API, lihat [CreateVpnConnection](#) di Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini membuat koneksi VPN antara gateway pribadi virtual yang ditentukan dan gateway pelanggan yang ditentukan. Outputnya mencakup informasi konfigurasi yang dibutuhkan administrator jaringan Anda, dalam format XHTML.

```
New-EC2VpnConnection -Type ipsec.1 -CustomerGatewayId cgw-1a2b3c4d -VpnGatewayId vgw-1a2b3c4d
```

Output:

```
CustomerGatewayConfiguration : [XML document]
CustomerGatewayId            : cgw-1a2b3c4d
Options                       :
Routes                       : {}
State                        : pending
Tags                         : {}
Type                         :
VgwTelemetry                 : {}
VpnConnectionId             : vpn-12345678
VpnGatewayId                : vgw-1a2b3c4d
```

Contoh 2: Contoh ini membuat koneksi VPN dan menangkap konfigurasi dalam file dengan nama yang ditentukan.

```
(New-EC2VpnConnection -CustomerGatewayId cgw-1a2b3c4d -VpnGatewayId vgw-1a2b3c4d).CustomerGatewayConfiguration | Out-File C:\path\vpn-configuration.xml
```

Contoh 3: Contoh ini membuat koneksi VPN, dengan perutean statis, antara gateway pribadi virtual yang ditentukan dan gateway pelanggan yang ditentukan.

```
New-EC2VpnConnection -Type ipsec.1 -CustomerGatewayId cgw-1a2b3c4d -VpnGatewayId vgw-1a2b3c4d -Options_StaticRoutesOnly $true
```

- Untuk detail API, lihat [CreateVpnConnection](#) di Referensi AWS Tools for PowerShell Cmdlet.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan `CreateVpnConnectionRoute` dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `CreateVpnConnectionRoute`.

CLI

AWS CLI

Untuk membuat rute statis untuk koneksi VPN

Contoh ini membuat rute statis untuk koneksi VPN yang ditentukan. Jika perintah berhasil, tidak ada output yang akan ditampilkan.

Perintah:

```
aws ec2 create-vpn-connection-route --vpn-connection-id vpn-40f41529 --
destination-cidr-block 11.12.0.0/16
```

- Untuk detail API, lihat [CreateVpnConnectionRoute](#) di Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini membuat rute statis yang ditentukan untuk koneksi VPN yang ditentukan.

```
New-EC2VpnConnectionRoute -VpnConnectionId vpn-12345678 -DestinationCidrBlock
11.12.0.0/16
```

- Untuk detail API, lihat [CreateVpnConnectionRoute](#) di Referensi AWS Tools for PowerShell Cmdlet.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan `CreateVpnGateway` dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `CreateVpnGateway`.

CLI

AWS CLI

Untuk membuat gateway pribadi virtual

Contoh ini membuat gateway pribadi virtual.

Perintah:

```
aws ec2 create-vpn-gateway --type ipsec.1
```

Output:

```
{
  "VpnGateway": {
    "AmazonSideAsn": 64512,
    "State": "available",
    "Type": "ipsec.1",
    "VpnGatewayId": "vgw-9a4cacf3",
    "VpcAttachments": []
  }
}
```

Untuk membuat gateway pribadi virtual dengan ASN sisi Amazon tertentu

Contoh ini membuat gateway pribadi virtual dan menentukan Autonomous System Number (ASN) untuk sisi Amazon dari sesi BGP.

Perintah:

```
aws ec2 create-vpn-gateway --type ipsec.1 --amazon-side-asn 65001
```

Output:

```
{
  "VpnGateway": {
    "AmazonSideAsn": 65001,
```

```
    "State": "available",
    "Type": "ipsec.1",
    "VpnGatewayId": "vgw-9a4cacf3",
    "VpcAttachments": []
  }
}
```

- Untuk detail API, lihat [CreateVpnGateway](#) di Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini membuat gateway pribadi virtual yang ditentukan.

```
New-EC2VpnGateway -Type ipsec.1
```

Output:

```
AvailabilityZone :
State            : available
Tags            : {}
Type            : ipsec.1
VpcAttachments  : {}
VpnGatewayId    : vgw-1a2b3c4d
```

- Untuk detail API, lihat [CreateVpnGateway](#) di Referensi AWS Tools for PowerShell Cmdlet.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **DeleteCustomerGateway** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `DeleteCustomerGateway`.

CLI

AWS CLI

Untuk menghapus gateway pelanggan

Contoh ini menghapus gateway pelanggan yang ditentukan. Jika perintah berhasil, tidak ada output yang akan ditampilkan.

Perintah:

```
aws ec2 delete-customer-gateway --customer-gateway-id cgw-0e11f167
```

- Untuk detail API, lihat [DeleteCustomerGateway](#) di Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini menghapus gateway pelanggan yang ditentukan. Anda diminta untuk konfirmasi sebelum operasi berlangsung, kecuali Anda juga menentukan parameter Force.

```
Remove-EC2CustomerGateway -CustomerGatewayId cgw-1a2b3c4d
```

Output:

```
Confirm
Are you sure you want to perform this action?
Performing operation "Remove-EC2CustomerGateway (DeleteCustomerGateway)" on
Target "cgw-1a2b3c4d".
[Y] Yes [A] Yes to All [N] No [L] No to All [S] Suspend [?] Help (default is
"Y"):
```

- Untuk detail API, lihat [DeleteCustomerGateway](#) di Referensi AWS Tools for PowerShell Cmdlet.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **DeleteDhcpOptions** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `DeleteDhcpOptions`.

CLI

AWS CLI

Untuk menghapus set opsi DHCP

Contoh ini menghapus set opsi DHCP yang ditentukan. Jika perintah berhasil, tidak ada output yang akan ditampilkan.

Perintah:

```
aws ec2 delete-dhcp-options --dhcp-options-id dopt-d9070ebb
```

- Untuk detail API, lihat [DeleteDhcpOptions](#) di Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini menghapus set opsi DHCP yang ditentukan. Anda diminta untuk konfirmasi sebelum operasi berlangsung, kecuali Anda juga menentukan parameter Force.

```
Remove-EC2DhcpOption -DhcpOptionsId dopt-1a2b3c4d
```

Output:

```
Confirm
Are you sure you want to perform this action?
Performing operation "Remove-EC2DhcpOption (DeleteDhcpOptions)" on Target
"dopt-1a2b3c4d".
[Y] Yes [A] Yes to All [N] No [L] No to All [S] Suspend [?] Help (default is
"Y"):
```

- Untuk detail API, lihat [DeleteDhcpOptions](#) di Referensi AWS Tools for PowerShell Cmdlet.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **DeleteFlowLogs** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `DeleteFlowLogs`.

CLI

AWS CLI

Untuk menghapus log aliran

`delete-flow-logs` Contoh berikut menghapus log aliran yang ditentukan.

```
aws ec2 delete-flow-logs --flow-log-id fl-11223344556677889
```

Output:

```
{
  "Unsuccessful": []
}
```

- Untuk detail API, lihat [DeleteFlowLogs](#) di Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini menghapus `FlowLogId fl-01a2b3456a789c01` yang diberikan

```
Remove-EC2FlowLog -FlowLogId fl-01a2b3456a789c01
```

Output:

```
Confirm
Are you sure you want to perform this action?
Performing the operation "Remove-EC2FlowLog (DeleteFlowLogs)" on target
"fl-01a2b3456a789c01".
[Y] Yes [A] Yes to All [N] No [L] No to All [S] Suspend [?] Help (default is
"Y"): Y
```

- Untuk detail API, lihat [DeleteFlowLogs](#) di Referensi AWS Tools for PowerShell Cmdlet.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan `DeleteInternetGateway` dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `DeleteInternetGateway`.

CLI

AWS CLI

Untuk menghapus gateway internet

`delete-internet-gateway` Contoh berikut menghapus gateway internet yang ditentukan.

```
aws ec2 delete-internet-gateway \  
  --internet-gateway-id igw-0d0fb496b3EXAMPLE
```

Perintah ini tidak menghasilkan output.

Untuk informasi lebih lanjut, lihat [Gateway internet](#) di Panduan Pengguna Amazon VPC.

- Untuk detail API, lihat [DeleteInternetGateway](#) di Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini menghapus gateway Internet yang ditentukan. Anda diminta untuk konfirmasi sebelum operasi berlangsung, kecuali Anda juga menentukan parameter `Force`.

```
Remove-EC2InternetGateway -InternetGatewayId igw-1a2b3c4d
```

Output:

```
Confirm  
Are you sure you want to perform this action?  
Performing operation "Remove-EC2InternetGateway (DeleteInternetGateway)" on  
Target "igw-1a2b3c4d".  
[Y] Yes [A] Yes to All [N] No [L] No to All [S] Suspend [?] Help (default is  
"Y"):
```

- Untuk detail API, lihat [DeleteInternetGateway](#) di Referensi AWS Tools for PowerShell Cmdlet.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **DeleteKeyPair** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `DeleteKeyPair`.

Contoh tindakan adalah kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Anda dapat melihat tindakan ini dalam konteks dalam contoh kode berikut:

- [Memulai instans](#)

.NET

AWS SDK for .NET

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
/// <summary>
/// Delete an Amazon EC2 key pair.
/// </summary>
/// <param name="keyPairName">The name of the key pair to delete.</param>
/// <returns>A Boolean value indicating the success of the action.</returns>
public async Task<bool> DeleteKeyPair(string keyPairName)
{
    try
    {
        await _amazonEC2.DeleteKeyPairAsync(new
DeleteKeyPairRequest(keyPairName)).ConfigureAwait(false);
        return true;
    }
    catch (Exception ex)
```

```

        {
            Console.WriteLine($"Couldn't delete the key pair because:
{ex.Message}");
            return false;
        }
    }

    /// <summary>
    /// Delete the temporary file where the key pair information was saved.
    /// </summary>
    /// <param name="tempFileName">The path to the temporary file.</param>
    public void DeleteTempFile(string tempFileName)
    {
        if (File.Exists(tempFileName))
        {
            File.Delete(tempFileName);
        }
    }
}

```

- Untuk detail API, lihat [DeleteKeyPair](#) di Referensi AWS SDK for .NET API.

Bash

AWS CLI dengan skrip Bash

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```

#####
# function ec2_delete_keypair
#
# This function deletes an Amazon EC2 ED25519 or 2048-bit RSA key pair.
#
# Parameters:
#     -n key_pair_name - A key pair name.
#
# And:
#     0 - If successful.

```

```

#      1 - If it fails.
#####
function ec2_delete_keypair() {
    local key_pair_name response

    local option OPTARG # Required to use getopt command in a function.
    # bashsupport disable=BP5008
    function usage() {
        echo "function ec2_delete_keypair"
        echo "Deletes an Amazon EC2 ED25519 or 2048-bit RSA key pair."
        echo "  -n key_pair_name - A key pair name."
        echo ""
    }

    # Retrieve the calling parameters.
    while getopt "n:h" option; do
        case "${option}" in
            n) key_pair_name="${OPTARG}" ;;
            h)
                usage
                return 0
                ;;
            \?)
                echo "Invalid parameter"
                usage
                return 1
                ;;
        esac
    done
    export OPTIND=1

    if [[ -z "$key_pair_name" ]]; then
        errecho "ERROR: You must provide a key pair name with the -n parameter."
        usage
        return 1
    fi

    response=$(aws ec2 delete-key-pair \
        --key-name "$key_pair_name") || {
        aws_cli_error_log ${?}
        errecho "ERROR: AWS reports delete-key-pair operation failed.$response"
        return 1
    }
}

```

```

return 0
}

```

Fungsi utilitas yang digunakan dalam contoh ini.

```

#####
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
#####
function errecho() {
    printf "%s\n" "$*" 1>&2
}

#####
# function aws_cli_error_log()
#
# This function is used to log the error messages from the AWS CLI.
#
# The function expects the following argument:
#     $1 - The error code returned by the AWS CLI.
#
# Returns:
#     0: - Success.
#
#####
function aws_cli_error_log() {
    local err_code=$1
    errecho "Error code : $err_code"
    if [ "$err_code" == 1 ]; then
        errecho " One or more S3 transfers failed."
    elif [ "$err_code" == 2 ]; then
        errecho " Command line failed to parse."
    elif [ "$err_code" == 130 ]; then
        errecho " Process received SIGINT."
    elif [ "$err_code" == 252 ]; then
        errecho " Command syntax invalid."
    elif [ "$err_code" == 253 ]; then
        errecho " The system environment or configuration was invalid."
    elif [ "$err_code" == 254 ]; then
        errecho " The service returned an error."
    elif [ "$err_code" == 255 ]; then

```

```
    errecho " 255 is a catch-all error."
fi

return 0
}
```

- Untuk detail API, lihat [DeleteKeyPair](#) di Referensi AWS CLI Perintah.

C++

SDK untuk C++

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
Aws::EC2::EC2Client ec2Client(clientConfiguration);
Aws::EC2::Model::DeleteKeyPairRequest request;

request.SetKeyName(keyPairName);
const Aws::EC2::Model::DeleteKeyPairOutcome outcome =
ec2Client.DeleteKeyPair(
    request);

if (!outcome.IsSuccess()) {
    std::cerr << "Failed to delete key pair " << keyPairName <<
        ":" << outcome.GetError().GetMessage() << std::endl;
}
else {
    std::cout << "Successfully deleted key pair named " << keyPairName <<
        std::endl;
}
}
```

- Untuk detail API, lihat [DeleteKeyPair](#) di Referensi AWS SDK for C++ API.

CLI

AWS CLI

Untuk menghapus pasangan kunci

`delete-key-pair` Contoh berikut menghapus key pair yang ditentukan.

```
aws ec2 delete-key-pair \  
  --key-name my-key-pair
```

Output:

```
{  
  "Return": true,  
  "KeyPairId": "key-03c8d3aceb53b507"  
}
```

Untuk informasi selengkapnya, lihat [Membuat dan menghapus pasangan kunci](#) di Panduan Pengguna Antarmuka Baris AWS Perintah.

- Untuk detail API, lihat [DeleteKeyPair](#) di Referensi AWS CLI Perintah.

Java

SDK untuk Java 2.x

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
public static void deleteKeys(Ec2Client ec2, String keyPair) {  
    try {  
        DeleteKeyPairRequest request = DeleteKeyPairRequest.builder()  
            .keyName(keyPair)  
            .build();  
  
        ec2.deleteKeyPair(request);  
    }  
}
```



```
        System.out.println("Successfully deleted key pair named " + keyPair);
    } catch (Ec2Exception e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
}
```

- Untuk detail API, lihat [DeleteKeyPair](#) di Referensi AWS SDK for Java 2.x API.

JavaScript

SDK untuk JavaScript (v3)

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
import { DeleteKeyPairCommand } from "@aws-sdk/client-ec2";

import { client } from "../libs/client.js";

export const main = async () => {
    const command = new DeleteKeyPairCommand({
        KeyName: "KEY_PAIR_NAME",
    });

    try {
        await client.send(command);
        console.log("Successfully deleted key pair.");
    } catch (err) {
        console.error(err);
    }
};
```

- Untuk detail API, lihat [DeleteKeyPair](#) di Referensi AWS SDK for JavaScript API.

Kotlin

SDK untuk Kotlin

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
suspend fun deleteKeys(keyPair: String?) {
    val request =
        DeleteKeyPairRequest {
            keyName = keyPair
        }

    Ec2Client { region = "us-west-2" }.use { ec2 ->
        ec2.deleteKeyPair(request)
        println("Successfully deleted key pair named $keyPair")
    }
}
```

- Untuk detail API, lihat [DeleteKeyPair](#) di AWS SDK untuk referensi API Kotlin.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini menghapus key pair yang ditentukan. Anda diminta untuk konfirmasi sebelum operasi berlangsung, kecuali Anda juga menentukan parameter Force.

```
Remove-EC2KeyPair -KeyName my-key-pair
```

Output:

```
Confirm
Are you sure you want to perform this action?
Performing operation "Remove-EC2KeyPair (DeleteKeyPair)" on Target "my-key-pair".
```

```
[Y] Yes [A] Yes to All [N] No [L] No to All [S] Suspend [?] Help (default is "Y"):
```

- Untuk detail API, lihat [DeleteKeyPair](#) di Referensi AWS Tools for PowerShell Cmdlet.

Python

SDK untuk Python (Boto3)

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
class KeyPairWrapper:
    """Encapsulates Amazon Elastic Compute Cloud (Amazon EC2) key pair
    actions."""

    def __init__(self, ec2_resource, key_file_dir, key_pair=None):
        """
        :param ec2_resource: A Boto3 Amazon EC2 resource. This high-level
        resource
                               is used to create additional high-level objects
                               that wrap low-level Amazon EC2 service actions.
        :param key_file_dir: The folder where the private key information is
        stored.
                               This should be a secure folder.
        :param key_pair: A Boto3 KeyPair object. This is a high-level object that
        wraps key pair actions.
        """
        self.ec2_resource = ec2_resource
        self.key_pair = key_pair
        self.key_file_path = None
        self.key_file_dir = key_file_dir

    @classmethod
    def from_resource(cls):
        ec2_resource = boto3.resource("ec2")
        return cls(ec2_resource, tempfile.TemporaryDirectory())
```

```
def delete(self):
    """
    Deletes a key pair.
    """
    if self.key_pair is None:
        logger.info("No key pair to delete.")
        return

    key_name = self.key_pair.name
    try:
        self.key_pair.delete()
        self.key_pair = None
    except ClientError as err:
        logger.error(
            "Couldn't delete key %s. Here's why: %s : %s",
            key_name,
            err.response["Error"]["Code"],
            err.response["Error"]["Message"],
        )
        raise
```

- Untuk detail API, lihat [DeleteKeyPair](#) di AWS SDK for Python (Boto3) Referensi API.

SAP ABAP

SDK untuk SAP ABAP

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
TRY.
  lo_ec2->deletekeypair( iv_keyname = iv_key_name ).
  MESSAGE 'Amazon EC2 key pair deleted.' TYPE 'I'.
CATCH /aws1/cx_rt_service_generic INTO DATA(lo_exception).
  DATA(lv_error) = |"{ lo_exception->av_err_code }" - { lo_exception-
>av_err_msg }|.
  MESSAGE lv_error TYPE 'E'.
```

```
ENDTRY.
```

- Untuk detail API, lihat [DeleteKeyPair](#) di AWS SDK untuk referensi SAP ABAP API.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan `DeleteLaunchTemplate` dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `DeleteLaunchTemplate`.

Contoh tindakan adalah kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Anda dapat melihat tindakan ini dalam konteks dalam contoh kode berikut:

- [Membangun dan mengelola layanan yang tangguh](#)

.NET

AWS SDK for .NET

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
/// <summary>
/// Delete a launch template by name.
/// </summary>
/// <param name="templateName">The name of the template to delete.</param>
/// <returns>Async task.</returns>
public async Task DeleteTemplateByName(string templateName)
{
    try
    {
        await _amazonEc2.DeleteLaunchTemplateAsync(
            new DeleteLaunchTemplateRequest()
```

```
        {
            LaunchTemplateName = templateName
        });
    }
    catch (AmazonClientException)
    {
        Console.WriteLine($"Unable to delete template {templateName}.");
    }
}
```

- Untuk detail API, lihat [DeleteLaunchTemplate](#) di Referensi AWS SDK for .NET API.

CLI

AWS CLI

Untuk menghapus templat peluncuran

Contoh ini menghapus templat peluncuran yang ditentukan.

Perintah:

```
aws ec2 delete-launch-template --launch-template-id lt-0abcd290751193123
```

Output:

```
{
  "LaunchTemplate": {
    "LatestVersionNumber": 2,
    "LaunchTemplateId": "lt-0abcd290751193123",
    "LaunchTemplateName": "TestTemplate",
    "DefaultVersionNumber": 2,
    "CreatedBy": "arn:aws:iam::123456789012:root",
    "CreateTime": "2017-11-23T16:46:25.000Z"
  }
}
```

- Untuk detail API, lihat [DeleteLaunchTemplate](#) di Referensi AWS CLI Perintah.

JavaScript

SDK untuk JavaScript (v3)

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
await client.send(  
  new DeleteLaunchTemplateCommand({  
    LaunchTemplateName: NAMES.launchTemplateName,  
  }),  
);
```

- Untuk detail API, lihat [DeleteLaunchTemplate](#) di Referensi AWS SDK for JavaScript API.

Python

SDK untuk Python (Boto3)

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
class AutoScaler:  
    """  
    Encapsulates Amazon EC2 Auto Scaling and EC2 management actions.  
    """  
  
    def __init__(  
        self,  
        resource_prefix,  
        inst_type,  
        ami_param,  
        autoscaling_client,
```

```

        ec2_client,
        ssm_client,
        iam_client,
    ):
        """
        :param resource_prefix: The prefix for naming AWS resources that are
        created by this class.
        :param inst_type: The type of EC2 instance to create, such as t3.micro.
        :param ami_param: The Systems Manager parameter used to look up the AMI
        that is
                created.
        :param autoscaling_client: A Boto3 EC2 Auto Scaling client.
        :param ec2_client: A Boto3 EC2 client.
        :param ssm_client: A Boto3 Systems Manager client.
        :param iam_client: A Boto3 IAM client.
        """
        self.inst_type = inst_type
        self.ami_param = ami_param
        self.autoscaling_client = autoscaling_client
        self.ec2_client = ec2_client
        self.ssm_client = ssm_client
        self.iam_client = iam_client
        self.launch_template_name = f"{resource_prefix}-template"
        self.group_name = f"{resource_prefix}-group"
        self.instance_policy_name = f"{resource_prefix}-pol"
        self.instance_role_name = f"{resource_prefix}-role"
        self.instance_profile_name = f"{resource_prefix}-prof"
        self.bad_creds_policy_name = f"{resource_prefix}-bc-pol"
        self.bad_creds_role_name = f"{resource_prefix}-bc-role"
        self.bad_creds_profile_name = f"{resource_prefix}-bc-prof"
        self.key_pair_name = f"{resource_prefix}-key-pair"

    def delete_template(self):
        """
        Deletes a launch template.
        """
        try:
            self.ec2_client.delete_launch_template(
                LaunchTemplateName=self.launch_template_name
            )
            self.delete_instance_profile(
                self.instance_profile_name, self.instance_role_name
            )

```



```
        log.info("Launch template %s deleted.", self.launch_template_name)
    except ClientError as err:
        if (
            err.response["Error"]["Code"]
            == "InvalidLaunchTemplateName.NotFoundException"
        ):
            log.info(
                "Launch template %s does not exist, nothing to do.",
                self.launch_template_name,
            )
        else:
            raise AutoScalerError(
                f"Couldn't delete launch template
                {self.launch_template_name}: {err}."
            )
```

- Untuk detail API, lihat [DeleteLaunchTemplate](#) di AWS SDK for Python (Boto3) Referensi API.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **DeleteNetworkAcl** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `DeleteNetworkAcl`.

CLI

AWS CLI

Untuk menghapus jaringan ACL

Contoh ini menghapus ACL jaringan tertentu. Jika perintah berhasil, tidak ada output yang akan ditampilkan.

Perintah:

```
aws ec2 delete-network-acl --network-acl-id acl-5fb85d36
```

- Untuk detail API, lihat [DeleteNetworkAcl](#) di Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini menghapus ACL jaringan tertentu. Anda diminta untuk konfirmasi sebelum operasi berlangsung, kecuali Anda juga menentukan parameter Force.

```
Remove-EC2NetworkAcl -NetworkAclId acl-12345678
```

Output:

```
Confirm
Are you sure you want to perform this action?
Performing operation "Remove-EC2NetworkAcl (DeleteNetworkAcl)" on Target
"acl-12345678".
[Y] Yes [A] Yes to All [N] No [L] No to All [S] Suspend [?] Help (default is
"Y"):
```

- Untuk detail API, lihat [DeleteNetworkAcl](#) di Referensi AWS Tools for PowerShell Cmdlet.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan `DeleteNetworkAclEntry` dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `DeleteNetworkAclEntry`.

CLI

AWS CLI

Untuk menghapus entri ACL jaringan

Contoh ini menghapus aturan ingress nomor 100 dari ACL jaringan yang ditentukan. Jika perintah berhasil, tidak ada output yang akan ditampilkan.

Perintah:

```
aws ec2 delete-network-acl-entry --network-acl-id acl-5fb85d36 --ingress --rule-
number 100
```

- Untuk detail API, lihat [DeleteNetworkAclEntry](#) di Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini menghapus aturan yang ditentukan dari ACL jaringan tertentu. Anda diminta untuk konfirmasi sebelum operasi berlangsung, kecuali Anda juga menentukan parameter Force.

```
Remove-EC2NetworkAclEntry -NetworkAclId acl-12345678 -Egress $false -RuleNumber 100
```

Output:

```
Confirm
Are you sure you want to perform this action?
Performing operation "Remove-EC2NetworkAclEntry (DeleteNetworkAclEntry)" on
Target "acl-12345678".
[Y] Yes [A] Yes to All [N] No [L] No to All [S] Suspend [?] Help (default is
"Y"):
```

- Untuk detail API, lihat [DeleteNetworkAclEntry](#) di Referensi AWS Tools for PowerShell Cmdlet.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **DeleteNetworkInterface** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `DeleteNetworkInterface`.

CLI

AWS CLI

Untuk menghapus antarmuka jaringan

Contoh ini menghapus antarmuka jaringan yang ditentukan. Jika perintah berhasil, tidak ada output yang akan ditampilkan.

Perintah:

```
aws ec2 delete-network-interface --network-interface-id eni-e5aa89a3
```

- Untuk detail API, lihat [DeleteNetworkInterface](#) di Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini menghapus antarmuka jaringan yang ditentukan. Anda diminta untuk konfirmasi sebelum operasi berlangsung, kecuali Anda juga menentukan parameter Force.

```
Remove-EC2NetworkInterface -NetworkInterfaceId eni-12345678
```

Output:

```
Confirm
Are you sure you want to perform this action?
Performing operation "Remove-EC2NetworkInterface (DeleteNetworkInterface)" on
Target "eni-12345678".
[Y] Yes [A] Yes to All [N] No [L] No to All [S] Suspend [?] Help (default is
"Y"):
```

- Untuk detail API, lihat [DeleteNetworkInterface](#) di Referensi AWS Tools for PowerShell Cmdlet.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **DeletePlacementGroup** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `DeletePlacementGroup`.

CLI

AWS CLI

Untuk menghapus grup penempatan

Perintah contoh ini menghapus grup penempatan yang ditentukan.

Perintah:

```
aws ec2 delete-placement-group --group-name my-cluster
```

- Untuk detail API, lihat [DeletePlacementGroup](#) di Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini menghapus grup penempatan yang ditentukan. Anda diminta untuk konfirmasi sebelum operasi berlangsung, kecuali jika Anda juga menentukan parameter Force.

```
Remove-EC2PlacementGroup -GroupName my-placement-group
```

Output:

```
Confirm
Are you sure you want to perform this action?
Performing operation "Remove-EC2PlacementGroup (DeletePlacementGroup)" on Target
"my-placement-group".
[Y] Yes [A] Yes to All [N] No [L] No to All [S] Suspend [?] Help (default is
"Y"):
```

- Untuk detail API, lihat [DeletePlacementGroup](#) di Referensi AWS Tools for PowerShell Cmdlet.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **DeleteRoute** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `DeleteRoute`.

CLI

AWS CLI

Untuk menghapus rute

Contoh ini menghapus rute yang ditentukan dari tabel rute yang ditentukan. Jika perintah berhasil, tidak ada output yang akan ditampilkan.

Perintah:

```
aws ec2 delete-route --route-table-id rtb-22574640 --destination-cidr-block 0.0.0.0/0
```

- Untuk detail API, lihat [DeleteRoute](#) di Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini menghapus rute yang ditentukan dari tabel rute yang ditentukan. Anda diminta untuk konfirmasi sebelum operasi berlangsung, kecuali jika Anda juga menentukan parameter `Force`.

```
Remove-EC2Route -RouteTableId rtb-1a2b3c4d -DestinationCidrBlock 0.0.0.0/0
```

Output:

```
Confirm
Are you sure you want to perform this action?
Performing operation "Remove-EC2Route (DeleteRoute)" on Target "rtb-1a2b3c4d".
[Y] Yes [A] Yes to All [N] No [L] No to All [S] Suspend [?] Help (default is "Y"):
```

- Untuk detail API, lihat [DeleteRoute](#) di Referensi AWS Tools for PowerShell Cmdlet.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan `DeleteRouteTable` dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `DeleteRouteTable`.

CLI

AWS CLI

Untuk menghapus tabel rute

Contoh ini menghapus tabel rute yang ditentukan. Jika perintah berhasil, tidak ada output yang akan ditampilkan.

Perintah:

```
aws ec2 delete-route-table --route-table-id rtb-22574640
```

- Untuk detail API, lihat [DeleteRouteTable](#) di Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini menghapus tabel rute yang ditentukan. Anda diminta untuk konfirmasi sebelum operasi berlangsung, kecuali jika Anda juga menentukan parameter `Force`.

```
Remove-EC2RouteTable -RouteTableId rtb-1a2b3c4d
```

Output:

```
Confirm
Are you sure you want to perform this action?
Performing operation "Remove-EC2RouteTable (DeleteRouteTable)" on Target
"rtb-1a2b3c4d".
[Y] Yes  [A] Yes to All  [N] No  [L] No to All  [S] Suspend  [?] Help (default is
"Y"):
```

- Untuk detail API, lihat [DeleteRouteTable](#) di Referensi AWS Tools for PowerShell Cmdlet.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **DeleteSecurityGroup** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `DeleteSecurityGroup`.

Contoh tindakan adalah kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Anda dapat melihat tindakan ini dalam konteks dalam contoh kode berikut:

- [Memulai instans](#)

.NET

AWS SDK for .NET

Note


Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
/// <summary>
/// Delete an Amazon EC2 security group.
/// </summary>
/// <param name="groupName">The name of the group to delete.</param>
/// <returns>A Boolean value indicating the success of the action.</returns>
public async Task<bool> DeleteSecurityGroup(string groupId)
{
    var response = await _amazonEC2.DeleteSecurityGroupAsync(new
DeleteSecurityGroupRequest { GroupId = groupId });
    return response.HttpStatusCode == HttpStatusCode.OK;
}
```

- Untuk detail API, lihat [DeleteSecurityGroup](#) di Referensi AWS SDK for .NET API.

Bash

AWS CLI dengan skrip Bash

 Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
#####
# function ec2_delete_security_group
#
# This function deletes an Amazon Elastic Compute Cloud (Amazon EC2) security
# group.
#
# Parameters:
#     -i security_group_id - The ID of the security group to delete.
#
# And:
#     0 - If successful.
#     1 - If it fails.
#####
function ec2_delete_security_group() {
    local security_group_id response
    local option OPTARG # Required to use getopt command in a function.

    # bashsupport disable=BP5008
    function usage() {
        echo "function ec2_delete_security_group"
        echo "Deletes an Amazon Elastic Compute Cloud (Amazon EC2) security group."
        echo "  -i security_group_id - The ID of the security group to delete."
        echo ""
    }

    # Retrieve the calling parameters.
    while getopt "i:h" option; do
        case "${option}" in
            i) security_group_id="${OPTARG}" ;;
            h)
                usage
                return 0
        esac
    done
}
```

```

        ;;
    \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
done
export OPTIND=1

if [[ -z "$security_group_id" ]]; then
    errecho "ERROR: You must provide a security group ID with the -i parameter."
    usage
    return 1
fi

response=$(aws ec2 delete-security-group --group-id "$security_group_id" --
output text) || {
    aws_cli_error_log ${?}
    errecho "ERROR: AWS reports delete-security-group operation failed.$response"
    return 1
}

return 0
}

```

Fungsi utilitas yang digunakan dalam contoh ini.

```

#####
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
#####
function errecho() {
    printf "%s\n" "$*" 1>&2
}

#####
# function aws_cli_error_log()
#
# This function is used to log the error messages from the AWS CLI.
#

```

```

# The function expects the following argument:
#     $1 - The error code returned by the AWS CLI.
#
# Returns:
#     0: - Success.
#
#####
function aws_cli_error_log() {
    local err_code=$1
    errecho "Error code : $err_code"
    if [ "$err_code" == 1 ]; then
        errecho " One or more S3 transfers failed."
    elif [ "$err_code" == 2 ]; then
        errecho " Command line failed to parse."
    elif [ "$err_code" == 130 ]; then
        errecho " Process received SIGINT."
    elif [ "$err_code" == 252 ]; then
        errecho " Command syntax invalid."
    elif [ "$err_code" == 253 ]; then
        errecho " The system environment or configuration was invalid."
    elif [ "$err_code" == 254 ]; then
        errecho " The service returned an error."
    elif [ "$err_code" == 255 ]; then
        errecho " 255 is a catch-all error."
    fi

    return 0
}

```

- Untuk detail API, lihat [DeleteSecurityGroup](#) di Referensi AWS CLI Perintah.

C++

SDK untuk C++

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
Aws::EC2::EC2Client ec2Client(clientConfiguration);
Aws::EC2::Model::DeleteSecurityGroupRequest request;

request.SetGroupId(securityGroupID);
auto outcome = ec2Client.DeleteSecurityGroup(request);

if (!outcome.IsSuccess()) {
    std::cerr << "Failed to delete security group " << securityGroupID <<
        ":" << outcome.GetError().GetMessage() << std::endl;
}
else {
    std::cout << "Successfully deleted security group " << securityGroupID <<
        std::endl;
}
```

- Untuk detail API, lihat [DeleteSecurityGroup](#) di Referensi AWS SDK for C++ API.

CLI

AWS CLI

[EC2-Classic] Untuk menghapus grup keamanan

Contoh ini menghapus grup keamanan bernama MySecurityGroup. Jika perintah berhasil, tidak ada output yang akan ditampilkan.

Perintah:

```
aws ec2 delete-security-group --group-name MySecurityGroup
```

[EC2-VPC] Untuk menghapus grup keamanan

Contoh ini menghapus grup keamanan dengan ID sg-903004f8. Perhatikan bahwa Anda tidak dapat mereferensikan grup keamanan untuk EC2-VPC berdasarkan nama. Jika perintah berhasil, tidak ada output yang akan ditampilkan.

Perintah:

```
aws ec2 delete-security-group --group-id sg-903004f8
```

Untuk informasi selengkapnya, lihat Menggunakan Grup Keamanan di Panduan Pengguna Antarmuka Baris Perintah AWS .

- Untuk detail API, lihat [DeleteSecurityGroup](#) di Referensi AWS CLI Perintah.

Java

SDK untuk Java 2.x

Note

Ada lebih banyak tentang GitHub. Temukan contoh selengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
public static void deleteEC2SecGroup(Ec2Client ec2, String groupId) {
    try {
        DeleteSecurityGroupRequest request =
DeleteSecurityGroupRequest.builder()
            .groupId(groupId)
            .build();

        ec2.deleteSecurityGroup(request);
        System.out.println("Successfully deleted security group with Id " +
groupId);
    } catch (Ec2Exception e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
}
```

- Untuk detail API, lihat [DeleteSecurityGroup](#) di Referensi AWS SDK for Java 2.x API.

JavaScript

SDK untuk JavaScript (v3)

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
import { DeleteSecurityGroupCommand } from "@aws-sdk/client-ec2";

import { client } from "../libs/client.js";

export const main = async () => {
  const command = new DeleteSecurityGroupCommand({
    GroupId: "GROUP_ID",
  });

  try {
    await client.send(command);
    console.log("Security group deleted successfully.");
  } catch (err) {
    console.error(err);
  }
};
```

- Untuk detail API, lihat [DeleteSecurityGroup](#) di Referensi AWS SDK for JavaScript API.

Kotlin

SDK untuk Kotlin

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
suspend fun deleteEC2SecGroup(groupIdVal: String) {
    val request =
        DeleteSecurityGroupRequest {
            groupId = groupIdVal
        }

    Ec2Client { region = "us-west-2" }.use { ec2 ->
        ec2.deleteSecurityGroup(request)
        println("Successfully deleted Security Group with id $groupIdVal")
    }
}
```

- Untuk detail API, lihat [DeleteSecurityGroup](#) di AWS SDK untuk referensi API Kotlin.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini menghapus grup keamanan yang ditentukan untuk EC2-VPC. Anda diminta untuk konfirmasi sebelum operasi berlangsung, kecuali jika Anda juga menentukan parameter Force.

```
Remove-EC2SecurityGroup -GroupId sg-12345678
```

Output:

```
Confirm
Are you sure you want to perform this action?
Performing operation "Remove-EC2SecurityGroup (DeleteSecurityGroup)" on Target
"sg-12345678".
[Y] Yes [A] Yes to All [N] No [L] No to All [S] Suspend [?] Help (default is
"Y"):
```

Contoh 2: Contoh ini menghapus grup keamanan yang ditentukan untuk EC2-Classic.

```
Remove-EC2SecurityGroup -GroupName my-security-group -Force
```

- Untuk detail API, lihat [DeleteSecurityGroup](#) di Referensi AWS Tools for PowerShell Cmdlet.

Python

SDK untuk Python (Boto3)

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
class SecurityGroupWrapper:
    """Encapsulates Amazon Elastic Compute Cloud (Amazon EC2) security group
    actions."""

    def __init__(self, ec2_resource, security_group=None):
        """
        :param ec2_resource: A Boto3 Amazon EC2 resource. This high-level
        resource
                               is used to create additional high-level objects
                               that wrap low-level Amazon EC2 service actions.
        :param security_group: A Boto3 SecurityGroup object. This is a high-level
        object
                               that wraps security group actions.
        """
        self.ec2_resource = ec2_resource
        self.security_group = security_group

    @classmethod
    def from_resource(cls):
        ec2_resource = boto3.resource("ec2")
        return cls(ec2_resource)

    def delete(self):
        """
        Deletes the security group.
        """
        if self.security_group is None:
            logger.info("No security group to delete.")
            return

        group_id = self.security_group.id
```



```

try:
    self.security_group.delete()
except ClientError as err:
    logger.error(
        "Couldn't delete security group %s. Here's why: %s: %s",
        group_id,
        err.response["Error"]["Code"],
        err.response["Error"]["Message"],
    )
    raise

```

- Untuk detail API, lihat [DeleteSecurityGroup](#) di AWS SDK for Python (Boto3) Referensi API.

SAP ABAP

SDK untuk SAP ABAP

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```

TRY.
    lo_ec2->deletesecuritygroup( iv_groupid = iv_security_group_id ).
    MESSAGE 'Security group deleted.' TYPE 'I'.
    CATCH /aws1/cx_rt_service_generic INTO DATA(lo_exception).
    DATA(lv_error) = |"{ lo_exception->av_err_code }" - { lo_exception-
>av_err_msg }|.
    MESSAGE lv_error TYPE 'E'.
ENDTRY.

```

- Untuk detail API, lihat [DeleteSecurityGroup](#) di AWS SDK untuk referensi SAP ABAP API.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **DeleteSnapshot** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `DeleteSnapshot`.

CLI

AWS CLI

Untuk menghapus snapshot

Perintah contoh ini menghapus snapshot dengan ID snapshot `snap-1234567890abcdef0`.
Jika perintah berhasil, tidak ada output yang akan ditampilkan.

Perintah:

```
aws ec2 delete-snapshot --snapshot-id snap-1234567890abcdef0
```

- Untuk detail API, lihat [DeleteSnapshot](#) di Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini menghapus snapshot yang ditentukan. Anda diminta untuk konfirmasi sebelum operasi berlangsung, kecuali jika Anda juga menentukan parameter `Force`.

```
Remove-EC2Snapshot -SnapshotId snap-12345678
```

Output:

```
Confirm
Are you sure you want to perform this action?
Performing the operation "Remove-EC2Snapshot (DeleteSnapshot)" on target
"snap-12345678".
[Y] Yes [A] Yes to All [N] No [L] No to All [S] Suspend [?] Help (default is
"Y"):
```

- Untuk detail API, lihat [DeleteSnapshot](#) di Referensi AWS Tools for PowerShell Cmdlet.

Rust

SDK untuk Rust

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
async fn delete_snapshot(client: &Client, id: &str) -> Result<(), Error> {
    client.delete_snapshot().snapshot_id(id).send().await?;

    println!("Deleted");

    Ok(())
}
```

- Untuk detail API, lihat [DeleteSnapshot](#) referensi AWS SDK for Rust API.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **DeleteSpotDatafeedSubscription** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `DeleteSpotDatafeedSubscription`.

CLI

AWS CLI

Untuk membatalkan langganan umpan data Instans Spot

Perintah contoh ini menghapus langganan umpan data Spot untuk akun tersebut. Jika perintah berhasil, tidak ada output yang akan ditampilkan.

Perintah:

```
aws ec2 delete-spot-datafeed-subscription
```

- Untuk detail API, lihat [DeleteSpotDatafeedSubscription](#) di Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini menghapus umpan data instans Spot Anda. Anda diminta untuk konfirmasi sebelum operasi berlangsung, kecuali jika Anda juga menentukan parameter Force.

```
Remove-EC2SpotDatafeedSubscription
```

Output:

```
Confirm
Are you sure you want to perform this action?
Performing operation "Remove-EC2SpotDatafeedSubscription
(DeleteSpotDatafeedSubscription)" on Target "".
[Y] Yes [A] Yes to All [N] No [L] No to All [S] Suspend [?] Help (default is
"Y"):
```

- Untuk detail API, lihat [DeleteSpotDatafeedSubscription](#) di Referensi AWS Tools for PowerShell Cmdlet.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **DeleteSubnet** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `DeleteSubnet`.

CLI

AWS CLI

Untuk menghapus subnet

Contoh ini menghapus subnet yang ditentukan. Jika perintah berhasil, tidak ada output yang akan ditampilkan.

Perintah:

```
aws ec2 delete-subnet --subnet-id subnet-9d4a7b6c
```

- Untuk detail API, lihat [DeleteSubnet](#) di Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini menghapus subnet yang ditentukan. Anda diminta untuk konfirmasi sebelum operasi berlangsung, kecuali jika Anda juga menentukan parameter Force.

```
Remove-EC2Subnet -SubnetId subnet-1a2b3c4d
```

Output:

```
Confirm
Are you sure you want to perform this action?
Performing operation "Remove-EC2Subnet (DeleteSubnet)" on Target
"subnet-1a2b3c4d".
[Y] Yes [A] Yes to All [N] No [L] No to All [S] Suspend [?] Help (default is
"Y"):
```

- Untuk detail API, lihat [DeleteSubnet](#) di Referensi AWS Tools for PowerShell Cmdlet.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **DeleteTags** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `DeleteTags`.

CLI

AWS CLI

Contoh 1: Untuk menghapus tag dari sumber daya

`delete-tags` Contoh berikut menghapus tag `Stack=Test` dari gambar yang ditentukan. Saat Anda menentukan nilai dan nama kunci, tag akan dihapus hanya jika nilai tag cocok dengan nilai yang ditentukan.

```
aws ec2 delete-tags \  
  --resources ami-1234567890abcdef0 \  
  --tags Key=Stack,Value=Test
```

Ini opsional untuk menentukan nilai untuk tag. `delete-tags` Contoh berikut menghapus tag dengan nama kunci `purpose` dari contoh yang ditentukan, terlepas dari nilai tag untuk tag.

```
aws ec2 delete-tags \  
  --resources i-1234567890abcdef0 \  
  --tags Key=purpose
```

Jika Anda menentukan string kosong sebagai nilai tag, tag akan dihapus hanya jika nilai tag adalah string kosong. `delete-tags` Contoh berikut menentukan string kosong sebagai nilai tag untuk tag untuk menghapus.

```
aws ec2 delete-tags \  
  --resources i-1234567890abcdef0 \  
  --tags Key=Name,Value=
```

Contoh 2: Untuk menghapus tag dari beberapa sumber

`delete-tags` Contoh berikut menghapus tag `purpose=test` dari kedua instance dan AMI. Seperti yang ditunjukkan pada contoh sebelumnya, Anda dapat menghilangkan nilai tag dari perintah.

```
aws ec2 delete-tags \  
  --resources i-1234567890abcdef0 ami-1234567890abcdef0 \  
  --tags Key=Purpose
```

- Untuk detail API, lihat [DeleteTags](#) di Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini menghapus tag yang ditentukan dari sumber daya yang ditentukan, terlepas dari nilai tag. Sintaks yang digunakan oleh contoh ini memerlukan PowerShell versi 3 atau yang lebih baru.

```
Remove-EC2Tag -Resource i-12345678 -Tag @{ Key="myTag" } -Force
```

Contoh 2: Contoh ini menghapus tag yang ditentukan dari sumber daya yang ditentukan, tetapi hanya jika nilai tag cocok. Sintaks yang digunakan oleh contoh ini memerlukan PowerShell versi 3 atau yang lebih baru.

```
Remove-EC2Tag -Resource i-12345678 -Tag @{ Key="myTag";Value="myTagValue" } -Force
```

Contoh 3: Contoh ini menghapus tag yang ditentukan dari sumber daya yang ditentukan, terlepas dari nilai tag.

```
$tag = New-Object Amazon.EC2.Model.Tag
$tag.Key = "myTag"

Remove-EC2Tag -Resource i-12345678 -Tag $tag -Force
```

Contoh 4: Contoh ini menghapus tag yang ditentukan dari sumber daya yang ditentukan, tetapi hanya jika nilai tag cocok.

```
$tag = New-Object Amazon.EC2.Model.Tag
$tag.Key = "myTag"
$tag.Value = "myTagValue"

Remove-EC2Tag -Resource i-12345678 -Tag $tag -Force
```

- Untuk detail API, lihat [DeleteTags](#) di Referensi AWS Tools for PowerShell Cmdlet.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **DeleteVolume** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `DeleteVolume`.

CLI

AWS CLI

Untuk menghapus volume

Perintah contoh ini menghapus volume yang tersedia dengan ID volume dari.

`vol-049df61146c4d7901` Jika perintah berhasil, tidak ada output yang akan ditampilkan.

Perintah:

```
aws ec2 delete-volume --volume-id vol-049df61146c4d7901
```

- Untuk detail API, lihat [DeleteVolume](#) di Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini melepaskan volume yang ditentukan. Anda diminta untuk konfirmasi sebelum operasi berlangsung, kecuali jika Anda juga menentukan parameter `Force`.

```
Remove-EC2Volume -VolumeId vol-12345678
```

Output:

```
Confirm
Are you sure you want to perform this action?
Performing the operation "Remove-EC2Volume (DeleteVolume)" on target
"vol-12345678".
[Y] Yes [A] Yes to All [N] No [L] No to All [S] Suspend [?] Help (default is
"Y"):
```

- Untuk detail API, lihat [DeleteVolume](#) di Referensi AWS Tools for PowerShell Cmdlet.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **DeleteVpc** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `DeleteVpc`.

CLI

AWS CLI

Untuk menghapus VPC

Contoh ini menghapus VPC yang ditentukan. Jika perintah berhasil, tidak ada output yang akan ditampilkan.

Perintah:

```
aws ec2 delete-vpc --vpc-id vpc-a01106c2
```

- Untuk detail API, lihat [DeleteVpc](#) di Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini menghapus VPC yang ditentukan. Anda diminta untuk konfirmasi sebelum operasi berlangsung, kecuali jika Anda juga menentukan parameter `Force`.

```
Remove-EC2Vpc -VpcId vpc-12345678
```

Output:

```
Confirm
Are you sure you want to perform this action?
Performing operation "Remove-EC2Vpc (DeleteVpc)" on Target "vpc-12345678".
[Y] Yes [A] Yes to All [N] No [L] No to All [S] Suspend [?] Help (default is
"Y"):
```

- Untuk detail API, lihat [DeleteVpc](#) di Referensi AWS Tools for PowerShell Cmdlet.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **DeleteVpnConnection** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `DeleteVpnConnection`.

CLI

AWS CLI

Untuk menghapus koneksi VPN

Contoh ini menghapus koneksi VPN yang ditentukan. Jika perintah berhasil, tidak ada output yang akan ditampilkan.

Perintah:

```
aws ec2 delete-vpn-connection --vpn-connection-id vpn-40f41529
```

- Untuk detail API, lihat [DeleteVpnConnection](#) di Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini menghapus koneksi VPN yang ditentukan. Anda diminta untuk konfirmasi sebelum operasi berlangsung, kecuali jika Anda juga menentukan parameter `Force`.

```
Remove-EC2VpnConnection -VpnConnectionId vpn-12345678
```

Output:

```
Confirm
Are you sure you want to perform this action?
Performing operation "Remove-EC2VpnConnection (DeleteVpnConnection)" on Target
"vpn-12345678".
[Y] Yes  [A] Yes to All  [N] No  [L] No to All  [S] Suspend  [?] Help (default is
"Y"):
```

- Untuk detail API, lihat [DeleteVpnConnection](#) di Referensi AWS Tools for PowerShell Cmdlet.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan `DeleteVpnConnectionRoute` dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `DeleteVpnConnectionRoute`.

CLI

AWS CLI

Untuk menghapus rute statis dari koneksi VPN

Contoh ini menghapus rute statis yang ditentukan dari koneksi VPN yang ditentukan. Jika perintah berhasil, tidak ada output yang akan ditampilkan.

Perintah:

```
aws ec2 delete-vpn-connection-route --vpn-connection-id vpn-40f41529 --
destination-cidr-block 11.12.0.0/16
```

- Untuk detail API, lihat [DeleteVpnConnectionRoute](#) di Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini menghapus rute statis yang ditentukan dari koneksi VPN yang ditentukan. Anda diminta untuk konfirmasi sebelum operasi berlangsung, kecuali jika Anda juga menentukan parameter `Force`.

```
Remove-EC2VpnConnectionRoute -VpnConnectionId vpn-12345678 -DestinationCidrBlock
11.12.0.0/16
```

Output:

```
Confirm
Are you sure you want to perform this action?
```

```
Performing operation "Remove-EC2VpnConnectionRoute (DeleteVpnConnectionRoute)" on
Target "vpn-12345678".
[Y] Yes [A] Yes to All [N] No [L] No to All [S] Suspend [?] Help (default is
"Y"):
```

- Untuk detail API, lihat [DeleteVpnConnectionRoute](#) di Referensi AWS Tools for PowerShell Cmdlet.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **DeleteVpnGateway** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `DeleteVpnGateway`.

CLI

AWS CLI

Untuk menghapus gateway pribadi virtual

Contoh ini menghapus gateway pribadi virtual yang ditentukan. Jika perintah berhasil, tidak ada output yang akan ditampilkan.

Perintah:

```
aws ec2 delete-vpn-gateway --vpn-gateway-id vgw-9a4cacf3
```

- Untuk detail API, lihat [DeleteVpnGateway](#) di Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini menghapus gateway pribadi virtual yang ditentukan. Anda diminta untuk konfirmasi sebelum operasi berlangsung, kecuali jika Anda juga menentukan parameter `Force`.

```
Remove-EC2VpnGateway -VpnGatewayId vgw-1a2b3c4d
```

Output:

```
Confirm
Are you sure you want to perform this action?
Performing operation "Remove-EC2VpnGateway (DeleteVpnGateway)" on Target
"vgw-1a2b3c4d".
[Y] Yes [A] Yes to All [N] No [L] No to All [S] Suspend [?] Help (default is
"Y"):
```

- Untuk detail API, lihat [DeleteVpnGateway](#) di Referensi AWS Tools for PowerShell Cmdlet.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **DeregisterImage** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `DeregisterImage`.

CLI

AWS CLI

Untuk membatalkan pendaftaran AMI

Contoh ini membatalkan pendaftaran AMI yang ditentukan. Jika perintah berhasil, tidak ada output yang akan ditampilkan.

Perintah:

```
aws ec2 deregister-image --image-id ami-4fa54026
```

- Untuk detail API, lihat [DeregisterImage](#) di Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini membatalkan pendaftaran AMI yang ditentukan.

```
Unregister-EC2Image -ImageId ami-12345678
```

- Untuk detail API, lihat [DeregisterImage](#) di Referensi AWS Tools for PowerShell Cmdlet.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **DescribeAccountAttributes** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `DescribeAccountAttributes`.

CLI

AWS CLI

Untuk menjelaskan semua atribut untuk AWS akun Anda

Contoh ini menjelaskan atribut untuk AWS akun Anda.

Perintah:

```
aws ec2 describe-account-attributes
```

Output:

```
{
  "AccountAttributes": [
    {
      "AttributeName": "vpc-max-security-groups-per-interface",
      "AttributeValues": [
        {
          "AttributeValue": "5"
        }
      ]
    },
    {
      "AttributeName": "max-instances",
      "AttributeValues": [
        {
          "AttributeValue": "20"
        }
      ]
    },
    {
      "AttributeName": "supported-platforms",
```

```
    "AttributeValues": [
      {
        "AttributeValue": "EC2"
      },
      {
        "AttributeValue": "VPC"
      }
    ],
    {
      "AttributeName": "default-vpc",
      "AttributeValues": [
        {
          "AttributeValue": "none"
        }
      ]
    },
    {
      "AttributeName": "max-elastic-ips",
      "AttributeValues": [
        {
          "AttributeValue": "5"
        }
      ]
    },
    {
      "AttributeName": "vpc-max-elastic-ips",
      "AttributeValues": [
        {
          "AttributeValue": "5"
        }
      ]
    }
  ]
}
```

Untuk mendeskripsikan satu atribut untuk AWS akun Anda

Contoh ini menjelaskan `supported-platforms` atribut untuk AWS akun Anda.

Perintah:

```
aws ec2 describe-account-attributes --attribute-names supported-platforms
```

Output:

```
{
  "AccountAttributes": [
    {
      "AttributeName": "supported-platforms",
      "AttributeValues": [
        {
          "AttributeValue": "EC2"
        },
        {
          "AttributeValue": "VPC"
        }
      ]
    }
  ]
}
```

- Untuk detail API, lihat [DescribeAccountAttributes](#) di Referensi AWS CLI Perintah.

PowerShell**Alat untuk PowerShell**

Contoh 1: Contoh ini menjelaskan apakah Anda dapat meluncurkan instance ke EC2-Classic dan EC2-VPC di wilayah tersebut, atau hanya ke EC2-VPC.

```
(Get-EC2AccountAttribute -AttributeName supported-platforms).AttributeValues
```

Output:

```
AttributeValue
-----
EC2
VPC
```

Contoh 2: Contoh ini menjelaskan VPC default Anda, atau 'tidak ada' jika Anda tidak memiliki VPC default di wilayah tersebut.

```
(Get-EC2AccountAttribute -AttributeName default-vpc).AttributeValues
```


Output:

```
AttributeValue
-----
vpc-12345678
```

Contoh 3: Contoh ini menjelaskan jumlah maksimum instans On-Demand yang dapat Anda jalankan.

```
(Get-EC2AccountAttribute -AttributeName max-instances).AttributeValues
```

Output:

```
AttributeValue
-----
20
```

- Untuk detail API, lihat [DescribeAccountAttributes](#) di Referensi AWS Tools for PowerShell Cmdlet.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **DescribeAddresses** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `DescribeAddresses`.

C++

SDK untuk C++

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
Aws::EC2::EC2Client ec2Client(clientConfiguration);
```

```

Aws::EC2::Model::DescribeAddressesRequest request;
auto outcome = ec2Client.DescribeAddresses(request);
if (outcome.IsSuccess()) {
    std::cout << std::left << std::setw(20) << "InstanceId" <<
        std::setw(15) << "Public IP" << std::setw(10) << "Domain" <<
        std::setw(30) << "Allocation ID" << std::setw(25) <<
        "NIC ID" << std::endl;

    const auto &addresses = outcome.GetResult().GetAddresses();
    for (const auto &address: addresses) {
        Aws::String domainString =
            Aws::EC2::Model::DomainTypeMapper::GetNameForDomainType(
                address.GetDomain());

        std::cout << std::left << std::setw(20) <<
            address.GetInstanceId() << std::setw(15) <<
            address.GetPublicIp() << std::setw(10) << domainString <<
            std::setw(30) << address.GetAllocationId() << std::setw(25)
            << address.GetNetworkInterfaceId() << std::endl;
    }
}
else {
    std::cerr << "Failed to describe Elastic IP addresses:" <<
        outcome.GetError().GetMessage() << std::endl;
}

```

- Untuk detail API, lihat [DescribeAddresses](#) di Referensi AWS SDK for C++ API.

CLI

AWS CLI

Contoh 1: Untuk mengambil detail tentang semua alamat IP Elastis Anda

Contoh `describe addresses` berikut menampilkan detail tentang alamat IP Elastis Anda.

```
aws ec2 describe-addresses
```

Output:

```
{
```

```

"Addresses": [
  {
    "InstanceId": "i-1234567890abcdef0",
    "PublicIp": "198.51.100.0",
    "PublicIpv4Pool": "amazon",
    "Domain": "standard"
  },
  {
    "Domain": "vpc",
    "PublicIpv4Pool": "amazon",
    "InstanceId": "i-1234567890abcdef0",
    "NetworkInterfaceId": "eni-12345678",
    "AssociationId": "eipassoc-12345678",
    "NetworkInterfaceOwnerId": "123456789012",
    "PublicIp": "203.0.113.0",
    "AllocationId": "eipalloc-12345678",
    "PrivateIpAddress": "10.0.1.241"
  }
]
}

```

Contoh 2: Guna mengambil detail alamat IP Elastis Anda untuk EC2-VPC

Contoh `describe-addresses` berikut menampilkan detail tentang alamat IP Elastis Anda untuk digunakan dengan instans di VPC.

```

aws ec2 describe-addresses \
  --filters "Name=domain,Values=vpc"

```

Output:

```

{
  "Addresses": [
    {
      "Domain": "vpc",
      "PublicIpv4Pool": "amazon",
      "InstanceId": "i-1234567890abcdef0",
      "NetworkInterfaceId": "eni-12345678",
      "AssociationId": "eipassoc-12345678",
      "NetworkInterfaceOwnerId": "123456789012",
      "PublicIp": "203.0.113.0",
      "AllocationId": "eipalloc-12345678",
      "PrivateIpAddress": "10.0.1.241"
    }
  ]
}

```

```
    }  
  ]  
}
```

Contoh 3: Untuk mengambil detail tentang alamat IP Elastis yang ditentukan oleh ID alokasi

Contoh `describe-addresses` berikut menampilkan detail tentang alamat IP Elastis dengan ID alokasi tertentu, yang dikaitkan dengan instans di EC2-VPC.

```
aws ec2 describe-addresses \  
  --allocation-ids eipalloc-282d9641
```

Output:

```
{  
  "Addresses": [  
    {  
      "Domain": "vpc",  
      "PublicIpv4Pool": "amazon",  
      "InstanceId": "i-1234567890abcdef0",  
      "NetworkInterfaceId": "eni-1a2b3c4d",  
      "AssociationId": "eipassoc-123abc12",  
      "NetworkInterfaceOwnerId": "1234567891012",  
      "PublicIp": "203.0.113.25",  
      "AllocationId": "eipalloc-282d9641",  
      "PrivateIpAddress": "10.251.50.12"  
    }  
  ]  
}
```

Contoh 4: Untuk mengambil detail tentang alamat IP Elastis yang ditentukan oleh alamat IP privat VPC-nya

Contoh `describe-addresses` berikut menampilkan detail tentang alamat IP Elastis yang terkait dengan alamat IP privat tertentu di EC2-VPC.

```
aws ec2 describe-addresses \  
  --filters "Name=private-ip-address,Values=10.251.50.12"
```

Contoh 5: Untuk mengambil detail tentang alamat IP Elastis di EC2-Classic

Contoh `describe-addresses` berikut menampilkan detail tentang alamat IP Elastis Anda untuk digunakan di EC2-Classic.

```
aws ec2 describe-addresses \  
  --filters "Name=domain,Values=standard"
```

Output:

```
{  
  "Addresses": [  
    {  
      "InstanceId": "i-1234567890abcdef0",  
      "PublicIp": "203.0.110.25",  
      "PublicIpv4Pool": "amazon",  
      "Domain": "standard"  
    }  
  ]  
}
```

Contoh 6: Untuk mengambil detail tentang alamat IP Elastis yang ditentukan oleh alamat IP publiknya

Contoh `describe-addresses` berikut menampilkan detail tentang alamat IP Elastis dengan nilai `203.0.110.25`, yang dikaitkan dengan instans di EC2-Classic.

```
aws ec2 describe-addresses \  
  --public-ips 203.0.110.25
```

Output:

```
{  
  "Addresses": [  
    {  
      "InstanceId": "i-1234567890abcdef0",  
      "PublicIp": "203.0.110.25",  
      "PublicIpv4Pool": "amazon",  
      "Domain": "standard"  
    }  
  ]  
}
```

- Untuk detail API, lihat [DescribeAddresses](#) di Referensi AWS CLI Perintah.

JavaScript

SDK untuk JavaScript (v3)

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
import { DescribeAddressesCommand } from "@aws-sdk/client-ec2";

import { client } from "../libs/client.js";

export const main = async () => {
  const command = new DescribeAddressesCommand({
    // You can omit this property to show all addresses.
    AllocationIds: ["ALLOCATION_ID"],
  });

  try {
    const { Addresses } = await client.send(command);
    const addressList = Addresses.map((address) => ` • ${address.PublicIp}`);
    console.log("Elastic IP addresses:");
    console.log(addressList.join("\n"));
  } catch (err) {
    console.error(err);
  }
};
```

- Untuk detail API, lihat [DescribeAddresses](#) di Referensi AWS SDK for JavaScript API.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini menjelaskan alamat IP Elastis yang ditentukan untuk instance di EC2-Classic.

```
Get-EC2Address -AllocationId eipalloc-12345678
```

Output:

```
AllocationId      : eipalloc-12345678
AssociationId     : eipassoc-12345678
Domain           : vpc
InstanceId       : i-87654321
NetworkInterfaceId : eni-12345678
NetworkInterfaceOwnerId : 12345678
PrivateIpAddress  : 10.0.2.172
PublicIp         : 198.51.100.2
```

Contoh 2: Contoh ini menjelaskan alamat IP Elastis Anda untuk instance di VPC. Sintaks ini membutuhkan PowerShell versi 3 atau yang lebih baru.

```
Get-EC2Address -Filter @{ Name="domain";Values="vpc" }
```

Contoh 3: Contoh ini menjelaskan alamat IP Elastis yang ditentukan untuk instance di EC2-Classic.

```
Get-EC2Address -PublicIp 203.0.113.17
```

Output:

```
AllocationId      :
AssociationId     :
Domain           : standard
InstanceId       : i-12345678
NetworkInterfaceId :
NetworkInterfaceOwnerId :
PrivateIpAddress  :
PublicIp         : 203.0.113.17
```

Contoh 4: Contoh ini menjelaskan alamat IP Elastis Anda untuk instance di EC2-Classic. Sintaks ini membutuhkan PowerShell versi 3 atau yang lebih baru.

```
Get-EC2Address -Filter @{ Name="domain";Values="standard" }
```

Contoh 5: Contoh ini menjelaskan semua alamat IP Elastis Anda.

```
Get-EC2Address
```

Contoh 6: Contoh ini mengembalikan IP publik dan pribadi untuk id contoh yang disediakan dalam filter

```
Get-EC2Address -Region eu-west-1 -Filter @{Name="instance-id";Values="i-0c12d3f4f567ffb89"} | Select-Object PrivateIpAddress, PublicIp
```

Output:

```
PrivateIpAddress PublicIp
-----
10.0.0.99          63.36.5.227
```

Contoh 7: Contoh ini mengambil semua IP Elastis dengan id alokasi, id asosiasi, dan id instans

```
Get-EC2Address -Region eu-west-1 | Select-Object InstanceId, AssociationId, AllocationId, PublicIp
```

Output:

```
InstanceId          AssociationId        AllocationId
-----
PublicIp
-----
-----
17.212.120.178
i-0c123dfd3415bac67 eipassoc-0e123456bb7890bdb eipalloc-01cd23ebf45f7890c
17.212.124.77
eipalloc-012345678eeabcfad
17.212.225.7
i-0123d405c67e89a0c eipassoc-0c123b456783966ba eipalloc-0123cdd456a8f7892
37.216.52.173
i-0f1bf2f34c5678d09 eipassoc-0e12934568a952d96 eipalloc-0e1c23e4d5e6789e4
37.218.222.278
i-012e3cb4df567e8aa eipassoc-0d1b2fa4d67d03810 eipalloc-0123f456f78a01b58
37.210.82.27
i-0123bcf4b567890e1 eipassoc-01d2345f678903fb1 eipalloc-0e1db23cfef5c45c7
37.215.222.270
```


Contoh 8: Contoh ini mengambil daftar alamat IP EC2 yang cocok dengan kunci tag 'Kategori' dengan nilai 'Prod'

```
Get-EC2Address -Filter @{"Name":"tag:Category";Values="Prod"}
```

Output:

```
AllocationId      : eipalloc-0123f456f81a01b58
AssociationId     : eipassoc-0d1b23a456d103810
CustomerOwnedIp  :
CustomerOwnedIpv4Pool :
Domain           : vpc
InstanceId       : i-012e3cb4df567e1aa
NetworkBorderGroup : eu-west-1
NetworkInterfaceId : eni-0123f41d5a60d5f40
NetworkInterfaceOwnerId : 123456789012
PrivateIpAddress : 192.168.1.84
PublicIp         : 34.250.81.29
PublicIpv4Pool   : amazon
Tags             : {Category, Name}
```

- Untuk detail API, lihat [DescribeAddresses](#) di Referensi AWS Tools for PowerShell Cmdlet.

SAP ABAP

SDK untuk SAP ABAP

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
TRY.
    oo_result = lo_ec2->describeaddresses( ) .
    oo_result is returned for testing purposes. "
    DATA(lt_addresses) = oo_result->get_addresses( ).
    MESSAGE 'Retrieved information about Elastic IP addresses.' TYPE 'I'.
    CATCH /aws1/cx_rt_service_generic INTO DATA(lo_exception).
    DATA(lv_error) = |"{ lo_exception->av_err_code }" - { lo_exception->av_err_msg }|.
ENDTRY.
```

```
MESSAGE lv_error TYPE 'E'.  
ENDTRY.
```

- Untuk detail API, lihat [DescribeAddresses](#) di AWS SDK untuk referensi SAP ABAP API.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **DescribeAvailabilityZones** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `DescribeAvailabilityZones`.

Contoh tindakan adalah kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Anda dapat melihat tindakan ini dalam konteks dalam contoh kode berikut:

- [Membangun dan mengelola layanan yang tangguh](#)

.NET

AWS SDK for .NET

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
/// <summary>  
/// Get a list of Availability Zones in the AWS Region of the Amazon EC2  
Client.  
/// </summary>  
/// <returns>A list of availability zones.</returns>  
public async Task<List<string>> DescribeAvailabilityZones()  
{  
    var zoneResponse = await _amazonEc2.DescribeAvailabilityZonesAsync(  
        new DescribeAvailabilityZonesRequest());  
    return zoneResponse.AvailabilityZones.Select(z => z.ZoneName).ToList();  
}
```

- Untuk detail API, lihat [DescribeAvailabilityZones](#) di Referensi AWS SDK for .NET API.

C++

SDK untuk C++

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
Aws::EC2::EC2Client ec2Client(clientConfiguration);

Aws::EC2::Model::DescribeAvailabilityZonesRequest describe_request;
auto describe_outcome =
ec2Client.DescribeAvailabilityZones(describe_request);

if (describe_outcome.IsSuccess()) {
    std::cout << std::left <<
        std::setw(32) << "ZoneName" <<
        std::setw(20) << "State" <<
        std::setw(32) << "Region" << std::endl;

    const auto &zones =
        describe_outcome.GetResult().GetAvailabilityZones();

    for (const auto &zone: zones) {
        Aws::String stateString =
            Aws::EC2::Model::AvailabilityZoneStateMapper::GetNameForAvailabilityZoneState(
                zone.GetState());
        std::cout << std::left <<
            std::setw(32) << zone.GetZoneName() <<
            std::setw(20) << stateString <<
            std::setw(32) << zone.GetRegionName() << std::endl;
    }
}
else {
    std::cerr << "Failed to describe availability zones:" <<
```

```
        describe_outcome.GetError().GetMessage() << std::endl;
    result = false;
}
```

- Untuk detail API, lihat [DescribeAvailabilityZones](#) di Referensi AWS SDK for C++ API.

CLI

AWS CLI

Untuk menjelaskan Zona Ketersediaan Anda

Contoh `describe-availability-zones` berikut menampilkan detail untuk Zona Ketersediaan yang tersedia untuk Anda. Responsnya mencakup Zona Ketersediaan hanya untuk Wilayah saat ini. Dalam contoh ini, respons menggunakan default profil Wilayah `us-west-2` (Oregon).

```
aws ec2 describe-availability-zones
```

Output:

```
{
  "AvailabilityZones": [
    {
      "State": "available",
      "OptInStatus": "opt-in-not-required",
      "Messages": [],
      "RegionName": "us-west-2",
      "ZoneName": "us-west-2a",
      "ZoneId": "usw2-az1",
      "GroupName": "us-west-2",
      "NetworkBorderGroup": "us-west-2"
    },
    {
      "State": "available",
      "OptInStatus": "opt-in-not-required",
      "Messages": [],
      "RegionName": "us-west-2",
      "ZoneName": "us-west-2b",
      "ZoneId": "usw2-az2",
      "GroupName": "us-west-2",

```

```

    "NetworkBorderGroup": "us-west-2"
  },
  {
    "State": "available",
    "OptInStatus": "opt-in-not-required",
    "Messages": [],
    "RegionName": "us-west-2",
    "ZoneName": "us-west-2c",
    "ZoneId": "usw2-az3",
    "GroupName": "us-west-2",
    "NetworkBorderGroup": "us-west-2"
  },
  {
    "State": "available",
    "OptInStatus": "opt-in-not-required",
    "Messages": [],
    "RegionName": "us-west-2",
    "ZoneName": "us-west-2d",
    "ZoneId": "usw2-az4",
    "GroupName": "us-west-2",
    "NetworkBorderGroup": "us-west-2"
  },
  {
    "State": "available",
    "OptInStatus": "opted-in",
    "Messages": [],
    "RegionName": "us-west-2",
    "ZoneName": "us-west-2-lax-1a",
    "ZoneId": "usw2-lax1-az1",
    "GroupName": "us-west-2-lax-1",
    "NetworkBorderGroup": "us-west-2-lax-1"
  }
]
}

```

- Untuk detail API, lihat [DescribeAvailabilityZones](#) di Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini menjelaskan Availability Zone untuk wilayah saat ini yang tersedia untuk Anda.

```
Get-EC2AvailabilityZone
```

Output:

Messages	RegionName	State	ZoneName
{} -----	us-west-2 -----	available -----	us-west-2a -----
{} -----	us-west-2 -----	available -----	us-west-2b -----
{} -----	us-west-2 -----	available -----	us-west-2c -----

Contoh 2: Contoh ini menjelaskan Availability Zone yang berada dalam keadaan terganggu. Sintaks yang digunakan oleh contoh ini membutuhkan PowerShell versi 3 atau lebih tinggi.

```
Get-EC2AvailabilityZone -Filter @{ Name="state";Values="impaired" }
```

Contoh 3: Dengan PowerShell versi 2, Anda harus menggunakan New-Object untuk membuat filter.

```
$filter = New-Object Amazon.EC2.Model.Filter
$filter.Name = "state"
$filter.Values = "impaired"

Get-EC2AvailabilityZone -Filter $filter
```

- Untuk detail API, lihat [DescribeAvailabilityZones](#) di Referensi AWS Tools for PowerShell Cmdlet.

Python

SDK untuk Python (Boto3)

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
class AutoScaler:
    """
```

```
Encapsulates Amazon EC2 Auto Scaling and EC2 management actions.
"""

def __init__(
    self,
    resource_prefix,
    inst_type,
    ami_param,
    autoscaling_client,
    ec2_client,
    ssm_client,
    iam_client,
):
    """
    :param resource_prefix: The prefix for naming AWS resources that are
    created by this class.
    :param inst_type: The type of EC2 instance to create, such as t3.micro.
    :param ami_param: The Systems Manager parameter used to look up the AMI
    that is
        created.
    :param autoscaling_client: A Boto3 EC2 Auto Scaling client.
    :param ec2_client: A Boto3 EC2 client.
    :param ssm_client: A Boto3 Systems Manager client.
    :param iam_client: A Boto3 IAM client.
    """
    self.inst_type = inst_type
    self.ami_param = ami_param
    self.autoscaling_client = autoscaling_client
    self.ec2_client = ec2_client
    self.ssm_client = ssm_client
    self.iam_client = iam_client
    self.launch_template_name = f"{resource_prefix}-template"
    self.group_name = f"{resource_prefix}-group"
    self.instance_policy_name = f"{resource_prefix}-pol"
    self.instance_role_name = f"{resource_prefix}-role"
    self.instance_profile_name = f"{resource_prefix}-prof"
    self.bad_creds_policy_name = f"{resource_prefix}-bc-pol"
    self.bad_creds_role_name = f"{resource_prefix}-bc-role"
    self.bad_creds_profile_name = f"{resource_prefix}-bc-prof"
    self.key_pair_name = f"{resource_prefix}-key-pair"

def get_availability_zones(self):
    """
```

```
Gets a list of Availability Zones in the AWS Region of the Amazon EC2
client.
```

```
:return: The list of Availability Zones for the client Region.
"""
try:
    response = self.ec2_client.describe_availability_zones()
    zones = [zone["ZoneName"] for zone in response["AvailabilityZones"]]
except ClientError as err:
    raise AutoScalerError(f"Couldn't get availability zones: {err}.")
else:
    return zones
```

- Untuk detail API, lihat [DescribeAvailabilityZones](#) di AWS SDK for Python (Boto3) Referensi API.

SAP ABAP

SDK untuk SAP ABAP

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
TRY.
    oo_result = lo_ec2->describeavailabilityzones( ).
    " oo_result is returned for testing purposes. "
    DATA(lt_zones) = oo_result->get_availabilityzones( ).
    MESSAGE 'Retrieved information about Availability Zones.' TYPE 'I'.

    CATCH /aws1/cx_rt_service_generic INTO DATA(lo_exception).
    DATA(lv_error) = |"{ lo_exception->av_err_code }" - { lo_exception-
>av_err_msg }|.
    MESSAGE lv_error TYPE 'E'.
ENDTRY.
```


- Untuk detail API, lihat [DescribeAvailabilityZones](#) di AWS SDK untuk referensi SAP ABAP API.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **DescribeBundleTasks** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `DescribeBundleTasks`.

CLI

AWS CLI

Untuk mendeskripsikan tugas bundel Anda

Contoh ini menjelaskan semua tugas bundel Anda.

Perintah:

```
aws ec2 describe-bundle-tasks
```

Output:

```
{
  "BundleTasks": [
    {
      "UpdateTime": "2015-09-15T13:26:54.000Z",
      "InstanceId": "i-1234567890abcdef0",
      "Storage": {
        "S3": {
          "Prefix": "winami",
          "Bucket": "bundletasks"
        }
      },
      "State": "bundling",
      "StartTime": "2015-09-15T13:24:35.000Z",
      "Progress": "3%",
      "BundleId": "bun-2a4e041c"
    }
  ]
}
```

```
}
```

- Untuk detail API, lihat [DescribeBundleTasks](#) di Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini menjelaskan tugas bundel yang ditentukan.

```
Get-EC2BundleTask -BundleId bun-12345678
```

Contoh 2: Contoh ini menjelaskan tugas bundel yang statusnya 'lengkap' atau 'gagal'.

```
$filter = New-Object Amazon.EC2.Model.Filter
$filter.Name = "state"
$filter.Values = @( "complete", "failed" )

Get-EC2BundleTask -Filter $filter
```

- Untuk detail API, lihat [DescribeBundleTasks](#) di Referensi AWS Tools for PowerShell Cmdlet.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **DescribeCapacityReservations** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `DescribeCapacityReservations`.

CLI

AWS CLI

Contoh 1: Untuk menggambarkan satu atau lebih reservasi kapasitas Anda

`describe-capacity-reservations` Contoh berikut menampilkan detail tentang semua reservasi kapasitas Anda di AWS Wilayah saat ini.

```
aws ec2 describe-capacity-reservations
```

Output:

```
{
  "CapacityReservations": [
    {
      "CapacityReservationId": "cr-1234abcd56EXAMPLE ",
      "EndDateType": "unlimited",
      "AvailabilityZone": "eu-west-1a",
      "InstanceMatchCriteria": "open",
      "Tags": [],
      "EphemeralStorage": false,
      "CreateDate": "2019-08-16T09:03:18.000Z",
      "AvailableInstanceCount": 1,
      "InstancePlatform": "Linux/UNIX",
      "TotalInstanceCount": 1,
      "State": "active",
      "Tenancy": "default",
      "EbsOptimized": true,
      "InstanceType": "a1.medium"
    },
    {
      "CapacityReservationId": "cr-abcdEXAMPLE9876ef ",
      "EndDateType": "unlimited",
      "AvailabilityZone": "eu-west-1a",
      "InstanceMatchCriteria": "open",
      "Tags": [],
      "EphemeralStorage": false,
      "CreateDate": "2019-08-07T11:34:19.000Z",
      "AvailableInstanceCount": 3,
      "InstancePlatform": "Linux/UNIX",
      "TotalInstanceCount": 3,
      "State": "cancelled",
      "Tenancy": "default",
      "EbsOptimized": true,
      "InstanceType": "m5.large"
    }
  ]
}
```

Contoh 2: Untuk menggambarkan satu atau lebih reservasi kapasitas Anda

`describe-capacity-reservations` Contoh berikut menampilkan rincian tentang reservasi kapasitas yang ditentukan.

```
aws ec2 describe-capacity-reservations \  
  --capacity-reservation-ids cr-1234abcd56EXAMPLE
```

Output:

```
{  
  "CapacityReservations": [  
    {  
      "CapacityReservationId": "cr-1234abcd56EXAMPLE",  
      "EndDateType": "unlimited",  
      "AvailabilityZone": "eu-west-1a",  
      "InstanceMatchCriteria": "open",  
      "Tags": [],  
      "EphemeralStorage": false,  
      "CreateDate": "2019-08-16T09:03:18.000Z",  
      "AvailableInstanceCount": 1,  
      "InstancePlatform": "Linux/UNIX",  
      "TotalInstanceCount": 1,  
      "State": "active",  
      "Tenancy": "default",  
      "EbsOptimized": true,  
      "InstanceType": "a1.medium"  
    }  
  ]  
}
```

Untuk informasi selengkapnya, lihat [Melihat Reservasi Kapasitas](#) di Panduan Pengguna Amazon Elastic Compute Cloud untuk Instans Linux.

- Untuk detail API, lihat [DescribeCapacityReservations](#) di Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini menjelaskan satu atau beberapa Reservasi Kapasitas Anda untuk wilayah tersebut

```
Get-EC2CapacityReservation -Region eu-west-1
```

Output:

```
AvailabilityZone      : eu-west-1b
AvailableInstanceCount : 2
CapacityReservationId : cr-0c1f2345db6f7cdba
CreateDate           : 3/28/2019 9:29:41 AM
EbsOptimized         : True
EndDate              : 1/1/0001 12:00:00 AM
EndDateType          : unlimited
EphemeralStorage     : False
InstanceMatchCriteria : open
InstancePlatform     : Windows
InstanceType         : m4.xlarge
State                : active
Tags                 : {}
Tenancy              : default
TotalInstanceCount   : 2
```

- Untuk detail API, lihat [DescribeCapacityReservations](#) di Referensi AWS Tools for PowerShell Cmdlet.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **DescribeCustomerGateways** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `DescribeCustomerGateways`.

CLI

AWS CLI

Untuk mendeskripsikan gateway pelanggan Anda

Contoh ini menjelaskan gateway pelanggan Anda.

Perintah:

```
aws ec2 describe-customer-gateways
```

Output:

```
{
  "CustomerGateways": [
    {
      "CustomerGatewayId": "cgw-b4dc3961",
      "IpAddress": "203.0.113.12",
      "State": "available",
      "Type": "ipsec.1",
      "BgpAsn": "65000"
    },
    {
      "CustomerGatewayId": "cgw-0e11f167",
      "IpAddress": "12.1.2.3",
      "State": "available",
      "Type": "ipsec.1",
      "BgpAsn": "65534"
    }
  ]
}
```

Untuk menggambarkan gateway pelanggan tertentu

Contoh ini menjelaskan gateway pelanggan yang ditentukan.

Perintah:

```
aws ec2 describe-customer-gateways --customer-gateway-ids cgw-0e11f167
```

Output:

```
{
  "CustomerGateways": [
    {
      "CustomerGatewayId": "cgw-0e11f167",
      "IpAddress": "12.1.2.3",
      "State": "available",
      "Type": "ipsec.1",
      "BgpAsn": "65534"
    }
  ]
}
```

- Untuk detail API, lihat [DescribeCustomerGateways](#) di Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini menjelaskan gateway pelanggan yang ditentukan.

```
Get-EC2CustomerGateway -CustomerGatewayId cgw-1a2b3c4d
```

Output:

```
BgpAsn           : 65534
CustomerGatewayId : cgw-1a2b3c4d
IpAddress        : 203.0.113.12
State            : available
Tags             : {}
Type             : ipsec.1
```

Contoh 2: Contoh ini menjelaskan gateway pelanggan yang statusnya tertunda atau tersedia.

```
$filter = New-Object Amazon.EC2.Model.Filter
$filter.Name = "state"
$filter.Values = @( "pending", "available" )

Get-EC2CustomerGateway -Filter $filter
```

Contoh 3: Contoh ini menjelaskan semua gateway pelanggan Anda.

```
Get-EC2CustomerGateway
```

- Untuk detail API, lihat [DescribeCustomerGateways](#) di Referensi AWS Tools for PowerShell Cmdlet.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **DescribeDhcpOptions** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `DescribeDhcpOptions`.

CLI

AWS CLI

Contoh 1: Untuk menjelaskan opsi DHCP Anda

`describe-dhcp-options` Contoh berikut mengambil rincian tentang opsi DHCP Anda.

```
aws ec2 describe-dhcp-options
```

Output:

```
{
  "DhcpOptions": [
    {
      "DhcpConfigurations": [
        {
          "Key": "domain-name",
          "Values": [
            {
              "Value": "us-east-2.compute.internal"
            }
          ]
        },
        {
          "Key": "domain-name-servers",
          "Values": [
            {
              "Value": "AmazonProvidedDNS"
            }
          ]
        }
      ]
    },
    {
      "DhcpOptionsId": "dopt-19edf471",
      "OwnerId": "111122223333"
    }
  ],
  {
    "DhcpConfigurations": [
      {
        "Key": "domain-name",
        "Values": [
          {
            "Value": "us-east-2.compute.internal"
          }
        ]
      }
    ]
  }
}
```



```

        }
      ]
    },
    {
      "Key": "domain-name-servers",
      "Values": [
        {
          "Value": "AmazonProvidedDNS"
        }
      ]
    }
  ],
  "DhcpOptionsId": "dopt-fEXAMPLE",
  "OwnerId": "111122223333"
}
]
}

```

Untuk informasi selengkapnya, lihat [Bekerja dengan Set Opsi DHCP](#) di Panduan AWS Pengguna VPC.

Contoh 2: Untuk menjelaskan opsi DHCP Anda dan memfilter output

`describe-dhcp-options` Contoh berikut menjelaskan opsi DHCP Anda dan menggunakan filter untuk mengembalikan hanya opsi DHCP yang dimiliki `example.com` untuk server nama domain. Contoh menggunakan `--query` parameter untuk menampilkan hanya informasi konfigurasi dan ID dalam output.

```

aws ec2 describe-dhcp-options \
  --filters Name=key,Values=domain-name-servers Name=value,Values=example.com \
  --query "DhcpOptions[*].[DhcpConfigurations,DhcpOptionsId]"

```

Output:

```

[
  [
    [
      {
        "Key": "domain-name",
        "Values": [
          {
            "Value": "example.com"
          }
        ]
      }
    ]
  ]
]

```

```

    }
  ],
},
{
  "Key": "domain-name-servers",
  "Values": [
    {
      "Value": "172.16.16.16"
    }
  ]
}
],
"dopt-001122334455667ab"
]
]

```

Untuk informasi selengkapnya, lihat [Bekerja dengan Set Opsi DHCP](#) di Panduan AWS Pengguna VPC.

- Untuk detail API, lihat [DescribeDhcpOptions](#) di Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini mencantumkan set opsi DHCP Anda.

```
Get-EC2DhcpOption
```

Output:

DhcpConfigurations	DhcpOptionsId	Tag
{domain-name, domain-name-servers}	dopt-1a2b3c4d	{}
{domain-name, domain-name-servers}	dopt-2a3b4c5d	{}
{domain-name-servers}	dopt-3a4b5c6d	{}

Contoh 2: Contoh ini mendapatkan detail konfigurasi untuk set opsi DHCP yang ditentukan.

```
(Get-EC2DhcpOption -DhcpOptionsId dopt-1a2b3c4d).DhcpConfigurations
```

Output:

Key	Values
---	-----
domain-name	{abc.local}
domain-name-servers	{10.0.0.101, 10.0.0.102}

- Untuk detail API, lihat [DescribeDhcpOptions](#) di Referensi AWS Tools for PowerShell Cmdlet.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **DescribeFlowLogs** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `DescribeFlowLogs`.

CLI

AWS CLI

Contoh 1: Untuk mendeskripsikan semua log aliran Anda

`describe-flow-logs` Contoh berikut menampilkan detail untuk semua log aliran Anda.

```
aws ec2 describe-flow-logs
```

Output:

```
{
  "FlowLogs": [
    {
      "CreationTime": "2018-02-21T13:22:12.644Z",
      "DeliverLogsPermissionArn": "arn:aws:iam::123456789012:role/flow-logs-role",
      "DeliverLogsStatus": "SUCCESS",
      "FlowLogId": "fl-aabbccdd112233445",
      "MaxAggregationInterval": 600,
      "FlowLogStatus": "ACTIVE",
      "LogGroupName": "FlowLogGroup",
      "ResourceId": "subnet-12345678901234567",
      "TrafficType": "ALL",
      "LogDestinationType": "cloud-watch-logs",
    }
  ]
}
```

```

        "LogFormat": "${version} ${account-id} ${interface-id} ${srcaddr}
${dstaddr} ${srcport} ${dstport} ${protocol} ${packets} ${bytes} ${start} ${end}
${action} ${log-status}"
    },
    {
        "CreationTime": "2020-02-04T15:22:29.986Z",
        "DeliverLogsStatus": "SUCCESS",
        "FlowLogId": "fl-01234567890123456",
        "MaxAggregationInterval": 60,
        "FlowLogStatus": "ACTIVE",
        "ResourceId": "vpc-00112233445566778",
        "TrafficType": "ACCEPT",
        "LogDestinationType": "s3",
        "LogDestination": "arn:aws:s3:::my-flow-log-bucket/custom",
        "LogFormat": "${version} ${vpc-id} ${subnet-id} ${instance-id}
${interface-id} ${account-id} ${type} ${srcaddr} ${dstaddr} ${srcport}
${dstport} ${pkt-srcaddr} ${pkt-dstaddr} ${protocol} ${bytes} ${packets}
${start} ${end} ${action} ${tcp-flags} ${log-status}"
    }
]
}

```

Contoh 2: Untuk mendeskripsikan subset dari log aliran Anda

`describe-flow-logs` Contoh berikut menggunakan filter untuk menampilkan detail hanya untuk log aliran yang ada di grup log yang ditentukan di Amazon CloudWatch Logs.

```
aws ec2 describe-flow-logs \
  --filter "Name=log-group-name,Values=MyFlowLogs"
```

- Untuk detail API, lihat [DescribeFlowLogs](#) di Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini menjelaskan satu atau lebih flow log dengan tipe tujuan log 's3'

```
Get-EC2FlowLog -Filter @{Name="log-destination-type";Values="s3"}
```

Output:

```
CreationTime      : 2/25/2019 9:07:36 PM
DeliverLogsErrorMessage :
DeliverLogsPermissionArn :
DeliverLogsStatus : SUCCESS
FlowLogId        : f1-01b2e3d45f67f8901
FlowLogStatus    : ACTIVE
LogDestination   : arn:aws:s3:::my-bucket-dd-tata
LogDestinationType : s3
LogGroupName     :
ResourceId       : eni-01d2dda3456b7e890
TrafficType      : ALL
```

- Untuk detail API, lihat [DescribeFlowLogs](#) di Referensi AWS Tools for PowerShell Cmdlet.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **DescribeHostReservationOfferings** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `DescribeHostReservationOfferings`.

CLI

AWS CLI

Untuk menjelaskan penawaran Reservasi Tuan Rumah Khusus

Contoh ini menjelaskan Reservasi Tuan Rumah Khusus untuk keluarga instans M4 yang tersedia untuk dibeli.

Perintah:

```
aws ec2 describe-host-reservation-offerings --filter Name=instance-family,Values=m4
```

Output:

```
{
  "OfferingSet": [
```

```
{
  "HourlyPrice": "1.499",
  "OfferingId": "hro-03f707bf363b6b324",
  "InstanceFamily": "m4",
  "PaymentOption": "NoUpfront",
  "UpfrontPrice": "0.000",
  "Duration": 31536000
},
{
  "HourlyPrice": "1.045",
  "OfferingId": "hro-0ef9181cabdef7a02",
  "InstanceFamily": "m4",
  "PaymentOption": "NoUpfront",
  "UpfrontPrice": "0.000",
  "Duration": 94608000
},
{
  "HourlyPrice": "0.714",
  "OfferingId": "hro-04567a15500b92a51",
  "InstanceFamily": "m4",
  "PaymentOption": "PartialUpfront",
  "UpfrontPrice": "6254.000",
  "Duration": 31536000
},
{
  "HourlyPrice": "0.484",
  "OfferingId": "hro-0d5d7a9d23ed7fbfe",
  "InstanceFamily": "m4",
  "PaymentOption": "PartialUpfront",
  "UpfrontPrice": "12720.000",
  "Duration": 94608000
},
{
  "HourlyPrice": "0.000",
  "OfferingId": "hro-05da4108ca998c2e5",
  "InstanceFamily": "m4",
  "PaymentOption": "AllUpfront",
  "UpfrontPrice": "23913.000",
  "Duration": 94608000
},
{
  "HourlyPrice": "0.000",
  "OfferingId": "hro-0a9f9be3b95a3dc8f",
  "InstanceFamily": "m4",
```

```

        "PaymentOption": "AllUpfront",
        "UpfrontPrice": "12257.000",
        "Duration": 31536000
    }
]
}

```

- Untuk detail API, lihat [DescribeHostReservationOfferings](#) di Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini menjelaskan reservasi Host Khusus yang tersedia untuk dibeli untuk filter yang diberikan 'instance-family' di mana " PaymentOption NoUpfront

```

Get-EC2HostReservationOffering -Filter @{Name="instance-family";Values="m4"} |
Where-Object PaymentOption -eq NoUpfront

```

Output:

```

CurrencyCode      :
Duration          : 94608000
HourlyPrice       : 1.307
InstanceFamily    : m4
OfferingId        : hro-0c1f234567890d9ab
PaymentOption     : NoUpfront
UpfrontPrice      : 0.000

CurrencyCode      :
Duration          : 31536000
HourlyPrice       : 1.830
InstanceFamily    : m4
OfferingId        : hro-04ad12aaaf34b5a67
PaymentOption     : NoUpfront
UpfrontPrice      : 0.000

```

- Untuk detail API, lihat [DescribeHostReservationOfferings](#) di Referensi AWS Tools for PowerShell Cmdlet.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **DescribeHosts** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `DescribeHosts`.

CLI

AWS CLI

Untuk melihat detail tentang Host Khusus

`describe-hosts` Contoh berikut menampilkan detail untuk Host available Khusus di AWS akun Anda.

```
aws ec2 describe-hosts --filter "Name=state,Values=available"
```

Output:

```
{
  "Hosts": [
    {
      "HostId": "h-07879acf49EXAMPLE",
      "Tags": [
        {
          "Value": "production",
          "Key": "purpose"
        }
      ],
      "HostProperties": {
        "Cores": 48,
        "TotalVCpus": 96,
        "InstanceType": "m5.large",
        "Sockets": 2
      },
      "Instances": [],
      "State": "available",
      "AvailabilityZone": "eu-west-1a",
      "AvailableCapacity": {
        "AvailableInstanceCapacity": [
```



```

        {
            "AvailableCapacity": 48,
            "InstanceType": "m5.large",
            "TotalCapacity": 48
        }
    ],
    "AvailableVCpus": 96
},
"HostRecovery": "on",
"AllocationTime": "2019-08-19T08:57:44.000Z",
"AutoPlacement": "off"
}
]
}

```

Untuk informasi selengkapnya, lihat [Melihat Host Khusus](#) di Panduan Pengguna Amazon Elastic Compute Cloud untuk Instans Linux.

- Untuk detail API, lihat [DescribeHosts](#) di Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini mengembalikan rincian host EC2

```
Get-EC2Host
```

Output:

```

AllocationTime      : 3/23/2019 4:55:22 PM
AutoPlacement       : off
AvailabilityZone     : eu-west-1b
AvailableCapacity   : Amazon.EC2.Model.AvailableCapacity
ClientToken         :
HostId              : h-01e23f4cd567890f1
HostProperties       : Amazon.EC2.Model.HostProperties
HostReservationId   :
Instances           : {}
ReleaseTime         : 1/1/0001 12:00:00 AM
State               : available
Tags                : {}

```

Contoh 2: Contoh ini menanyakan host AvailableInstanceCapacity h-01e23f4cd567899f1

```
Get-EC2Host -HostId h-01e23f4cd567899f1 | Select-Object -ExpandProperty
  AvailableCapacity | Select-Object -expand AvailableInstanceCapacity
```

Output:

```
AvailableCapacity InstanceType TotalCapacity
-----
11                m4.xlarge    11
```

- Untuk detail API, lihat [DescribeHosts](#) di Referensi AWS Tools for PowerShell Cmdlet.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **DescribeIamInstanceProfileAssociations** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `DescribeIamInstanceProfileAssociations`.

Contoh tindakan adalah kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Anda dapat melihat tindakan ini dalam konteks dalam contoh kode berikut:

- [Membangun dan mengelola layanan yang tangguh](#)

.NET

AWS SDK for .NET

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
/// <summary>
```

```
/// Get the instance profile association data for an instance.
/// </summary>
/// <param name="instanceId">The Id of the instance.</param>
/// <returns>Instance profile associations data.</returns>
public async Task<IamInstanceProfileAssociation> GetInstanceProfile(string
instanceId)
{
    var response = await
    _amazonEc2.DescribeIamInstanceProfileAssociationsAsync(
        new DescribeIamInstanceProfileAssociationsRequest()
        {
            Filters = new List<Amazon.EC2.Model.Filter>()
            {
                new ("instance-id", new List<string>() { instanceId })
            },
        });
    return response.IamInstanceProfileAssociations[0];
}
```

- Untuk detail API, lihat [DescribeIamInstanceProfileAssociations](#) di Referensi AWS SDK for .NET API.

CLI

AWS CLI

Untuk menjelaskan kaitan profil instans IAM

Contoh ini menjelaskan semua kaitan profil instans IAM Anda.

Perintah:

```
aws ec2 describe-iam-instance-profile-associations
```

Output:

```
{
  "IamInstanceProfileAssociations": [
    {
      "InstanceId": "i-09eb09efa73ec1dee",
      "State": "associated",
```

```

    "AssociationId": "iip-assoc-0db249b1f25fa24b8",
    "IamInstanceProfile": {
      "Id": "AIPAJVQN4F5WVLGCJDRGM",
      "Arn": "arn:aws:iam::123456789012:instance-profile/admin-role"
    }
  },
  {
    "InstanceId": "i-0402909a2f4dffd14",
    "State": "associating",
    "AssociationId": "iip-assoc-0d1ec06278d29f44a",
    "IamInstanceProfile": {
      "Id": "AGJAJVQN4F5WVLGCJABCM",
      "Arn": "arn:aws:iam::123456789012:instance-profile/user1-role"
    }
  }
]
}

```

- Untuk detail API, lihat [DescribeIamInstanceProfileAssociations](#) di Referensi AWS CLI Perintah.

JavaScript

SDK untuk JavaScript (v3)

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```

const ec2Client = new EC2Client({});
const { IamInstanceProfileAssociations } = await ec2Client.send(
  new DescribeIamInstanceProfileAssociationsCommand({
    Filters: [
      { Name: "instance-id", Values: [state.targetInstance.InstanceId] },
    ],
  }),
);

```

- Untuk detail API, lihat [DescribeIamInstanceProfileAssociations](#) di Referensi AWS SDK for JavaScript API.

Python

SDK untuk Python (Boto3)

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
class AutoScaler:
    """
    Encapsulates Amazon EC2 Auto Scaling and EC2 management actions.
    """

    def __init__(
        self,
        resource_prefix,
        inst_type,
        ami_param,
        autoscaling_client,
        ec2_client,
        ssm_client,
        iam_client,
    ):
        """
        :param resource_prefix: The prefix for naming AWS resources that are
        created by this class.
        :param inst_type: The type of EC2 instance to create, such as t3.micro.
        :param ami_param: The Systems Manager parameter used to look up the AMI
        that is
                created.
        :param autoscaling_client: A Boto3 EC2 Auto Scaling client.
        :param ec2_client: A Boto3 EC2 client.
        :param ssm_client: A Boto3 Systems Manager client.
        :param iam_client: A Boto3 IAM client.
        """
        self.inst_type = inst_type
```

```
self.ami_param = ami_param
self.autoscaling_client = autoscaling_client
self.ec2_client = ec2_client
self.ssm_client = ssm_client
self.iam_client = iam_client
self.launch_template_name = f"{resource_prefix}-template"
self.group_name = f"{resource_prefix}-group"
self.instance_policy_name = f"{resource_prefix}-pol"
self.instance_role_name = f"{resource_prefix}-role"
self.instance_profile_name = f"{resource_prefix}-prof"
self.bad_creds_policy_name = f"{resource_prefix}-bc-pol"
self.bad_creds_role_name = f"{resource_prefix}-bc-role"
self.bad_creds_profile_name = f"{resource_prefix}-bc-prof"
self.key_pair_name = f"{resource_prefix}-key-pair"

def get_instance_profile(self, instance_id):
    """
    Gets data about the profile associated with an instance.

    :param instance_id: The ID of the instance to look up.
    :return: The profile data.
    """
    try:
        response =
self.ec2_client.describe_iam_instance_profile_associations(
            Filters=[{"Name": "instance-id", "Values": [instance_id]}]
        )
    except ClientError as err:
        raise AutoScalerError(
            f"Couldn't get instance profile association for instance
{instance_id}: {err}")
    else:
        return response["IamInstanceProfileAssociations"][0]
```

- Untuk detail API, lihat [DescribeIamInstanceProfileAssociations](#) di AWS SDK for Python (Boto3) Referensi API.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **DescribeIdFormat** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `DescribeIdFormat`.

CLI

AWS CLI

Contoh 1: Untuk mendeskripsikan format ID sumber daya

`describe-id-format` Contoh berikut menjelaskan format ID untuk grup keamanan.

```
aws ec2 describe-id-format \
  --resource security-group
```

Dalam contoh output berikut, `Deadline` nilai menunjukkan bahwa batas waktu untuk jenis sumber daya ini untuk secara permanen beralih dari format ID pendek ke format ID panjang berakhir pada 00:00 UTC pada 15 Agustus 2018.

```
{
  "Statuses": [
    {
      "Deadline": "2018-08-15T00:00:00.000Z",
      "Resource": "security-group",
      "UseLongIds": true
    }
  ]
}
```

Contoh 2: Untuk mendeskripsikan format ID untuk semua sumber daya

`describe-id-format` Contoh berikut menjelaskan format ID untuk semua jenis sumber daya. Semua jenis sumber daya yang mendukung format ID pendek dialihkan untuk menggunakan format ID panjang.

```
aws ec2 describe-id-format
```

- Untuk detail API, lihat [DescribeIdFormat](#) di Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini menjelaskan format ID untuk jenis sumber daya tertentu.

```
Get-EC2IdFormat -Resource instance
```

Output:

Resource	UseLongIds
-----	-----
instance	False

Contoh 2: Contoh ini menjelaskan format ID untuk semua jenis sumber daya yang mendukung ID yang lebih panjang.

```
Get-EC2IdFormat
```

Output:

Resource	UseLongIds
-----	-----
reservation	False
instance	False

- Untuk detail API, lihat [DescribeIdFormat](#) di Referensi AWS Tools for PowerShell Cmdlet.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **DescribeIdentityIdFormat** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `DescribeIdentityIdFormat`.

CLI

AWS CLI

Untuk menjelaskan format ID untuk peran IAM

`describe-identity-id-format` Contoh berikut menjelaskan format ID yang diterima oleh instance yang dibuat oleh peran IAM EC2Role di akun Anda AWS .

```
aws ec2 describe-identity-id-format \  
  --principal-arn arn:aws:iam::123456789012:role/my-iam-role \  
  --resource instance
```

Output berikut menunjukkan bahwa instance yang dibuat oleh peran ini menerima ID dalam format ID panjang.

```
{  
  "Statuses": [  
    {  
      "Deadline": "2016-12-15T00:00:00Z",  
      "Resource": "instance",  
      "UseLongIds": true  
    }  
  ]  
}
```

Untuk mendeskripsikan format ID untuk pengguna IAM

`describe-identity-id-format` Contoh berikut menjelaskan format ID yang diterima oleh snapshot yang dibuat oleh pengguna IAM AdminUser di akun Anda AWS .

```
aws ec2 describe-identity-id-format \  
  --principal-arn arn:aws:iam::123456789012:user/AdminUser \  
  --resource snapshot
```

Output menunjukkan bahwa snapshot yang dibuat oleh pengguna ini menerima ID dalam format ID panjang.

```
{  
  "Statuses": [  
    {  
      "Deadline": "2016-12-15T00:00:00Z",
```

```

        "Resource": "snapshot",
        "UseLongIds": true
    }
]
}

```

- Untuk detail API, lihat [DescribeIdentityIdFormat](#) di Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini mengembalikan format ID untuk sumber 'image' untuk peran yang diberikan

```

Get-EC2IdentityIdFormat -PrincipalArn arn:aws:iam::123456789511:role/JDBC -
Resource image

```

Output:

```

Deadline                Resource UseLongIds
-----                -
8/2/2018 11:30:00 PM image    True

```

- Untuk detail API, lihat [DescribeIdentityIdFormat](#) di Referensi AWS Tools for PowerShell Cmdlet.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **DescribeImageAttribute** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `DescribeImageAttribute`.

CLI

AWS CLI

Untuk menjelaskan izin peluncuran untuk AMI

Contoh ini menjelaskan izin peluncuran untuk AMI yang ditentukan.

Perintah:

```
aws ec2 describe-image-attribute --image-id ami-5731123e --attribute
  launchPermission
```

Output:

```
{
  "LaunchPermissions": [
    {
      "UserId": "123456789012"
    }
  ],
  "ImageId": "ami-5731123e",
}
```

Untuk mendeskripsikan kode produk untuk AMI

Contoh ini menjelaskan kode produk untuk AMI yang ditentukan. Perhatikan bahwa AMI ini tidak memiliki kode produk.

Perintah:

```
aws ec2 describe-image-attribute --image-id ami-5731123e --attribute productCodes
```

Output:

```
{
  "ProductCodes": [],
  "ImageId": "ami-5731123e",
}
```

- Untuk detail API, lihat [DescribeImageAttribute](#) di Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini mendapatkan deskripsi untuk AMI yang ditentukan.

```
Get-EC2ImageAttribute -ImageId ami-12345678 -Attribute description
```

Output:

```
BlockDeviceMappings : {}  
Description          : My image description  
ImageId             : ami-12345678  
KernelId           :  
LaunchPermissions   : {}  
ProductCodes        : {}  
RamdiskId           :  
SriovNetSupport     :
```

Contoh 2: Contoh ini mendapatkan izin peluncuran untuk AMI yang ditentukan.

```
Get-EC2ImageAttribute -ImageId ami-12345678 -Attribute launchPermission
```

Output:

```
BlockDeviceMappings : {}  
Description          :  
ImageId             : ami-12345678  
KernelId           :  
LaunchPermissions   : {all}  
ProductCodes        : {}  
RamdiskId           :  
SriovNetSupport     :
```

Contoh 3: Contoh ini menguji apakah jaringan yang ditingkatkan diaktifkan.

```
Get-EC2ImageAttribute -ImageId ami-12345678 -Attribute sriovNetSupport
```

Output:

```
BlockDeviceMappings : {}  
Description          :  
ImageId             : ami-12345678  
KernelId           :  
LaunchPermissions   : {}  
ProductCodes        : {}  
RamdiskId           :
```

```
SriovNetSupport : simple
```

- Untuk detail API, lihat [DescribeImageAttribute](#) di Referensi AWS Tools for PowerShell Cmdlet.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **DescribeImages** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `DescribeImages`.

Contoh tindakan adalah kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Anda dapat melihat tindakan ini dalam konteks dalam contoh kode berikut:

- [Memulai instans](#)

Bash

AWS CLI dengan skrip Bash

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
#####  
# function ec2_describe_images  
#  
# This function describes one or more Amazon Elastic Compute Cloud (Amazon EC2)  
# images.  
#  
# Parameters:  
#     -i image_ids - A space-separated list of image IDs (optional).  
#     -h - Display help.  
#  
# And:  
#     0 - If successful.
```

```

#      1 - If it fails.
#####
function ec2_describe_images() {
    local image_ids response
    local option OPTARG # Required to use getopt command in a function.

    # bashsupport disable=BP5008
    function usage() {
        echo "function ec2_describe_images"
        echo "Describes one or more Amazon Elastic Compute Cloud (Amazon EC2)
images."
        echo "  -i image_ids - A space-separated list of image IDs (optional)."
        echo "  -h - Display help."
        echo ""
    }

    # Retrieve the calling parameters.
    while getopt "i:h" option; do
        case "${option}" in
            i) image_ids="${OPTARG}" ;;
            h)
                usage
                return 0
                ;;
            \?)
                echo "Invalid parameter"
                usage
                return 1
                ;;
        esac
    done
    export OPTIND=1

    local aws_cli_args=()

    if [[ -n "$image_ids" ]]; then
        # shellcheck disable=SC2206
        aws_cli_args+=("--image-ids" $image_ids)
    fi

    response=$(aws ec2 describe-images \
        "${aws_cli_args[@]}" \
        --query 'Images[*].[Description,Architecture,ImageId]' \
        --output text) || {

```

```

aws_cli_error_log ${?}
errecho "ERROR: AWS reports describe-images operation failed.$response"
return 1
}

echo "$response"

return 0
}

```

Fungsi utilitas yang digunakan dalam contoh ini.

```

#####
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
#####
function errecho() {
    printf "%s\n" "$*" 1>&2
}

#####
# function aws_cli_error_log()
#
# This function is used to log the error messages from the AWS CLI.
#
# The function expects the following argument:
#     $1 - The error code returned by the AWS CLI.
#
# Returns:
#     0: - Success.
#
#####
function aws_cli_error_log() {
    local err_code=$1
    errecho "Error code : $err_code"
    if [ "$err_code" == 1 ]; then
        errecho " One or more S3 transfers failed."
    elif [ "$err_code" == 2 ]; then
        errecho " Command line failed to parse."
    elif [ "$err_code" == 130 ]; then
        errecho " Process received SIGINT."
    fi
}

```

```
elif [ "$err_code" == 252 ]; then
    errecho " Command syntax invalid."
elif [ "$err_code" == 253 ]; then
    errecho " The system environment or configuration was invalid."
elif [ "$err_code" == 254 ]; then
    errecho " The service returned an error."
elif [ "$err_code" == 255 ]; then
    errecho " 255 is a catch-all error."
fi

return 0
}
```

- Untuk detail API, lihat [DescribeImages](#) di Referensi AWS CLI Perintah.

CLI

AWS CLI

Contoh 1: Untuk menjelaskan AMI

Contoh `describe-images` berikut menjelaskan AMI tertentu di Wilayah tertentu.

```
aws ec2 describe-images \
  --region us-east-1 \
  --image-ids ami-1234567890EXAMPLE
```

Output:

```
{
  "Images": [
    {
      "VirtualizationType": "hvm",
      "Description": "Provided by Red Hat, Inc.",
      "PlatformDetails": "Red Hat Enterprise Linux",
      "EnaSupport": true,
      "Hypervisor": "xen",
      "State": "available",
      "SriovNetSupport": "simple",
      "ImageId": "ami-1234567890EXAMPLE",
      "UsageOperation": "RunInstances:0010",
      "BlockDeviceMappings": [
```



```

        {
            "DeviceName": "/dev/sda1",
            "Ebs": {
                "SnapshotId": "snap-111222333444aaabb",
                "DeleteOnTermination": true,
                "VolumeType": "gp2",
                "VolumeSize": 10,
                "Encrypted": false
            }
        },
        "Architecture": "x86_64",
        "ImageLocation": "123456789012/RHEL-8.0.0_HVM-20190618-x86_64-1-
Hourly2-GP2",
        "RootDeviceType": "ebs",
        "OwnerId": "123456789012",
        "RootDeviceName": "/dev/sda1",
        "CreationDate": "2019-05-10T13:17:12.000Z",
        "Public": true,
        "ImageType": "machine",
        "Name": "RHEL-8.0.0_HVM-20190618-x86_64-1-Hourly2-GP2"
    }
]
}

```

Untuk informasi selengkapnya, lihat [Amazon Machine Image \(AMI\)](#) di Panduan Pengguna Amazon EC2.

Contoh 2: Untuk menjelaskan AMI berdasarkan filter

Contoh `describe-images` berikut menjelaskan AMI Windows yang disediakan oleh Amazon yang didukung oleh Amazon EBS.

```

aws ec2 describe-images \
  --owners amazon \
  --filters "Name=platform,Values=windows" "Name=root-device-type,Values=ebs"

```

Untuk contoh output `describe-images`, lihat Contoh 1.

Untuk contoh tambahan menggunakan filter, lihat [Mencantumkan dan memfilter sumber daya Anda](#) di Panduan Pengguna Amazon EC2.

Contoh 3: Untuk menjelaskan AMI berdasarkan tanda

Contoh `describe-images` berikut menjelaskan semua AMI yang memiliki tanda `Type=Custom`. Contoh tersebut menggunakan parameter `--query` untuk hanya menampilkan ID AMI.

```
aws ec2 describe-images \  
  --filters "Name=tag:Type,Values=Custom" \  
  --query 'Images[*].[ImageId]' \  
  --output text
```

Output:

```
ami-1234567890EXAMPLE  
ami-0abcdef1234567890
```

Untuk contoh tambahan menggunakan filter tanda, lihat [Bekerja dengan tanda](#) di Panduan Pengguna Amazon EC2.

- Untuk detail API, lihat [DescribeImages](#) di Referensi AWS CLI Perintah.

JavaScript

SDK untuk JavaScript (v3)

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
import { paginateDescribeImages } from "@aws-sdk/client-ec2";  
  
import { client } from "../libs/client.js";  
  
// List at least the first i386 image available for EC2 instances.  
export const main = async () => {  
  // The paginate function is a wrapper around the base command.  
  const paginator = paginateDescribeImages(  
    // Without limiting the page size, this call can take a long time. pageSize  
    // is just sugar for  
    // the MaxResults property in the base command.  
    { client, pageSize: 25 },
```

```
{
  // There are almost 70,000 images available. Be specific with your
  filtering
  // to increase efficiency.
  // See https://docs.aws.amazon.com/AWSJavaScriptSDK/v3/latest/clients/
  client-ec2/interfaces/describeimagescommandinput.html#filters
  Filters: [{ Name: "architecture", Values: ["x86_64"] }],
},
);

try {
  const arm64Images = [];
  for await (const page of paginator) {
    if (page.Images.length) {
      arm64Images.push(...page.Images);
      // Once we have at least 1 result, we can stop.
      if (arm64Images.length >= 1) {
        break;
      }
    }
  }
  console.log(arm64Images);
} catch (err) {
  console.error(err);
}
};
```

- Untuk detail API, lihat [DescribeImages](#) di Referensi AWS SDK for JavaScript API.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini menjelaskan AMI yang ditentukan.

```
Get-EC2Image -ImageId ami-12345678
```

Output:

```
Architecture      : x86_64
BlockDeviceMappings : {/dev/xvda}
```

```
CreationDate      : 2014-10-20T00:56:28.000Z
Description       : My image
Hypervisor        : xen
ImageId           : ami-12345678
ImageLocation     : 123456789012/my-image
ImageOwnerAlias   :
ImageType         : machine
KernelId          :
Name              : my-image
OwnerId           : 123456789012
Platform          :
ProductCodes      : {}
Public            : False
RamdiskId         :
RootDeviceName    : /dev/xvda
RootDeviceType    : ebs
SriovNetSupport   : simple
State             : available
StateReason       :
Tags              : {Name}
VirtualizationType : hvm
```

Contoh 2: Contoh ini menjelaskan AMI yang Anda miliki.

```
Get-EC2Image -owner self
```

Contoh 3: Contoh ini menjelaskan AMI publik yang menjalankan Microsoft Windows Server.

```
Get-EC2Image -Filter @{ Name="platform"; Values="windows" }
```

Contoh 4: Contoh ini menjelaskan semua AMI publik di wilayah 'us-west-2'.

```
Get-EC2Image -Region us-west-2
```

- Untuk detail API, lihat [DescribeImages](#) di Referensi AWS Tools for PowerShell Cmdlet.

Python

SDK untuk Python (Boto3)

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
class InstanceWrapper:
    """Encapsulates Amazon Elastic Compute Cloud (Amazon EC2) instance
    actions."""

    def __init__(self, ec2_resource, instance=None):
        """
        :param ec2_resource: A Boto3 Amazon EC2 resource. This high-level
        resource
                                is used to create additional high-level objects
                                that wrap low-level Amazon EC2 service actions.
        :param instance: A Boto3 Instance object. This is a high-level object
        that
                                wraps instance actions.
        """
        self.ec2_resource = ec2_resource
        self.instance = instance

    @classmethod
    def from_resource(cls):
        ec2_resource = boto3.resource("ec2")
        return cls(ec2_resource)

    def get_images(self, image_ids):
        """
        Gets information about Amazon Machine Images (AMIs) from a list of AMI
        IDs.

        :param image_ids: The list of AMIs to look up.
        :return: A list of Boto3 Image objects that represent the requested AMIs.
        """
        try:
```

```

        images = list(self.ec2_resource.images.filter(ImageIds=image_ids))
    except ClientError as err:
        logger.error(
            "Couldn't get images. Here's why: %s: %s",
            err.response["Error"]["Code"],
            err.response["Error"]["Message"],
        )
        raise
    else:
        return images

```

- Untuk detail API, lihat [DescribeImages](#) di AWS SDK for Python (Boto3) Referensi API.

Rust

SDK untuk Rust

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```

// A simple Rust program to list all Amazon images in the currently configured
// region.
#[tokio::main]
async fn main() -> Result<(), Error> {
    let config = aws_config::load_from_env().await;
    let client = Client::new(&config);

    let resp = client.describe_images().owners("amazon").send().await?;

    println!("AWS SDK for Rust v{}", PKG_VERSION);
    println!("Describing Amazon Machine Images (AMIs):");

    let mut images: Vec<_> = resp
        .images()
        .iter()
        .filter(|i| {
            i.description()

```

```
        .filter(|i| i.contains("Amazon Linux AMI 2023"))
        .is_some()
    })
    .collect();
images.sort_by(|a, b| a.description.cmp(&b.description));

if images.is_empty() {
    println!("No images found.");
    return Ok(());
}

for image in images {
    let id = image.image_id().unwrap_or_default();
    let description = image.description().unwrap_or_default();

    println!("{id}: {description}");
}

Ok(())
}
```

- Untuk detail API, lihat [DescribeImages](#) referensi AWS SDK for Rust API.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **DescribeImportImageTasks** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `DescribeImportImageTasks`.

CLI

AWS CLI

Untuk memantau tugas impor gambar

`describe-import-image-tasks` Contoh berikut memeriksa status tugas gambar impor yang ditentukan.

```
aws ec2 describe-import-image-tasks \
```

```
--import-task-ids import-ami-1234567890abcdef0
```

Output untuk tugas impor gambar yang sedang berlangsung.

```
{
  "ImportImageTasks": [
    {
      "ImportTaskId": "import-ami-1234567890abcdef0",
      "Progress": "28",
      "SnapshotDetails": [
        {
          "DiskImageSize": 705638400.0,
          "Format": "ova",
          "Status": "completed",
          "UserBucket": {
            "S3Bucket": "my-import-bucket",
            "S3Key": "vms/my-server-vm.ova"
          }
        }
      ],
      "Status": "active",
      "StatusMessage": "converting"
    }
  ]
}
```

Output untuk tugas gambar impor yang selesai. ID AMI yang dihasilkan disediakan oleh `ImageId`.

```
{
  "ImportImageTasks": [
    {
      "ImportTaskId": "import-ami-1234567890abcdef0",
      "ImageId": "ami-1234567890abcdef0",
      "SnapshotDetails": [
        {
          "DiskImageSize": 705638400.0,
          "Format": "ova",
          "SnapshotId": "snap-1234567890abcdef0",
          "Status": "completed",
          "UserBucket": {
            "S3Bucket": "my-import-bucket",
            "S3Key": "vms/my-server-vm.ova"
          }
        }
      ]
    }
  ]
}
```



```

        }
      ],
      "Status": "completed"
    }
  ]
}

```

- Untuk detail API, lihat [DescribeImportImageTasks](#) di Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini menjelaskan tugas impor gambar yang ditentukan.

```
Get-EC2ImportImageTask -ImportTaskId import-ami-hgfedcba
```

Output:

```

Architecture      : x86_64
Description       : Windows Image 2
Hypervisor        :
ImageId           : ami-1a2b3c4d
ImportTaskId      : import-ami-hgfedcba
LicenseType       : AWS
Platform          : Windows
Progress          :
SnapshotDetails   : {/dev/sda1}
Status            : completed
StatusMessage     :

```

Contoh 2: Contoh ini menjelaskan semua tugas impor gambar Anda.

```
Get-EC2ImportImageTask
```

Output:

```

Architecture      :
Description       : Windows Image 1
Hypervisor        :

```

```

ImageId      :
ImportTaskId : import-ami-abcdefgh
LicenseType  : AWS
Platform     : Windows
Progress     :
SnapshotDetails : {}
Status       : deleted
StatusMessage : User initiated task cancelation

Architecture : x86_64
Description   : Windows Image 2
Hypervisor    :
ImageId       : ami-1a2b3c4d
ImportTaskId  : import-ami-hgfedcba
LicenseType   : AWS
Platform      : Windows
Progress      :
SnapshotDetails : {/dev/sda1}
Status        : completed
StatusMessage :

```

- Untuk detail API, lihat [DescribeImportImageTasks](#) di Referensi AWS Tools for PowerShell Cmdlet.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **DescribeImportSnapshotTasks** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `DescribeImportSnapshotTasks`.

CLI

AWS CLI

Untuk memantau tugas snapshot impor

`describe-import-snapshot-tasks` Contoh berikut memeriksa status tugas snapshot impor yang ditentukan.

```
aws ec2 describe-import-snapshot-tasks \
```

```
--import-task-ids import-snap-1234567890abcdef0
```

Output untuk tugas snapshot impor yang sedang berlangsung:

```
{
  "ImportSnapshotTasks": [
    {
      "Description": "My server VMDK",
      "ImportTaskId": "import-snap-1234567890abcdef0",
      "SnapshotTaskDetail": {
        "Description": "My server VMDK",
        "DiskImageSize": "705638400.0",
        "Format": "VMDK",
        "Progress": "42",
        "Status": "active",
        "StatusMessage": "downloading/converting",
        "UserBucket": {
          "S3Bucket": "my-import-bucket",
          "S3Key": "vms/my-server-vm.vmdk"
        }
      }
    }
  ]
}
```

Output untuk tugas snapshot impor yang selesai. ID snapshot yang dihasilkan disediakan oleh `SnapshotId`.

```
{
  "ImportSnapshotTasks": [
    {
      "Description": "My server VMDK",
      "ImportTaskId": "import-snap-1234567890abcdef0",
      "SnapshotTaskDetail": {
        "Description": "My server VMDK",
        "DiskImageSize": "705638400.0",
        "Format": "VMDK",
        "SnapshotId": "snap-1234567890abcdef0"
        "Status": "completed",
        "UserBucket": {
          "S3Bucket": "my-import-bucket",
          "S3Key": "vms/my-server-vm.vmdk"
        }
      }
    }
  ]
}
```

```

    }
  }
]
}

```

- Untuk detail API, lihat [DescribeImportSnapshotTasks](#) di Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini menjelaskan tugas impor snapshot yang ditentukan.

```
Get-EC2ImportSnapshotTask -ImportTaskId import-snap-abcdefgh
```

Output:

Description	ImportTaskId	SnapshotTaskDetail
-----	-----	-----
Disk Image Import 1	import-snap-abcdefgh	Amazon.EC2.Model.SnapshotTaskDetail

Contoh 2: Contoh ini menjelaskan semua tugas impor snapshot Anda.

```
Get-EC2ImportSnapshotTask
```

Output:

Description	ImportTaskId	SnapshotTaskDetail
-----	-----	-----
Disk Image Import 1	import-snap-abcdefgh	Amazon.EC2.Model.SnapshotTaskDetail
Disk Image Import 2	import-snap-hgfedcba	Amazon.EC2.Model.SnapshotTaskDetail

- Untuk detail API, lihat [DescribeImportSnapshotTasks](#) di Referensi AWS Tools for PowerShell Cmdlet.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **DescribeInstanceAttribute** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `DescribeInstanceAttribute`.

CLI

AWS CLI

Untuk menggambarkan jenis instance

Contoh ini menjelaskan jenis instance dari instance yang ditentukan.

Perintah:

```
aws ec2 describe-instance-attribute --instance-id i-1234567890abcdef0 --attribute instanceType
```

Output:

```
{
  "InstanceId": "i-1234567890abcdef0"
  "InstanceType": {
    "Value": "t1.micro"
  }
}
```

Untuk menggambarkan `disableApiTermination` atribut

Contoh ini menjelaskan `disableApiTermination` atribut dari contoh yang ditentukan.

Perintah:

```
aws ec2 describe-instance-attribute --instance-id i-1234567890abcdef0 --attribute disableApiTermination
```

Output:

```
{
  "InstanceId": "i-1234567890abcdef0"
  "DisableApiTermination": {
    "Value": "false"
  }
}
```

Untuk mendeskripsikan pemetaan perangkat blok untuk sebuah instance

Contoh ini menjelaskan `blockDeviceMapping` atribut dari contoh yang ditentukan.

Perintah:

```
aws ec2 describe-instance-attribute --instance-id i-1234567890abcdef0 --attribute
blockDeviceMapping
```

Output:

```
{
  "InstanceId": "i-1234567890abcdef0"
  "BlockDeviceMappings": [
    {
      "DeviceName": "/dev/sda1",
      "Ebs": {
        "Status": "attached",
        "DeleteOnTermination": true,
        "VolumeId": "vol-049df61146c4d7901",
        "AttachTime": "2013-05-17T22:42:34.000Z"
      }
    },
    {
      "DeviceName": "/dev/sdf",
      "Ebs": {
        "Status": "attached",
        "DeleteOnTermination": false,
        "VolumeId": "vol-049df61146c4d7901",
        "AttachTime": "2013-09-10T23:07:00.000Z"
      }
    }
  ],
}
```

- Untuk detail API, lihat [DescribeInstanceAttribute](#) di Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini menjelaskan jenis instance dari instance yang ditentukan.

```
Get-EC2InstanceAttribute -InstanceId i-12345678 -Attribute instanceType
```

Output:

```
InstanceType           : t2.micro
```

Contoh 2: Contoh ini menjelaskan apakah jaringan yang ditingkatkan diaktifkan untuk instance tertentu.

```
Get-EC2InstanceAttribute -InstanceId i-12345678 -Attribute sriovNetSupport
```

Output:

```
SriovNetSupport        : simple
```

Contoh 3: Contoh ini menjelaskan grup keamanan untuk instance tertentu.

```
(Get-EC2InstanceAttribute -InstanceId i-12345678 -Attribute groupSet).Groups
```

Output:

```
GroupId
-----
sg-12345678
sg-45678901
```

Contoh 4: Contoh ini menjelaskan apakah optimasi EBS diaktifkan untuk instance tertentu.

```
Get-EC2InstanceAttribute -InstanceId i-12345678 -Attribute ebsOptimized
```

Output:

```
EbsOptimized : False
```

Contoh 5: Contoh ini menjelaskan atribut `disableApiTermination` " dari contoh yang ditentukan.

```
Get-EC2InstanceAttribute -InstanceId i-12345678 -Attribute disableApiTermination
```

Output:

```
DisableApiTermination : False
```

Contoh 6: Contoh ini menjelaskan atribut 'instanceInitiatedShutdownPerilaku' dari contoh yang ditentukan.

```
Get-EC2InstanceAttribute -InstanceId i-12345678 -Attribute  
instanceInitiatedShutdownBehavior
```

Output:

```
InstanceInitiatedShutdownBehavior : stop
```

- Untuk detail API, lihat [DescribeInstanceAttribute](#) di Referensi AWS Tools for PowerShell Cmdlet.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **DescribeInstanceStatus** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `DescribeInstanceStatus`.

CLI

AWS CLI

Untuk menjelaskan status instans

Contoh `describe-instance-status` berikut menjelaskan status saat ini dari instans tertentu.

```
aws ec2 describe-instance-status \  
  --instance-ids i-1234567890abcdef0
```

Output:

```
{  
  "InstanceStatuses": [  
    {  
      "InstanceId": "i-1234567890abcdef0",  
      "InstanceState": {  
        "Code": 16,  
        "Name": "running"  
      },  
      "AvailabilityZone": "us-east-1d",  
      "SystemStatus": {  
        "Status": "ok",  
        "Details": [  
          {  
            "Status": "passed",  
            "Name": "reachability"  
          }  
        ]  
      },  
      "InstanceStatus": {  
        "Status": "ok",  
        "Details": [  
          {  
            "Status": "passed",  
            "Name": "reachability"  
          }  
        ]  
      }  
    }  
  ]  
}
```

Untuk informasi selengkapnya, lihat [Pantau status instans Anda](#) di Panduan Pengguna Amazon EC2.

- Untuk detail API, lihat [DescribeInstanceStatus](#) di Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini menjelaskan status instance yang ditentukan.

```
Get-EC2InstanceStatus -InstanceId i-12345678
```

Output:

```
AvailabilityZone : us-west-2a
Events           : {}
InstanceId       : i-12345678
InstanceState    : Amazon.EC2.Model.InstanceState
Status          : Amazon.EC2.Model.InstanceStatusSummary
SystemStatus     : Amazon.EC2.Model.InstanceStatusSummary
```

```
$status = Get-EC2InstanceStatus -InstanceId i-12345678
$status.InstanceState
```

Output:

```
Code    Name
----    -
16      running
```

```
$status.Status
```

Output:

```
Details      Status
-----      -
{reachability} ok
```

```
$status.SystemStatus
```

Output:

```
Details      Status
-----      -
```

```
{reachability}    ok
```

- Untuk detail API, lihat [DescribeInstanceStatus](#) di Referensi AWS Tools for PowerShell Cmdlet.

Rust

SDK untuk Rust

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
async fn show_all_events(client: &Client) -> Result<(), Error> {
    let resp = client.describe_regions().send().await.unwrap();

    for region in resp.regions.unwrap_or_default() {
        let reg: &'static str =
Box::leak(Box::from(region.region_name().unwrap()));
        let region_provider =
RegionProviderChain::default_provider().or_else(reg);
        let config = aws_config::from_env().region(region_provider).load().await;
        let new_client = Client::new(&config);

        let resp = new_client.describe_instance_status().send().await;

        println!("Instances in region {}: ", reg);
        println!();

        for status in resp.unwrap().instance_statuses() {
            println!(
                "  Events scheduled for instance ID: {}",
                status.instance_id().unwrap_or_default()
            );
            for event in status.events() {
                println!("    Event ID:      {}",
event.instance_event_id().unwrap());
                println!("    Description:  {}", event.description().unwrap());
                println!("    Event code:   {}", event.code().unwrap().as_ref());
            }
        }
    }
}
```

```
        println!();
    }
}
}
Ok(())
}
```

- Untuk detail API, lihat [DescribeInstanceStatus](#) referensi AWS SDK for Rust API.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **DescribeInstanceTypes** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `DescribeInstanceTypes`.

Contoh tindakan adalah kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Anda dapat melihat tindakan ini dalam konteks dalam contoh kode berikut:

- [Memulai instans](#)

.NET

AWS SDK for .NET

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
/// <summary>
/// Describe the instance types available.
/// </summary>
/// <returns>A list of instance type information.</returns>
public async Task<List<InstanceTypeInfo>>
DescribeInstanceTypes(ArchitectureValues architecture)
```

```

{
    var request = new DescribeInstanceTypesRequest();

    var filters = new List<Filter>
        { new Filter("processor-info.supported-architecture", new
List<string> { architecture.ToString() }) };
    filters.Add(new Filter("instance-type", new() { "*.micro", "*.small" }));

    request.Filters = filters;
    var instanceTypes = new List<InstanceTypeInfo>();

    var paginator = _amazonEC2.Paginators.DescribeInstanceTypes(request);
    await foreach (var instanceType in paginator.InstanceTypes)
    {
        instanceTypes.Add(instanceType);
    }
    return instanceTypes;
}

```

- Untuk detail API, lihat [DescribeInstanceTypes](#) di Referensi AWS SDK for .NET API.

Bash

AWS CLI dengan skrip Bash

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```

#####
# ec2_describe_instance_types
#
# This function describes EC2 instance types filtered by processor architecture
# and optionally by instance type. It takes the following arguments:
#
# -a, --architecture ARCHITECTURE  Specify the processor architecture (e.g.,
x86_64)
# -t, --type INSTANCE_TYPE          Comma-separated list of instance types (e.g.,
t2.micro)

```

```

# -h, --help                Show the usage help
#
# The function prints the instance type and supported architecture for each
# matching instance type.
#####
function ec2_describe_instance_types() {
    local architecture=""
    local instance_types=""

    # bashsupport disable=BP5008
    function usage() {
        echo "Usage: ec2_describe_instance_types [-a|--architecture ARCHITECTURE] [-t|--type INSTANCE_TYPE] [-h|--help]"
        echo "  -a, --architecture ARCHITECTURE  Specify the processor architecture (e.g., x86_64)"
        echo "  -t, --type INSTANCE_TYPE          Comma-separated list of instance types (e.g., t2.micro)"
        echo "  -h, --help                        Show this help message"
    }

    while [[ $# -gt 0 ]]; do
        case "$1" in
            -a | --architecture)
                architecture="$2"
                shift 2
                ;;
            -t | --type)
                instance_types="$2"
                shift 2
                ;;
            -h | --help)
                usage
                return 0
                ;;
            *)
                echo "Unknown argument: $1"
                return 1
                ;;
        esac
    done

    if [[ -z "$architecture" ]]; then
        errecho "Error: Architecture not specified."
        usage
    fi
}

```

```
    return 1
fi

if [[ -z "$instance_types" ]]; then
    errecho "Error: Instance type not specified."
    usage
    return 1
fi

local tmp_json_file="temp_ec2.json"
echo -n '['
{
    "Name": "processor-info.supported-architecture",
    "Values": [' >"$tmp_json_file"

local items
IFS=', ' read -ra items <<<"$architecture"
local array_size
array_size=${#items[@]}
for i in $(seq 0 $((array_size - 1))); do
    echo -n '"'"${items[$i]}"'"' >>"$tmp_json_file"
    if [[ $i -lt $((array_size - 1)) ]]; then
        echo -n ', ' >>"$tmp_json_file"
    fi
done
echo -n ']],'
{
    "Name": "instance-type",
    "Values": [' >>"$tmp_json_file"
IFS=', ' read -ra items <<<"$instance_types"
local array_size
array_size=${#items[@]}
for i in $(seq 0 $((array_size - 1))); do
    echo -n '"'"${items[$i]}"'"' >>"$tmp_json_file"
    if [[ $i -lt $((array_size - 1)) ]]; then
        echo -n ', ' >>"$tmp_json_file"
    fi
done

echo -n ']]]' >>"$tmp_json_file"

local response
response=$(aws ec2 describe-instance-types --filters file://"${tmp_json_file}" \
    --query 'InstanceTypes[*].[InstanceType]' --output text)
```

```

local error_code=$?

rm "$tmp_json_file"

if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    echo "ERROR: AWS reports describe-instance-types operation failed."
    return 1
fi

echo "$response"
return 0
}

```

Fungsi utilitas yang digunakan dalam contoh ini.

```

#####
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
#####
function errecho() {
    printf "%s\n" "$*" 1>&2
}

#####
# function aws_cli_error_log()
#
# This function is used to log the error messages from the AWS CLI.
#
# The function expects the following argument:
#     $1 - The error code returned by the AWS CLI.
#
# Returns:
#     0: - Success.
#
#####
function aws_cli_error_log() {
    local err_code=$1
    errecho "Error code : $err_code"
    if [ "$err_code" == 1 ]; then

```



```
    errecho " One or more S3 transfers failed."
elif [ "$err_code" == 2 ]; then
    errecho " Command line failed to parse."
elif [ "$err_code" == 130 ]; then
    errecho " Process received SIGINT."
elif [ "$err_code" == 252 ]; then
    errecho " Command syntax invalid."
elif [ "$err_code" == 253 ]; then
    errecho " The system environment or configuration was invalid."
elif [ "$err_code" == 254 ]; then
    errecho " The service returned an error."
elif [ "$err_code" == 255 ]; then
    errecho " 255 is a catch-all error."
fi

return 0
}
```

- Untuk detail API, lihat [DescribeInstanceTypes](#) di Referensi AWS CLI Perintah.

CLI

AWS CLI

Contoh 1: Untuk menjelaskan tipe instans

Contoh `describe-instance-types` berikut menampilkan detail untuk tipe instans tertentu.

```
aws ec2 describe-instance-types \
  --instance-types t2.micro
```

Output:

```
{
  "InstanceTypes": [
    {
      "InstanceType": "t2.micro",
      "CurrentGeneration": true,
      "FreeTierEligible": true,
      "SupportedUsageClasses": [
        "on-demand",
        "spot"
      ]
    }
  ]
}
```

```
    ],
    "SupportedRootDeviceTypes": [
      "ebs"
    ],
    "BareMetal": false,
    "Hypervisor": "xen",
    "ProcessorInfo": {
      "SupportedArchitectures": [
        "i386",
        "x86_64"
      ],
      "SustainedClockSpeedInGhz": 2.5
    },
    "VCpuInfo": {
      "DefaultVCpus": 1,
      "DefaultCores": 1,
      "DefaultThreadsPerCore": 1,
      "ValidCores": [
        1
      ],
      "ValidThreadsPerCore": [
        1
      ]
    },
    "MemoryInfo": {
      "SizeInMiB": 1024
    },
    "InstanceStorageSupported": false,
    "EbsInfo": {
      "EbsOptimizedSupport": "unsupported",
      "EncryptionSupport": "supported"
    },
    "NetworkInfo": {
      "NetworkPerformance": "Low to Moderate",
      "MaximumNetworkInterfaces": 2,
      "Ipv4AddressesPerInterface": 2,
      "Ipv6AddressesPerInterface": 2,
      "Ipv6Supported": true,
      "EnaSupport": "unsupported"
    },
    "PlacementGroupInfo": {
      "SupportedStrategies": [
        "partition",
        "spread"
      ]
    }
  }
}
```

```
    ],
    },
    "HibernationSupported": false,
    "BurstablePerformanceSupported": true,
    "DedicatedHostsSupported": false,
    "AutoRecoverySupported": true
  }
]
}
```

Untuk informasi selengkapnya, lihat [Jenis Instans](#) di Panduan Pengguna Amazon Elastic Compute Cloud untuk Instans Linux.

Contoh 2: Untuk memfilter tipe instans yang tersedia

Anda dapat menentukan filter guna mencakup hasil untuk tipe instans yang memiliki karakteristik khusus. Contoh `describe-instance-types` berikut mencantumkan tipe instans yang mendukung hibernasi.

```
aws ec2 describe-instance-types \
  --filters Name=hibernation-supported,Values=true --query
  'InstanceTypes[*].InstanceType'
```

Output:

```
[
  "m5.8xlarge",
  "r3.large",
  "c3.8xlarge",
  "r5.large",
  "m4.4xlarge",
  "c4.large",
  "m5.xlarge",
  "m4.xlarge",
  "c3.large",
  "c4.8xlarge",
  "c4.4xlarge",
  "c5.xlarge",
  "c5.12xlarge",
  "r5.4xlarge",
  "c5.4xlarge"
]
```

Untuk informasi selengkapnya, lihat [Jenis Instans](#) di Panduan Pengguna Amazon Elastic Compute Cloud untuk Instans Linux.

- Untuk detail API, lihat [DescribeInstanceTypes](#) di Referensi AWS CLI Perintah.

Java

SDK untuk Java 2.x

Note

Ada lebih banyak tentang GitHub. Temukan contoh selengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
// Get a list of instance types.
public static String getInstanceTypes(Ec2Client ec2) {
    String instanceType;
    try {
        DescribeInstanceTypesRequest typesRequest =
DescribeInstanceTypesRequest.builder()
            .maxResults(10)
            .build();

        DescribeInstanceTypesResponse response =
ec2.describeInstanceTypes(typesRequest);
        List<InstanceTypeInfo> instanceTypes = response.instanceTypes();
        for (InstanceTypeInfo type : instanceTypes) {
            System.out.println("The memory information of this type is " +
type.memoryInfo().sizeInMiB());
            System.out.println("Network information is " +
type.networkInfo().toString());
            System.out.println("Instance type is " +
type.instanceType().toString());
            instanceType = type.instanceType().toString();
            if (instanceType.compareTo("t2.2xlarge") == 0){
                return instanceType;
            }
        }
    } catch (SsmException e) {
        System.err.println(e.getMessage());
    }
}
```

```
        System.exit(1);
    }
    return "";
}
```

- Untuk detail API, lihat [DescribeInstanceTypes](#) di Referensi AWS SDK for Java 2.x API.

JavaScript

SDK untuk JavaScript (v3)

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
import {
  paginateDescribeInstanceTypes,
  DescribeInstanceTypesCommand,
} from "@aws-sdk/client-ec2";

import { client } from "../libs/client.js";

// List at least the first arm64 EC2 instance type available.
export const main = async () => {
  // The paginate function is a wrapper around the underlying command.
  const paginator = paginateDescribeInstanceTypes(
    // Without limiting the page size, this call can take a long time. pageSize
    // is just sugar for
    // the MaxResults property in the underlying command.
    { client, pageSize: 25 },
    {
      Filters: [
        { Name: "processor-info.supported-architecture", Values: ["x86_64"] },
        { Name: "free-tier-eligible", Values: ["true"] },
      ],
    },
  );

  try {
```

```
const instanceTypes = [];  
  
for await (const page of paginator) {  
  if (page.InstanceTypes.length) {  
    instanceTypes.push(...page.InstanceTypes);  
  
    // When we have at least 1 result, we can stop.  
    if (instanceTypes.length >= 1) {  
      break;  
    }  
  }  
}  
console.log(instanceTypes);  
} catch (err) {  
  console.error(err);  
}  
};
```

- Untuk detail API, lihat [DescribeInstanceTypes](#) di Referensi AWS SDK for JavaScript API.

Kotlin

SDK untuk Kotlin

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
// Get a list of instance types.  
suspend fun getInstanceTypesSc(): String {  
  var instanceType = ""  
  val filterObs = ArrayList<Filter>()  
  val filter =  
    Filter {  
      name = "processor-info.supported-architecture"  
      values = listOf("arm64")  
    }  
  
  filterObs.add(filter)
```

```

val typesRequest =
    DescribeInstanceTypesRequest {
        filters = filterObs
        maxResults = 10
    }
Ec2Client { region = "us-west-2" }.use { ec2 ->
    val response = ec2.describeInstanceTypes(typesRequest)
    response.instanceTypes?.forEach { type ->
        println("The memory information of this type is
${type.memoryInfo?.sizeInMib}")
        println("Maximum number of network cards is
${type.networkInfo?.maximumNetworkCards}")
        instanceType = type.instanceType.toString()
    }
    return instanceType
}
}

```

- Untuk detail API, lihat [DescribeInstanceTypes](#) di AWS SDK untuk referensi API Kotlin.

Python

SDK untuk Python (Boto3)

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```

class InstanceWrapper:
    """Encapsulates Amazon Elastic Compute Cloud (Amazon EC2) instance
actions."""

    def __init__(self, ec2_resource, instance=None):
        """
        :param ec2_resource: A Boto3 Amazon EC2 resource. This high-level
resource
                                is used to create additional high-level objects
                                that wrap low-level Amazon EC2 service actions.

```

```
        :param instance: A Boto3 Instance object. This is a high-level object
that
        wraps instance actions.
        """
        self.ec2_resource = ec2_resource
        self.instance = instance

    @classmethod
    def from_resource(cls):
        ec2_resource = boto3.resource("ec2")
        return cls(ec2_resource)

    def get_instance_types(self, architecture):
        """
        Gets instance types that support the specified architecture and are
designated
        as either 'micro' or 'small'. When an instance is created, the instance
type
        you specify must support the architecture of the AMI you use.

        :param architecture: The kind of architecture the instance types must
support,
                           such as 'x86_64'.
        :return: A list of instance types that support the specified architecture
and are either 'micro' or 'small'.
        """
        try:
            inst_types = []
            it_paginator = self.ec2_resource.meta.client.get_paginator(
                "describe_instance_types"
            )
            for page in it_paginator.paginate(
                Filters=[
                    {
                        "Name": "processor-info.supported-architecture",
                        "Values": [architecture],
                    },
                    {"Name": "instance-type", "Values": ["*.micro", "*.small"]},
                ]
            ):
                inst_types += page["InstanceTypes"]
        except ClientError as err:
            logger.error(
```



```
        "Couldn't get instance types. Here's why: %s: %s",
        err.response["Error"]["Code"],
        err.response["Error"]["Message"],
    )
    raise
else:
    return inst_types
```

- Untuk detail API, lihat [DescribeInstanceTypes](#) di AWS SDK for Python (Boto3) Referensi API.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **DescribeInstances** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `DescribeInstances`.

Contoh tindakan adalah kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Anda dapat melihat tindakan ini dalam konteks dalam contoh kode berikut:

- [Membangun dan mengelola layanan yang tangguh](#)
- [Memulai instans](#)

.NET

AWS SDK for .NET

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
/// <summary>
/// Get information about existing EC2 images.
/// </summary>
```

```
/// <returns>Async task.</returns>
public async Task DescribeInstances()
{
    // List all EC2 instances.
    await GetInstanceDescriptions();

    string tagName = "IncludeInList";
    string tagValue = "Yes";
    await GetInstanceDescriptionsFiltered(tagName, tagValue);
}

/// <summary>
/// Get information for all existing Amazon EC2 instances.
/// </summary>
/// <returns>Async task.</returns>
public async Task GetInstanceDescriptions()
{
    Console.WriteLine("Showing all instances:");
    var paginator = _amazonEC2.Paginators.DescribeInstances(new
DescribeInstancesRequest());

    await foreach (var response in paginator.Responses)
    {
        foreach (var reservation in response.Reservations)
        {
            foreach (var instance in reservation.Instances)
            {
                Console.Write($"Instance ID: {instance.InstanceId}");
                Console.WriteLine($"\\tCurrent State: {instance.State.Name}");
            }
        }
    }
}

/// <summary>
/// Get information about EC2 instances filtered by a tag name and value.
/// </summary>
/// <param name="tagName">The name of the tag to filter on.</param>
/// <param name="tagValue">The value of the tag to look for.</param>
/// <returns>Async task.</returns>
public async Task GetInstanceDescriptionsFiltered(string tagName, string
tagValue)
{
    // This tag filters the results of the instance list.
```

```
var filters = new List<Filter>
{
    new Filter
    {
        Name = $"tag:{tagName}",
        Values = new List<string>
        {
            tagValue,
        },
    },
};
var request = new DescribeInstancesRequest
{
    Filters = filters,
};

Console.WriteLine("\nShowing instances with tag: \"IncludeInList\" set to
\"Yes\".");
var paginator = _amazonEC2.Paginators.DescribeInstances(request);

await foreach (var response in paginator.Responses)
{
    foreach (var reservation in response.Reservations)
    {
        foreach (var instance in reservation.Instances)
        {
            Console.Write($"Instance ID: {instance.InstanceId} ");
            Console.WriteLine($"\\tCurrent State: {instance.State.Name}");
        }
    }
}
```

- Untuk detail API, lihat [DescribeInstances](#) di Referensi AWS SDK for .NET API.

Bash

AWS CLI dengan skrip Bash

 Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
#####
# function ec2_describe_instances
#
# This function describes one or more Amazon Elastic Compute Cloud (Amazon EC2)
# instances.
#
# Parameters:
#     -i instance_id - The ID of the instance to describe (optional).
#     -q query - The query to filter the response (optional).
#     -h - Display help.
#
# Returns:
#     0 - If successful.
#     1 - If it fails.
#####
function ec2_describe_instances() {
    local instance_id query response
    local option OPTARG # Required to use getopt command in a function.

    # bashsupport disable=BP5008
    function usage() {
        echo "function ec2_describe_instances"
        echo "Describes one or more Amazon Elastic Compute Cloud (Amazon EC2)
instances."
        echo "  -i instance_id - The ID of the instance to describe (optional)."
        echo "  -q query - The query to filter the response (optional)."
        echo "  -h - Display help."
        echo ""
    }

    # Retrieve the calling parameters.
    while getopt "i:q:h" option; do
```

```

    case "${option}" in
        i) instance_id="${OPTARG}" ;;
        q) query="${OPTARG}" ;;
        h)
            usage
            return 0
            ;;
        \?)
            echo "Invalid parameter"
            usage
            return 1
            ;;
    esac
done
export OPTIND=1

local aws_cli_args=()

if [[ -n "$instance_id" ]]; then
    # shellcheck disable=SC2206
    aws_cli_args+=("--instance-ids" $instance_id)
fi

local query_arg=""
if [[ -n "$query" ]]; then
    query_arg="--query '$query'"
else
    query_arg="--query Reservations[*].Instances[*].
[InstanceId,ImageId,InstanceType,KeyName,VpcId,PublicIpAddress,State.Name]"
fi

# shellcheck disable=SC2086
response=$(aws ec2 describe-instances \
    "${aws_cli_args[@]}" \
    $query_arg \
    --output text) || {
    aws_cli_error_log ${?}
    errecho "ERROR: AWS reports describe-instances operation failed.$response"
    return 1
}

echo "$response"

return 0

```

```
}

```

Fungsi utilitas yang digunakan dalam contoh ini.

```
#####
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
#####
function errecho() {
    printf "%s\n" "$*" 1>&2
}

#####
# function aws_cli_error_log()
#
# This function is used to log the error messages from the AWS CLI.
#
# The function expects the following argument:
#     $1 - The error code returned by the AWS CLI.
#
# Returns:
#     0: - Success.
#
#####
function aws_cli_error_log() {
    local err_code=$1
    errecho "Error code : $err_code"
    if [ "$err_code" == 1 ]; then
        errecho " One or more S3 transfers failed."
    elif [ "$err_code" == 2 ]; then
        errecho " Command line failed to parse."
    elif [ "$err_code" == 130 ]; then
        errecho " Process received SIGINT."
    elif [ "$err_code" == 252 ]; then
        errecho " Command syntax invalid."
    elif [ "$err_code" == 253 ]; then
        errecho " The system environment or configuration was invalid."
    elif [ "$err_code" == 254 ]; then
        errecho " The service returned an error."
    elif [ "$err_code" == 255 ]; then
        errecho " 255 is a catch-all error."
    }
}

```

```
fi

return 0
}
```

- Untuk detail API, lihat [DescribeInstances](#) di Referensi AWS CLI Perintah.

C++

SDK untuk C++

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
Aws::EC2::EC2Client ec2Client(clientConfiguration);
Aws::EC2::Model::DescribeInstancesRequest request;
bool header = false;
bool done = false;
while (!done) {
    auto outcome = ec2Client.DescribeInstances(request);
    if (outcome.IsSuccess()) {
        if (!header) {
            std::cout << std::left <<
                std::setw(48) << "Name" <<
                std::setw(20) << "ID" <<
                std::setw(25) << "Ami" <<
                std::setw(15) << "Type" <<
                std::setw(15) << "State" <<
                std::setw(15) << "Monitoring" << std::endl;
            header = true;
        }

        const std::vector<Aws::EC2::Model::Reservation> &reservations =
            outcome.GetResult().GetReservations();

        for (const auto &reservation: reservations) {
            const std::vector<Aws::EC2::Model::Instance> &instances =
                reservation.GetInstances();
```

```

        for (const auto &instance: instances) {
            Aws::String instanceStateString =

Aws::EC2::Model::InstanceStateNameMapper::GetNameForInstanceStateName(
                instance.GetState().GetName());

            Aws::String typeString =

Aws::EC2::Model::InstanceTypeMapper::GetNameForInstanceType(
                instance.GetInstanceType());

            Aws::String monitorString =

Aws::EC2::Model::MonitoringStateMapper::GetNameForMonitoringState(
                instance.GetMonitoring().GetState());
            Aws::String name = "Unknown";

            const std::vector<Aws::EC2::Model::Tag> &tags =
instance.GetTags();
            auto nameIter = std::find_if(tags.cbegin(), tags.cend(),
                [](const Aws::EC2::Model::Tag
&tag) {
                    return tag.GetKey() ==
"Name";
                });
            if (nameIter != tags.cend()) {
                name = nameIter->GetValue();
            }
            std::cout <<
                std::setw(48) << name <<
                std::setw(20) << instance.GetInstanceId() <<
                std::setw(25) << instance.GetImageId() <<
                std::setw(15) << typeString <<
                std::setw(15) << instanceStateString <<
                std::setw(15) << monitorString << std::endl;
        }
    }

    if (!outcome.GetResult().GetNextToken().empty()) {
        request.SetNextToken(outcome.GetResult().GetNextToken());
    }
    else {
        done = true;
    }
}

```



```
    }
    else {
        std::cerr << "Failed to describe EC2 instances:" <<
            outcome.GetError().GetMessage() << std::endl;
        return false;
    }
}
```

- Untuk detail API, lihat [DescribeInstances](#) di Referensi AWS SDK for C++ API.

CLI

AWS CLI

Contoh 1: Untuk menjelaskan instans

Contoh `describe-instances` berikut menjelaskan instans tertentu.

```
aws ec2 describe-instances \
  --instance-ids i-1234567890abcdef0
```

Output:

```
{
  "Reservations": [
    {
      "Groups": [],
      "Instances": [
        {
          "AmiLaunchIndex": 0,
          "ImageId": "ami-0abcdef1234567890",
          "InstanceId": "i-1234567890abcdef0",
          "InstanceType": "t3.nano",
          "KeyName": "my-key-pair",
          "LaunchTime": "2022-11-15T10:48:59+00:00",
          "Monitoring": {
            "State": "disabled"
          },
          "Placement": {
            "AvailabilityZone": "us-east-2a",
            "GroupName": "",
            "Tenancy": "default"
          }
        }
      ]
    }
  ]
}
```

```
    },
    "PrivateDnsName": "ip-10-0-0-157.us-east-2.compute.internal",
    "PrivateIpAddress": "10-0-0-157",
    "ProductCodes": [],
    "PublicDnsName": "ec2-34-253-223-13.us-
east-2.compute.amazonaws.com",
    "PublicIpAddress": "34.253.223.13",
    "State": {
      "Code": 16,
      "Name": "running"
    },
    "StateTransitionReason": "",
    "SubnetId": "subnet-04a636d18e83cfac",
    "VpcId": "vpc-1234567890abcdef0",
    "Architecture": "x86_64",
    "BlockDeviceMappings": [
      {
        "DeviceName": "/dev/xvda",
        "Ebs": {
          "AttachTime": "2022-11-15T10:49:00+00:00",
          "DeleteOnTermination": true,
          "Status": "attached",
          "VolumeId": "vol-02e6ccdca7de29cf2"
        }
      }
    ],
    "ClientToken": "1234abcd-1234-abcd-1234-d46a8903e9bc",
    "EbsOptimized": true,
    "EnaSupport": true,
    "Hypervisor": "xen",
    "IamInstanceProfile": {
      "Arn": "arn:aws:iam::111111111111:instance-profile/
AmazonSSMRoleForInstancesQuickSetup",
      "Id": "11111111111111111111"
    },
    "NetworkInterfaces": [
      {
        "Association": {
          "IpOwnerId": "amazon",
          "PublicDnsName": "ec2-34-253-223-13.us-
east-2.compute.amazonaws.com",
          "PublicIp": "34.253.223.13"
        },
        "Attachment": {
```

```

        "AttachTime": "2022-11-15T10:48:59+00:00",
        "AttachmentId": "eni-attach-1234567890abcdefg",
        "DeleteOnTermination": true,
        "DeviceIndex": 0,
        "Status": "attached",
        "NetworkCardIndex": 0
    },
    "Description": "",
    "Groups": [
        {
            "GroupName": "launch-wizard-146",
            "GroupId": "sg-1234567890abcdefg"
        }
    ],
    "Ipv6Addresses": [],
    "MacAddress": "00:11:22:33:44:55",
    "NetworkInterfaceId": "eni-1234567890abcdefg",
    "OwnerId": "104024344472",
    "PrivateDnsName": "ip-10-0-0-157.us-
east-2.compute.internal",
    "PrivateIpAddress": "10-0-0-157",
    "PrivateIpAddresses": [
        {
            "Association": {
                "IpOwnerId": "amazon",
                "PublicDnsName": "ec2-34-253-223-13.us-
east-2.compute.amazonaws.com",
                "PublicIp": "34.253.223.13"
            },
            "Primary": true,
            "PrivateDnsName": "ip-10-0-0-157.us-
east-2.compute.internal",
            "PrivateIpAddress": "10-0-0-157"
        }
    ],
    "SourceDestCheck": true,
    "Status": "in-use",
    "SubnetId": "subnet-1234567890abcdefg",
    "VpcId": "vpc-1234567890abcdefg",
    "InterfaceType": "interface"
    }
],
"RootDeviceName": "/dev/xvda",
"RootDeviceType": "ebs",

```

```
"SecurityGroups": [
  {
    "GroupName": "launch-wizard-146",
    "GroupId": "sg-1234567890abcdefg"
  }
],
"SourceDestCheck": true,
"Tags": [
  {
    "Key": "Name",
    "Value": "my-instance"
  }
],
"VirtualizationType": "hvm",
"CpuOptions": {
  "CoreCount": 1,
  "ThreadsPerCore": 2
},
"CapacityReservationSpecification": {
  "CapacityReservationPreference": "open"
},
"HibernationOptions": {
  "Configured": false
},
"MetadataOptions": {
  "State": "applied",
  "HttpTokens": "optional",
  "HttpPutResponseHopLimit": 1,
  "HttpEndpoint": "enabled",
  "HttpProtocolIpv6": "disabled",
  "InstanceMetadataTags": "enabled"
},
"EnclaveOptions": {
  "Enabled": false
},
"PlatformDetails": "Linux/UNIX",
"UsageOperation": "RunInstances",
"UsageOperationUpdateTime": "2022-11-15T10:48:59+00:00",
"PrivateDnsNameOptions": {
  "HostnameType": "ip-name",
  "EnableResourceNameDnsARecord": true,
  "EnableResourceNameDnsAAAARecord": false
},
"MaintenanceOptions": {
```

```
        "AutoRecovery": "default"
      }
    }
  ],
  "OwnerId": "111111111111",
  "ReservationId": "r-1234567890abcdefg"
}
]
```

Contoh 2: Untuk memfilter instans dengan tipe tertentu

Contoh `describe-instances` berikut menggunakan filter guna mencakup hasil untuk instans dari tipe tertentu.

```
aws ec2 describe-instances \
  --filters Name=instance-type,Values=m5.large
```

Untuk contoh output, lihat Contoh 1.

Untuk informasi selengkapnya, lihat [Cantumkan dan filter menggunakan CLI](#) di Panduan Pengguna Amazon EC2.

Contoh 3: Untuk memfilter instans dengan tipe dan Zona Ketersediaan tertentu

Contoh `describe-instances` berikut menggunakan banyak filter guna mencakup hasil untuk instans dengan tipe tertentu yang juga ada di Zona Ketersediaan tertentu.

```
aws ec2 describe-instances \
  --filters Name=instance-type,Values=t2.micro,t3.micro Name=availability-
  zone,Values=us-east-2c
```

Untuk contoh output, lihat Contoh 1.

Contoh 4: Untuk memfilter instans dengan tipe dan Zona Ketersediaan tertentu menggunakan file JSON

Contoh `describe-instances` berikut menggunakan file input JSON untuk melakukan pemfilteran yang sama seperti contoh sebelumnya. Ketika filter menjadi lebih rumit, filter tersebut dapat lebih mudah ditentukan dalam file JSON.

```
aws ec2 describe-instances \
```

```
--filters file://filters.json
```

Isi dari `filters.json`:

```
[
  {
    "Name": "instance-type",
    "Values": ["t2.micro", "t3.micro"]
  },
  {
    "Name": "availability-zone",
    "Values": ["us-east-2c"]
  }
]
```

Untuk contoh output, lihat Contoh 1.

Contoh 5: Untuk memfilter instans dengan tanda Pemilik tertentu

Contoh `describe-instances` berikut menggunakan filter tanda guna mencakup hasil untuk instans yang memiliki tanda dengan kunci tanda tertentu (Pemilik), terlepas dari nilai tandanya.

```
aws ec2 describe-instances \
  --filters "Name=tag-key,Values=Owner"
```

Untuk contoh output, lihat Contoh 1.

Contoh 6: Untuk memfilter instans dengan nilai tanda `my-team` tertentu

Contoh `describe-instances` berikut menggunakan filter tanda guna mencakup hasil untuk instans yang memiliki tanda dengan kunci tanda tertentu (`my-team`), terlepas dari nilai tandanya.

```
aws ec2 describe-instances \
  --filters "Name=tag-value,Values=my-team"
```

Untuk contoh output, lihat Contoh 1.

Contoh 7: Untuk memfilter instans dengan tanda Pemilik dan nilai `my-team` tertentu

Contoh `describe-instances` berikut menggunakan filter tanda guna mencakup hasil untuk instans yang memiliki tanda tertentu (Pemilik=`my-team`).

```
aws ec2 describe-instances \  
  --filters "Name=tag:Owner,Values=my-team"
```

Untuk contoh output, lihat Contoh 1.

Contoh 8: Guna hanya menampilkan ID instans dan subnet untuk semua instans

Contoh `describe-instances` berikut menggunakan parameter `--query` untuk hanya menampilkan ID instans dan subnet untuk semua instans, dalam format JSON.

Linux dan macOS:

```
aws ec2 describe-instances \  
  --query 'Reservations[*].Instances[*].{Instance:InstanceId,Subnet:SubnetId}' \  
 \  
  --output json
```

Windows:

```
aws ec2 describe-instances ^ \  
  --query "Reservations[*].Instances[*].{Instance:InstanceId,Subnet:SubnetId}" \  
 ^ \  
  --output json
```

Output:

```
[  
  {  
    "Instance": "i-057750d42936e468a",  
    "Subnet": "subnet-069beee9b12030077"  
  },  
  {  
    "Instance": "i-001efd250faaa6ffa",  
    "Subnet": "subnet-0b715c6b7db68927a"  
  },  
  {  
    "Instance": "i-027552a73f021f3bd",  
    "Subnet": "subnet-0250c25a1f4e15235"  
  }  
  ...  
]
```

Contoh 9: Untuk memfilter instans dari tipe tertentu dan hanya menampilkan ID instansnya

Contoh `describe-instances` berikut menggunakan filter guna mencakup hasil untuk instans dari tipe tertentu dan parameter `--query` agar hanya menampilkan ID instans.

```
aws ec2 describe-instances \
  --filters "Name=instance-type,Values=t2.micro" \
  --query "Reservations[*].Instances[*].[InstanceId]" \
  --output text
```

Output:

```
i-031c0dc19de2fb70c
i-00d8bfff789a736b75
i-0b715c6b7db68927a
i-0626d4edd54f1286d
i-00b8ae04f9f99908e
i-0fc71c25d2374130c
```

Contoh 10: Untuk memfilter instans dari tipe tertentu dan hanya menampilkan ID instans, Zona Ketersediaan, dan nilai tanda tertentu

Contoh `describe-instances` berikut menampilkan ID instans, Zona Ketersediaan, dan nilai dari tanda Name untuk instans yang memiliki tanda dengan nama `tag-key`, dalam format tabel.

Linux dan macOS:

```
aws ec2 describe-instances \
  --filters Name=tag-key,Values=Name \
  --query 'Reservations[*].Instances[*].
{Instance:InstanceId,AZ:Placement.AvailabilityZone,Name:Tags[?Key==`Name`]|
[0].Value}' \
  --output table
```

Windows:

```
aws ec2 describe-instances ^
  --filters Name=tag-key,Values=Name ^
  --query "Reservations[*].Instances[*].
{Instance:InstanceId,AZ:Placement.AvailabilityZone,Name:Tags[?Key=='Name']|
[0].Value}" ^
```



```
--output table
```

Output:

```
-----
|                               DescribeInstances                               |
+-----+-----+-----+-----+
|      AZ      | Instance |      Name      |
+-----+-----+-----+-----+
| us-east-2b  | i-057750d42936e468a | my-prod-server |
| us-east-2a  | i-001efd250faaa6ffa | test-server-1  |
| us-east-2a  | i-027552a73f021f3bd | test-server-2  |
+-----+-----+-----+-----+
```

Contoh 11: Untuk menjelaskan instans dalam grup penempatan partisi

Contoh `describe-instances` berikut menjelaskan instans tertentu. Output-nya mencakup informasi penempatan untuk instans, yang berisi nama grup penempatan dan nomor partisi untuk instans tersebut.

```
aws ec2 describe-instances \
  --instance-ids i-0123a456700123456 \
  --query "Reservations[*].Instances[*].Placement"
```

Output:

```
[
  [
    {
      "AvailabilityZone": "us-east-1c",
      "GroupName": "HDFS-Group-A",
      "PartitionNumber": 3,
      "Tenancy": "default"
    }
  ]
]
```

Untuk informasi selengkapnya, lihat [Menjelaskan instans dalam grup penempatan](#) di Panduan Pengguna Amazon EC2.

Contoh 12: Untuk memfilter ke instans dengan grup penempatan dan nomor partisi tertentu

Contoh `describe-instances` berikut memfilter hasil menjadi hanya instans dengan grup penempatan dan nomor partisi tertentu.

```
aws ec2 describe-instances \  
  --filters "Name=placement-group-name,Values=HDFS-Group-A" "Name=placement-  
partition-number,Values=7"
```

Berikut ini hanya menunjukkan informasi yang relevan dari output.

```
"Instances": [  
  {  
    "InstanceId": "i-0123a456700123456",  
    "InstanceType": "r4.large",  
    "Placement": {  
      "AvailabilityZone": "us-east-1c",  
      "GroupName": "HDFS-Group-A",  
      "PartitionNumber": 7,  
      "Tenancy": "default"  
    }  
  },  
  {  
    "InstanceId": "i-9876a543210987654",  
    "InstanceType": "r4.large",  
    "Placement": {  
      "AvailabilityZone": "us-east-1c",  
      "GroupName": "HDFS-Group-A",  
      "PartitionNumber": 7,  
      "Tenancy": "default"  
    }  
  },  
]
```

Untuk informasi selengkapnya, lihat [Menjelaskan instans dalam grup penempatan](#) di Panduan Pengguna Amazon EC2.

Contoh 13: Untuk memfilter ke instans yang dikonfigurasi guna memungkinkan akses ke tanda dari metadata instans

Contoh `describe-instances` berikut memfilter hasil menjadi hanya instans yang dikonfigurasi guna memungkinkan akses ke tanda instans dari metadata instans.

```
aws ec2 describe-instances \
  --filters "Name=metadata-options.instance-metadata-tags,Values=enabled" \
  --query "Reservations[*].Instances[*].InstanceId" \
  --output text
```

Berikut menunjukkan output yang diharapkan.

```
i-1234567890abcdefg
i-abcdefg1234567890
i-111111111aaaaaaaa
i-aaaaaaaa111111111
```

Untuk informasi selengkapnya, lihat [Bekerja dengan tanda instans dalam metadata instans](#) di Panduan Pengguna Amazon EC2.

- Untuk detail API, lihat [DescribeInstances](#) di Referensi AWS CLI Perintah.

Java

SDK untuk Java 2.x

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.ec2.Ec2Client;
import software.amazon.awssdk.services.ec2.model.DescribeInstancesRequest;
import software.amazon.awssdk.services.ec2.model.Ec2Exception;
import software.amazon.awssdk.services.ec2.paginators.DescribeInstancesIterable;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
 * started.html
```

```
*/
public class DescribeInstances {
    public static void main(String[] args) {
        Region region = Region.US_EAST_1;
        Ec2Client ec2 = Ec2Client.builder()
            .region(region)
            .build();

        describeEC2Instances(ec2);
        ec2.close();
    }

    public static void describeEC2Instances(Ec2Client ec2) {
        try {
            DescribeInstancesRequest request = DescribeInstancesRequest.builder()
                .maxResults(10)
                .build();

            DescribeInstancesIterable instancesIterable =
ec2.describeInstancesPaginator(request);
            instancesIterable.stream()
                .flatMap(r -> r.reservations().stream())
                .flatMap(reservation -> reservation.instances().stream())
                .forEach(instance -> {
                    System.out.println("Instance Id is " +
instance.instanceId());
                    System.out.println("Image id is " + instance.imageId());
                    System.out.println("Instance type is " +
instance.instanceType());
                    System.out.println("Instance state name is " +
instance.state().name());
                    System.out.println("Monitoring information is " +
instance.monitoring().state());
                });

        } catch (Ec2Exception e) {
            System.err.println(e.awsErrorDetails().errorCode());
            System.exit(1);
        }
    }
}
```

- Untuk detail API, lihat [DescribeInstances](#) di Referensi AWS SDK for Java 2.x API.

JavaScript

SDK untuk JavaScript (v3)

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
import { DescribeInstancesCommand } from "@aws-sdk/client-ec2";

import { client } from "../libs/client.js";

// List all of your EC2 instances running with x86_64 architecture that were
// launched this month.
export const main = async () => {
  const d = new Date();
  const year = d.getFullYear();
  const month = `${d.getMonth() + 1}`.slice(-2);
  const launchTimePattern = `${year}-${month}-*`;
  const command = new DescribeInstancesCommand({
    Filters: [
      { Name: "architecture", Values: ["x86_64"] },
      { Name: "instance-state-name", Values: ["running"] },
      {
        Name: "launch-time",
        Values: [launchTimePattern],
      },
    ],
  });

  try {
    const { Reservations } = await client.send(command);
    const instanceList = Reservations.reduce((prev, current) => {
      return prev.concat(current.Instances);
    }, []);

    console.log(instanceList);
  } catch (err) {
    console.error(err);
  }
}
```

```
};
```

- Untuk detail API, lihat [DescribeInstances](#) di Referensi AWS SDK for JavaScript API.

Kotlin

SDK untuk Kotlin

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
suspend fun describeEC2Instances() {
    val request =
        DescribeInstancesRequest {
            maxResults = 6
        }

    Ec2Client { region = "us-west-2" }.use { ec2 ->
        val response = ec2.describeInstances(request)
        response.reservations?.forEach { reservation ->
            reservation.instances?.forEach { instance ->
                println("Instance Id is ${instance.instanceId}")
                println("Image id is ${instance.imageId}")
                println("Instance type is ${instance.instanceType}")
                println("Instance state name is ${instance.state?.name}")
                println("monitoring information is
                    ${instance.monitoring?.state}")
            }
        }
    }
}
```

- Untuk detail API, lihat [DescribeInstances](#) di AWS SDK untuk referensi API Kotlin.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini menjelaskan contoh yang ditentukan.

```
(Get-EC2Instance -InstanceId i-12345678).Instances
```

Output:

```
AmiLaunchIndex      : 0
Architecture        : x86_64
BlockDeviceMappings : {/dev/sda1}
ClientToken         : TleEy1448154045270
EbsOptimized        : False
Hypervisor          : xen
IamInstanceProfile  : Amazon.EC2.Model.IamInstanceProfile
ImageId             : ami-12345678
InstanceId          : i-12345678
InstanceLifecycle   :
InstanceType        : t2.micro
KernelId            :
KeyName             : my-key-pair
LaunchTime          : 12/4/2015 4:44:40 PM
Monitoring          : Amazon.EC2.Model.Monitoring
NetworkInterfaces   : {ip-10-0-2-172.us-west-2.compute.internal}
Placement           : Amazon.EC2.Model.Placement
Platform            : Windows
PrivateDnsName      : ip-10-0-2-172.us-west-2.compute.internal
PrivateIpAddress    : 10.0.2.172
ProductCodes        : {}
PublicDnsName       :
PublicIpAddress     :
RamdiskId           :
RootDeviceName      : /dev/sda1
RootDeviceType      : ebs
SecurityGroups      : {default}
SourceDestCheck     : True
SpotInstanceRequestId :
SriovNetSupport     :
State                : Amazon.EC2.Model.InstanceState
StateReason         :
StateTransitionReason :
```

```
SubnetId      : subnet-12345678
Tags         : {Name}
VirtualizationType : hvm
VpcId       : vpc-12345678
```

Contoh 2: Contoh ini menjelaskan semua instans Anda di wilayah saat ini, dikelompokkan berdasarkan reservasi. Untuk melihat detail instance, perluas koleksi Instances dalam setiap objek reservasi.

```
Get-EC2Instance
```

Output:

```
GroupNames    : {}
Groups        : {}
Instances     : {}
OwnerId       : 123456789012
RequesterId   : 226008221399
ReservationId : r-c5df370c

GroupNames    : {}
Groups        : {}
Instances     : {}
OwnerId       : 123456789012
RequesterId   : 854251627541
ReservationId : r-63e65bab
...
```

Contoh 3: Contoh ini menggambarkan penggunaan filter untuk kueri instans EC2 di subnet tertentu dari VPC.

```
(Get-EC2Instance -Filter @{Name="vpc-id";Values="vpc-1a2bc34d"},@{Name="subnet-id";Values="subnet-1a2b3c4d"}).Instances
```

Output:

```
InstanceId      InstanceType Platform PrivateIpAddress PublicIpAddress
SecurityGroups SubnetId      VpcId
-----
-----
```



```
i-01af...82cf180e19 t2.medium    Windows  10.0.0.98    ...
      subnet-1a2b3c4d vpc-1a2b3c4d
i-0374...7e9d5b0c45 t2.xlarge  Windows  10.0.0.53    ...
      subnet-1a2b3c4d vpc-1a2b3c4d
```

- Untuk detail API, lihat [DescribeInstances](#) di Referensi AWS Tools for PowerShell Cmdlet.

Python

SDK untuk Python (Boto3)

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
class InstanceWrapper:
    """Encapsulates Amazon Elastic Compute Cloud (Amazon EC2) instance
    actions."""

    def __init__(self, ec2_resource, instance=None):
        """
        :param ec2_resource: A Boto3 Amazon EC2 resource. This high-level
        resource
                                is used to create additional high-level objects
                                that wrap low-level Amazon EC2 service actions.
        :param instance: A Boto3 Instance object. This is a high-level object
        that
                                wraps instance actions.
        """
        self.ec2_resource = ec2_resource
        self.instance = instance

    @classmethod
    def from_resource(cls):
        ec2_resource = boto3.resource("ec2")
        return cls(ec2_resource)

    def display(self, indent=1):
        """
```

```

Displays information about an instance.

:param indent: The visual indent to apply to the output.
"""
if self.instance is None:
    logger.info("No instance to display.")
    return

try:
    self.instance.load()
    ind = "\t" * indent
    print(f"{ind}ID: {self.instance.id}")
    print(f"{ind}Image ID: {self.instance.image_id}")
    print(f"{ind}Instance type: {self.instance.instance_type}")
    print(f"{ind}Key name: {self.instance.key_name}")
    print(f"{ind}VPC ID: {self.instance.vpc_id}")
    print(f"{ind}Public IP: {self.instance.public_ip_address}")
    print(f"{ind}State: {self.instance.state['Name']}")
except ClientError as err:
    logger.error(
        "Couldn't display your instance. Here's why: %s: %s",
        err.response["Error"]["Code"],
        err.response["Error"]["Message"],
    )
    raise

```

- Untuk detail API, lihat [DescribeInstances](#) di AWS SDK for Python (Boto3) Referensi API.

Ruby

SDK untuk Ruby

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
require "aws-sdk-ec2"
```

```
# @param ec2_resource [Aws::EC2::Resource] An initialized EC2 resource object.
# @example
# list_instance_ids_states(Aws::EC2::Resource.new(region: 'us-west-2'))
def list_instance_ids_states(ec2_resource)
  response = ec2_resource.instances
  if response.count.zero?
    puts "No instances found."
  else
    puts "Instances -- ID, state:"
    response.each do |instance|
      puts "#{instance.id}, #{instance.state.name}"
    end
  end
end
rescue StandardError => e
  puts "Error getting information about instances: #{e.message}"
end

# Example usage:
def run_me
  region = ""
  # Print usage information and then stop.
  if ARGV[0] == "--help" || ARGV[0] == "-h"
    puts "Usage:  ruby ec2-ruby-example-get-all-instance-info.rb REGION"
    # Replace us-west-2 with the AWS Region you're using for Amazon EC2.
    puts "Example: ruby ec2-ruby-example-get-all-instance-info.rb us-west-2"
    exit 1
  # If no values are specified at the command prompt, use these default values.
  # Replace us-west-2 with the AWS Region you're using for Amazon EC2.
  elsif ARGV.count.zero?
    region = "us-west-2"
  # Otherwise, use the values as specified at the command prompt.
  else
    region = ARGV[0]
  end
  ec2_resource = Aws::EC2::Resource.new(region: region)
  list_instance_ids_states(ec2_resource)
end

run_me if $PROGRAM_NAME == __FILE__
```

- Untuk detail API, lihat [DescribeInstances](#) di Referensi AWS SDK for Ruby API.

Rust

SDK untuk Rust

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
async fn show_state(client: &Client, ids: Option<Vec<String>>) -> Result<(),
Error> {
    let resp = client
        .describe_instances()
        .set_instance_ids(ids)
        .send()
        .await?;


    for reservation in resp.reservations() {
        for instance in reservation.instances() {
            println!("Instance ID: {}", instance.instance_id().unwrap());
            println!(
                "State:      {:?}",
                instance.state().unwrap().name().unwrap()
            );
            println!();
        }
    }

    Ok(())
}
```

- Untuk detail API, lihat [DescribeInstances](#) referensi AWS SDK for Rust API.

SAP ABAP

SDK untuk SAP ABAP

 Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```

TRY.
    oo_result = lo_ec2->describeinstances( ) .
oo_result is returned for testing purposes. "

" Retrieving details of EC2 instances. "
DATA: lv_instance_id    TYPE /aws1/ec2string,
      lv_status         TYPE /aws1/ec2instancename,
      lv_instance_type  TYPE /aws1/ec2instancetype,
      lv_image_id       TYPE /aws1/ec2string.
LOOP AT oo_result->get_reservations( ) INTO DATA(lo_reservation).
  LOOP AT lo_reservation->get_instances( ) INTO DATA(lo_instance).
    lv_instance_id = lo_instance->get_instanceid( ).
    lv_status = lo_instance->get_state( )->get_name( ).
    lv_instance_type = lo_instance->get_instancetype( ).
    lv_image_id = lo_instance->get_imageid( ).
  ENDLLOOP.
ENDLOOP.
MESSAGE 'Retrieved information about EC2 instances.' TYPE 'I'.
CATCH /aws1/cx_rt_service_generic INTO DATA(lo_exception).
  DATA(lv_error) = |"{ lo_exception->av_err_code }" - { lo_exception-
>av_err_msg }|.
  MESSAGE lv_error TYPE 'E'.
ENDTRY.

```

- Untuk detail API, lihat [DescribeInstances](#) di AWS SDK untuk referensi SAP ABAP API.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan `DescribeInternetGateways` dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `DescribeInternetGateways`.

CLI

AWS CLI

Untuk menggambarkan gateway internet

`describe-internet-gateways` Contoh berikut menjelaskan gateway internet yang ditentukan.

```
aws ec2 describe-internet-gateways \
  --internet-gateway-ids igw-0d0fb496b3EXAMPLE
```

Output:

```
{
  "InternetGateways": [
    {
      "Attachments": [
        {
          "State": "available",
          "VpcId": "vpc-0a60eb65b4EXAMPLE"
        }
      ],
      "InternetGatewayId": "igw-0d0fb496b3EXAMPLE",
      "OwnerId": "123456789012",
      "Tags": [
        {
          "Key": "Name",
          "Value": "my-igw"
        }
      ]
    }
  ]
}
```

Untuk informasi lebih lanjut, lihat [Gateway internet](#) di Panduan Pengguna Amazon VPC.

- Untuk detail API, lihat [DescribeInternetGateways](#) di Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini menjelaskan gateway Internet yang ditentukan.

```
Get-EC2InternetGateway -InternetGatewayId igw-1a2b3c4d
```

Output:

Attachments	InternetGatewayId	Tags
{vpc-1a2b3c4d}	igw-1a2b3c4d	{}

Contoh 2: Contoh ini menjelaskan semua gateway Internet Anda.

```
Get-EC2InternetGateway
```

Output:

Attachments	InternetGatewayId	Tags
{vpc-1a2b3c4d}	igw-1a2b3c4d	{}
{}	igw-2a3b4c5d	{}

- Untuk detail API, lihat [DescribeInternetGateways](#) di Referensi AWS Tools for PowerShell Cmdlet.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **DescribeKeyPairs** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `DescribeKeyPairs`.

Contoh tindakan adalah kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Anda dapat melihat tindakan ini dalam konteks dalam contoh kode berikut:

- [Memulai instans](#)

.NET

AWS SDK for .NET

Note


Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
/// <summary>
/// Get information about an Amazon EC2 key pair.
/// </summary>
/// <param name="keyPairName">The name of the key pair.</param>
/// <returns>A list of key pair information.</returns>
public async Task<List<KeyPairInfo>> DescribeKeyPairs(string keyPairName)
{
    var request = new DescribeKeyPairsRequest();
    if (!string.IsNullOrEmpty(keyPairName))
    {
        request = new DescribeKeyPairsRequest
        {
            KeyNames = new List<string> { keyPairName }
        };
    }
    var response = await _amazonEC2.DescribeKeyPairsAsync(request);
    return response.KeyPairs.ToList();
}
```

- Untuk detail API, lihat [DescribeKeyPairs](#) di Referensi AWS SDK for .NET API.

Bash

AWS CLI dengan skrip Bash

 Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
#####
# function ec2_describe_key_pairs
#
# This function describes one or more Amazon Elastic Compute Cloud (Amazon EC2)
# key pairs.
#
# Parameters:
#     -h - Display help.
#
# And:
#     0 - If successful.
#     1 - If it fails.
#####
function ec2_describe_key_pairs() {
    local option OPTARG # Required to use getopt command in a function.

    # bashsupport disable=BP5008
    function usage() {
        echo "function ec2_describe_key_pairs"
        echo "Describes one or more Amazon Elastic Compute Cloud (Amazon EC2) key
pairs."
        echo "  -h - Display help."
        echo ""
    }

    # Retrieve the calling parameters.
    while getopt "h" option; do
        case "${option}" in
            h)
                usage
                return 0
            ;;
        esac
    done
}
```

```

    \?)
    echo "Invalid parameter"
    usage
    return 1
    ;;
esac
done
export OPTIND=1

local response

response=$(aws ec2 describe-key-pairs \
  --query 'KeyPairs[*].[KeyName, KeyFingerprint]' \
  --output text) || {
  aws_cli_error_log ${?}
  errecho "ERROR: AWS reports describe-key-pairs operation failed.$response"
  return 1
}

echo "$response"

return 0
}

```

Fungsi utilitas yang digunakan dalam contoh ini.

```

#####
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
#####
function errecho() {
  printf "%s\n" "$*" 1>&2
}

#####
# function aws_cli_error_log()
#
# This function is used to log the error messages from the AWS CLI.
#
# The function expects the following argument:
#     $1 - The error code returned by the AWS CLI.

```

```

#
# Returns:
#     0: - Success.
#
#####
function aws_cli_error_log() {
    local err_code=$1
    errecho "Error code : $err_code"
    if [ "$err_code" == 1 ]; then
        errecho " One or more S3 transfers failed."
    elif [ "$err_code" == 2 ]; then
        errecho " Command line failed to parse."
    elif [ "$err_code" == 130 ]; then
        errecho " Process received SIGINT."
    elif [ "$err_code" == 252 ]; then
        errecho " Command syntax invalid."
    elif [ "$err_code" == 253 ]; then
        errecho " The system environment or configuration was invalid."
    elif [ "$err_code" == 254 ]; then
        errecho " The service returned an error."
    elif [ "$err_code" == 255 ]; then
        errecho " 255 is a catch-all error."
    fi

    return 0
}

```

- Untuk detail API, lihat [DescribeKeyPairs](#) di Referensi AWS CLI Perintah.

C++

SDK untuk C++

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```

Aws::EC2::EC2Client ec2Client(clientConfiguration);
Aws::EC2::Model::DescribeKeyPairsRequest request;

```

```
auto outcome = ec2Client.DescribeKeyPairs(request);
if (outcome.IsSuccess()) {
    std::cout << std::left <<
        std::setw(32) << "Name" <<
        std::setw(64) << "Fingerprint" << std::endl;

    const std::vector<Aws::EC2::Model::KeyPairInfo> &key_pairs =
        outcome.GetResult().GetKeyPairs();
    for (const auto &key_pair: key_pairs) {
        std::cout << std::left <<
            std::setw(32) << key_pair.GetKeyName() <<
            std::setw(64) << key_pair.GetKeyFingerprint() << std::endl;
    }
}
else {
    std::cerr << "Failed to describe key pairs:" <<
        outcome.GetError().GetMessage() << std::endl;
}
```

- Untuk detail API, lihat [DescribeKeyPairs](#) di Referensi AWS SDK for C++ API.

CLI

AWS CLI

Untuk menampilkan pasangan kunci

Contoh `describe-key-pairs` berikut menampilkan informasi tentang pasangan kunci tertentu.

```
aws ec2 describe-key-pairs \
    --key-names my-key-pair
```

Output:

```
{
  "KeyPairs": [
    {
      "KeyId": "key-0b94643da6EXAMPLE",
```

```
        "KeyFingerprint":
          "1f:51:ae:28:bf:89:e9:d8:1f:25:5d:37:2d:7d:b8:ca:9f:f5:f1:6f",
          "KeyName": "my-key-pair",
          "KeyType": "rsa",
          "Tags": [],
          "CreateTime": "2022-05-27T21:51:16.000Z"
        }
      ]
    }
  }
```

Untuk informasi selengkapnya, lihat [Jelaskan kunci publik](#) di Panduan Pengguna Amazon EC2.

- Untuk detail API, lihat [DescribeKeyPairs](#) di Referensi AWS CLI Perintah.

Java

SDK untuk Java 2.x

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
public static void describeKeys(Ec2Client ec2) {
    try {
        DescribeKeyPairsResponse response = ec2.describeKeyPairs();
        response.keyPairs().forEach(keyPair -> System.out.printf(
            "Found key pair with name %s " +
            "and fingerprint %s",
            keyPair.keyName(),
            keyPair.keyFingerprint()));
    } catch (Ec2Exception e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
}
```

- Untuk detail API, lihat [DescribeKeyPairs](#) di Referensi AWS SDK for Java 2.x API.

JavaScript

SDK untuk JavaScript (v3)

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
import { DescribeKeyPairsCommand } from "@aws-sdk/client-ec2";

import { client } from "../libs/client.js";

export const main = async () => {
  const command = new DescribeKeyPairsCommand({});

  try {
    const { KeyPairs } = await client.send(command);
    const keyPairList = KeyPairs.map(
      (kp) => ` • ${kp.KeyPairId}: ${kp.KeyName}`,
    ).join("\n");
    console.log("The following key pairs were found in your account:");
    console.log(keyPairList);
  } catch (err) {
    console.error(err);
  }
};
```

- Untuk detail API, lihat [DescribeKeyPairs](#) di Referensi AWS SDK for JavaScript API.

Kotlin

SDK untuk Kotlin

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
suspend fun describeEC2Keys() {
    Ec2Client { region = "us-west-2" }.use { ec2 ->
        val response = ec2.describeKeyPairs(DescribeKeyPairsRequest {})
        response.keyPairs?.forEach { keyPair ->
            println("Found key pair with name ${keyPair.keyName} and fingerprint
                ${ keyPair.keyFingerprint}")
        }
    }
}
```

- Untuk detail API, lihat [DescribeKeyPairs](#) di AWS SDK untuk referensi API Kotlin.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini menjelaskan key pair yang ditentukan.

```
Get-EC2KeyPair -KeyName my-key-pair
```

Output:

KeyFingerprint	KeyName
-----	-----
1f:51:ae:28:bf:89:e9:d8:1f:25:5d:37:2d:7d:b8:ca:9f:f5:f1:6f	my-key-pair

Contoh 2: Contoh ini menjelaskan semua pasangan kunci Anda.

```
Get-EC2KeyPair
```

- Untuk detail API, lihat [DescribeKeyPairs](#) di Referensi AWS Tools for PowerShell Cmdlet.

Python

SDK untuk Python (Boto3)

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
class KeyPairWrapper:
    """Encapsulates Amazon Elastic Compute Cloud (Amazon EC2) key pair
    actions."""

    def __init__(self, ec2_resource, key_file_dir, key_pair=None):
        """
        :param ec2_resource: A Boto3 Amazon EC2 resource. This high-level
        resource
                                is used to create additional high-level objects
                                that wrap low-level Amazon EC2 service actions.
        :param key_file_dir: The folder where the private key information is
        stored.
                                This should be a secure folder.
        :param key_pair: A Boto3 KeyPair object. This is a high-level object that
        wraps key pair actions.
        """
        self.ec2_resource = ec2_resource
        self.key_pair = key_pair
        self.key_file_path = None
        self.key_file_dir = key_file_dir

    @classmethod
    def from_resource(cls):
        ec2_resource = boto3.resource("ec2")
        return cls(ec2_resource, tempfile.TemporaryDirectory())

    def list(self, limit):
        """
        Displays a list of key pairs for the current account.

        :param limit: The maximum number of key pairs to list.
```



```

"""
try:
    for kp in self.ec2_resource.key_pairs.limit(limit):
        print(f"Found {kp.key_type} key {kp.name} with fingerprint:")
        print(f"\t{kp.key_fingerprint}")
except ClientError as err:
    logger.error(
        "Couldn't list key pairs. Here's why: %s: %s",
        err.response["Error"]["Code"],
        err.response["Error"]["Message"],
    )
    raise

```

- Untuk detail API, lihat [DescribeKeyPairs](#) di AWS SDK for Python (Boto3) Referensi API.

SAP ABAP

SDK untuk SAP ABAP

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```

TRY.
    oo_result = lo_ec2->describekeypairs( ) .
    oo_result is returned for testing purposes. "
    DATA(lt_key_pairs) = oo_result->get_keypairs( ).
    MESSAGE 'Retrieved information about key pairs.' TYPE 'I'.
    CATCH /aws1/cx_rt_service_generic INTO DATA(lo_exception).
    DATA(lv_error) = |"{ lo_exception->av_err_code }" - { lo_exception-
>av_err_msg }|.
    MESSAGE lv_error TYPE 'E'.
ENDTRY.

```

- Untuk detail API, lihat [DescribeKeyPairs](#) di AWS SDK untuk referensi SAP ABAP API.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **DescribeNetworkAcls** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `DescribeNetworkAcls`.

CLI

AWS CLI

Untuk menggambarkan ACL jaringan Anda

`describe-network-acls` Contoh berikut mengambil rincian tentang ACL jaringan Anda.

```
aws ec2 describe-network-acls
```

Output:

```
{
  "NetworkAcls": [
    {
      "Associations": [
        {
          "NetworkAclAssociationId": "aclassoc-0c1679dc41EXAMPLE",
          "NetworkAclId": "acl-0ea1f54ca7EXAMPLE",
          "SubnetId": "subnet-0931fc2fa5EXAMPLE"
        }
      ],
      "Entries": [
        {
          "CidrBlock": "0.0.0.0/0",
          "Egress": true,
          "Protocol": "-1",
          "RuleAction": "allow",
          "RuleNumber": 100
        },
        {
          "CidrBlock": "0.0.0.0/0",
          "Egress": true,
          "Protocol": "-1",
          "RuleAction": "deny",

```

```
        "RuleNumber": 32767
    },
    {
        "CidrBlock": "0.0.0.0/0",
        "Egress": false,
        "Protocol": "-1",
        "RuleAction": "allow",
        "RuleNumber": 100
    },
    {
        "CidrBlock": "0.0.0.0/0",
        "Egress": false,
        "Protocol": "-1",
        "RuleAction": "deny",
        "RuleNumber": 32767
    }
],
"IsDefault": true,
"NetworkAclId": "acl-0ea1f54ca7EXAMPLE",
"Tags": [],
"VpcId": "vpc-06e4ab6c6cEXAMPLE",
"OwnerId": "111122223333"
},
{
    "Associations": [],
    "Entries": [
        {
            "CidrBlock": "0.0.0.0/0",
            "Egress": true,
            "Protocol": "-1",
            "RuleAction": "allow",
            "RuleNumber": 100
        },
        {
            "Egress": true,
            "Ipv6CidrBlock": ":::/0",
            "Protocol": "-1",
            "RuleAction": "allow",
            "RuleNumber": 101
        },
        {
            "CidrBlock": "0.0.0.0/0",
            "Egress": true,
            "Protocol": "-1",
```

```
        "RuleAction": "deny",
        "RuleNumber": 32767
    },
    {
        "Egress": true,
        "Ipv6CidrBlock": ":::/0",
        "Protocol": "-1",
        "RuleAction": "deny",
        "RuleNumber": 32768
    },
    {
        "CidrBlock": "0.0.0.0/0",
        "Egress": false,
        "Protocol": "-1",
        "RuleAction": "allow",
        "RuleNumber": 100
    },
    {
        "Egress": false,
        "Ipv6CidrBlock": ":::/0",
        "Protocol": "-1",
        "RuleAction": "allow",
        "RuleNumber": 101
    },
    {
        "CidrBlock": "0.0.0.0/0",
        "Egress": false,
        "Protocol": "-1",
        "RuleAction": "deny",
        "RuleNumber": 32767
    },
    {
        "Egress": false,
        "Ipv6CidrBlock": ":::/0",
        "Protocol": "-1",
        "RuleAction": "deny",
        "RuleNumber": 32768
    }
},
"IsDefault": true,
"NetworkAclId": "acl-0e2a78e4e2EXAMPLE",
"Tags": [],
"VpcId": "vpc-03914afb3eEXAMPLE",
"OwnerId": "111122223333"
```

```

    }
  ]
}

```

Untuk informasi selengkapnya, lihat [ACL Jaringan](#) di Panduan AWS Pengguna VPC.

- Untuk detail API, lihat [DescribeNetworkAcls](#) di Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini menjelaskan ACL jaringan tertentu.

```
Get-EC2NetworkAcl -NetworkAclId acl-12345678
```

Output:

```

Associations : {aclassoc-1a2b3c4d}
Entries      : {Amazon.EC2.Model.NetworkAclEntry,
               Amazon.EC2.Model.NetworkAclEntry}
IsDefault    : False
NetworkAclId : acl-12345678
Tags         : {Name}
VpcId        : vpc-12345678

```

Contoh 2: Contoh ini menjelaskan aturan untuk ACL jaringan yang ditentukan.

```
(Get-EC2NetworkAcl -NetworkAclId acl-12345678).Entries
```

Output:

```

CidrBlock    : 0.0.0.0/0
Egress       : True
IcmpTypeCode :
PortRange    :
Protocol     : -1
RuleAction   : deny
RuleNumber   : 32767
CidrBlock    : 0.0.0.0/0

```

```
Egress      : False
IcmpTypeCode :
PortRange   :
Protocol    : -1
RuleAction   : deny
RuleNumber   : 32767
```

Contoh 3: Contoh ini menjelaskan semua ACL jaringan Anda.

```
Get-EC2NetworkAcl
```

- Untuk detail API, lihat [DescribeNetworkAcls](#) di Referensi AWS Tools for PowerShell Cmdlet.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **DescribeNetworkInterfaceAttribute** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `DescribeNetworkInterfaceAttribute`.

CLI

AWS CLI

Untuk menggambarkan atribut lampiran dari antarmuka jaringan

Perintah contoh ini menjelaskan attachment atribut antarmuka jaringan yang ditentukan.

Perintah:

```
aws ec2 describe-network-interface-attribute --network-interface-id eni-686ea200
--attribute attachment
```

Output:

```
{
  "NetworkInterfaceId": "eni-686ea200",
  "Attachment": {
```

```
"Status": "attached",
"DeviceIndex": 0,
"AttachTime": "2015-05-21T20:02:20.000Z",
"InstanceId": "i-1234567890abcdef0",
"DeleteOnTermination": true,
"AttachmentId": "eni-attach-43348162",
"InstanceOwnerId": "123456789012"
}
}
```

Untuk menggambarkan atribut deskripsi antarmuka jaringan

Perintah contoh ini menjelaskan description atribut antarmuka jaringan yang ditentukan.

Perintah:

```
aws ec2 describe-network-interface-attribute --network-interface-id eni-686ea200
--attribute description
```

Output:

```
{
  "NetworkInterfaceId": "eni-686ea200",
  "Description": {
    "Value": "My description"
  }
}
```

Untuk menggambarkan atribut GroupSet dari antarmuka jaringan

Perintah contoh ini menjelaskan groupSet atribut antarmuka jaringan yang ditentukan.

Perintah:

```
aws ec2 describe-network-interface-attribute --network-interface-id eni-686ea200
--attribute groupSet
```

Output:

```
{
  "NetworkInterfaceId": "eni-686ea200",
  "Groups": [
```

```
{
  "GroupName": "my-security-group",
  "GroupId": "sg-903004f8"
}
]
```

Untuk menggambarkan sourceDestCheck atribut antarmuka jaringan

Perintah contoh ini menjelaskan sourceDestCheck atribut antarmuka jaringan yang ditentukan.

Perintah:

```
aws ec2 describe-network-interface-attribute --network-interface-id eni-686ea200
--attribute sourceDestCheck
```

Output:

```
{
  "NetworkInterfaceId": "eni-686ea200",
  "SourceDestCheck": {
    "Value": true
  }
}
```

- Untuk detail API, lihat [DescribeNetworkInterfaceAttribute](#) di Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini menjelaskan antarmuka jaringan yang ditentukan.

```
Get-EC2NetworkInterfaceAttribute -NetworkInterfaceId eni-12345678 -Attribute
Attachment
```

Output:

```
Attachment          : Amazon.EC2.Model.NetworkInterfaceAttachment
```


Contoh 2: Contoh ini menjelaskan antarmuka jaringan yang ditentukan.

```
Get-EC2NetworkInterfaceAttribute -NetworkInterfaceId eni-12345678 -Attribute
Description
```

Output:

```
Description          : My description
```

Contoh 3: Contoh ini menjelaskan antarmuka jaringan yang ditentukan.

```
Get-EC2NetworkInterfaceAttribute -NetworkInterfaceId eni-12345678 -Attribute
GroupSet
```

Output:

```
Groups              : {my-security-group}
```

Contoh 4: Contoh ini menjelaskan antarmuka jaringan yang ditentukan.

```
Get-EC2NetworkInterfaceAttribute -NetworkInterfaceId eni-12345678 -Attribute
SourceDestCheck
```

Output:

```
SourceDestCheck     : True
```

- Untuk detail API, lihat [DescribeNetworkInterfaceAttribute](#) di Referensi AWS Tools for PowerShell Cmdlet.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **DescribeNetworkInterfaces** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `DescribeNetworkInterfaces`.

CLI

AWS CLI

Untuk mendeskripsikan antarmuka jaringan Anda

Contoh ini menjelaskan semua antarmuka jaringan Anda.

Perintah:

```
aws ec2 describe-network-interfaces
```

Output:

```
{
  "NetworkInterfaces": [
    {
      "Status": "in-use",
      "MacAddress": "02:2f:8f:b0:cf:75",
      "SourceDestCheck": true,
      "VpcId": "vpc-a01106c2",
      "Description": "my network interface",
      "Association": {
        "PublicIp": "203.0.113.12",
        "AssociationId": "eipassoc-0fbb766a",
        "PublicDnsName": "ec2-203-0-113-12.compute-1.amazonaws.com",
        "IpOwnerId": "123456789012"
      },
      "NetworkInterfaceId": "eni-e5aa89a3",
      "PrivateIpAddresses": [
        {
          "PrivateDnsName": "ip-10-0-1-17.ec2.internal",
          "Association": {
            "PublicIp": "203.0.113.12",
            "AssociationId": "eipassoc-0fbb766a",
            "PublicDnsName":
"ec2-203-0-113-12.compute-1.amazonaws.com",
            "IpOwnerId": "123456789012"
          },
          "Primary": true,
          "PrivateIpAddress": "10.0.1.17"
        }
      ],
      "RequesterManaged": false,
```

```
    "Ipv6Addresses": [],
    "PrivateDnsName": "ip-10-0-1-17.ec2.internal",
    "AvailabilityZone": "us-east-1d",
    "Attachment": {
      "Status": "attached",
      "DeviceIndex": 1,
      "AttachTime": "2013-11-30T23:36:42.000Z",
      "InstanceId": "i-1234567890abcdef0",
      "DeleteOnTermination": false,
      "AttachmentId": "eni-attach-66c4350a",
      "InstanceOwnerId": "123456789012"
    },
    "Groups": [
      {
        "GroupName": "default",
        "GroupId": "sg-8637d3e3"
      }
    ],
    "SubnetId": "subnet-b61f49f0",
    "OwnerId": "123456789012",
    "TagSet": [],
    "PrivateIpAddress": "10.0.1.17"
  },
  {
    "Status": "in-use",
    "MacAddress": "02:58:f5:ef:4b:06",
    "SourceDestCheck": true,
    "VpcId": "vpc-a01106c2",
    "Description": "Primary network interface",
    "Association": {
      "PublicIp": "198.51.100.0",
      "IpOwnerId": "amazon"
    },
    "NetworkInterfaceId": "eni-f9ba99bf",
    "PrivateIpAddresses": [
      {
        "Association": {
          "PublicIp": "198.51.100.0",
          "IpOwnerId": "amazon"
        },
        "Primary": true,
        "PrivateIpAddress": "10.0.1.149"
      }
    ]
  }
],
```

```

    "RequesterManaged": false,
    "Ipv6Addresses": [],
    "AvailabilityZone": "us-east-1d",
    "Attachment": {
      "Status": "attached",
      "DeviceIndex": 0,
      "AttachTime": "2013-11-30T23:35:33.000Z",
      "InstanceId": "i-0598c7d356eba48d7",
      "DeleteOnTermination": true,
      "AttachmentId": "eni-attach-1b9db777",
      "InstanceOwnerId": "123456789012"
    },
    "Groups": [
      {
        "GroupName": "default",
        "GroupId": "sg-8637d3e3"
      }
    ],
    "SubnetId": "subnet-b61f49f0",
    "OwnerId": "123456789012",
    "TagSet": [],
    "PrivateIpAddress": "10.0.1.149"
  }
]
}

```

Contoh ini menjelaskan antarmuka jaringan yang memiliki tag dengan kunci Purpose dan nilainya Prod.

Perintah:

```
aws ec2 describe-network-interfaces --filters Name=tag:Purpose,Values=Prod
```

Output:

```

{
  "NetworkInterfaces": [
    {
      "Status": "available",
      "MacAddress": "12:2c:bd:f9:bf:17",
      "SourceDestCheck": true,
      "VpcId": "vpc-8941ebec",
      "Description": "ProdENI",

```

```

    "NetworkInterfaceId": "eni-b9a5ac93",
    "PrivateAddresses": [
      {
        "PrivateDnsName": "ip-10-0-1-55.ec2.internal",
        "Primary": true,
        "PrivateIpAddress": "10.0.1.55"
      },
      {
        "PrivateDnsName": "ip-10-0-1-117.ec2.internal",
        "Primary": false,
        "PrivateIpAddress": "10.0.1.117"
      }
    ],
    "RequesterManaged": false,
    "PrivateDnsName": "ip-10-0-1-55.ec2.internal",
    "AvailabilityZone": "us-east-1d",
    "Ipv6Addresses": [],
    "Groups": [
      {
        "GroupName": "MySG",
        "GroupId": "sg-905002f5"
      }
    ],
    "SubnetId": "subnet-31d6c219",
    "OwnerId": "123456789012",
    "TagSet": [
      {
        "Value": "Prod",
        "Key": "Purpose"
      }
    ],
    "PrivateIpAddress": "10.0.1.55"
  }
]
}

```

- Untuk detail API, lihat [DescribeNetworkInterfaces](#) di Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini menjelaskan antarmuka jaringan yang ditentukan.

```
Get-EC2NetworkInterface -NetworkInterfaceId eni-12345678
```

Output:

```
Association      :  
Attachment       : Amazon.EC2.Model.NetworkInterfaceAttachment  
AvailabilityZone : us-west-2c  
Description      :  
Groups          : {my-security-group}  
MacAddress       : 0a:e9:a6:19:4c:7f  
NetworkInterfaceId : eni-12345678  
OwnerId         : 123456789012  
PrivateDnsName  : ip-10-0-0-107.us-west-2.compute.internal  
PrivateIpAddress : 10.0.0.107  
PrivateAddresses : {ip-10-0-0-107.us-west-2.compute.internal}  
RequesterId     :  
RequesterManaged : False  
SourceDestCheck : True  
Status          : in-use  
SubnetId        : subnet-1a2b3c4d  
TagSet          : {}  
VpcId           : vpc-12345678
```

Contoh 2: Contoh ini menjelaskan semua antarmuka jaringan Anda.

```
Get-EC2NetworkInterface
```

- Untuk detail API, lihat [DescribeNetworkInterfaces](#) di Referensi AWS Tools for PowerShell Cmdlet.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **DescribePlacementGroups** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `DescribePlacementGroups`.

CLI

AWS CLI

Untuk menggambarkan grup penempatan Anda

Perintah contoh ini menjelaskan semua grup penempatan Anda.

Perintah:

```
aws ec2 describe-placement-groups
```

Output:

```
{
  "PlacementGroups": [
    {
      "GroupName": "my-cluster",
      "State": "available",
      "Strategy": "cluster"
    },
    ...
  ]
}
```

- Untuk detail API, lihat [DescribePlacementGroups](#) di Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini menjelaskan grup penempatan yang ditentukan.

```
Get-EC2PlacementGroup -GroupName my-placement-group
```

Output:

GroupName	State	Strategy
-----	-----	-----
my-placement-group	available	cluster

- Untuk detail API, lihat [DescribePlacementGroups](#) di Referensi AWS Tools for PowerShell Cmdlet.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **DescribePrefixLists** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `DescribePrefixLists`.

CLI

AWS CLI

Untuk menggambarkan daftar awalan

Contoh ini mencantumkan semua daftar awalan yang tersedia untuk wilayah tersebut.

Perintah:

```
aws ec2 describe-prefix-lists
```

Output:

```
{
  "PrefixLists": [
    {
      "PrefixListName": "com.amazonaws.us-east-1.s3",
      "Cidrs": [
        "54.231.0.0/17"
      ],
      "PrefixListId": "pl-63a5400a"
    }
  ]
}
```

- Untuk detail API, lihat [DescribePrefixLists](#) di Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini mengambil yang tersedia Layanan AWS dalam format daftar awalan untuk wilayah

```
Get-EC2PrefixList
```

Output:

```
Cidrs                                PrefixListId PrefixListName
-----                                -
{52.94.5.0/24, 52.119.240.0/21, 52.94.24.0/23} pl-6fa54006 com.amazonaws.eu-
west-1.dynamodb
{52.218.0.0/17, 54.231.128.0/19} pl-6da54004 com.amazonaws.eu-
west-1.s3
```

- Untuk detail API, lihat [DescribePrefixLists](#) di Referensi AWS Tools for PowerShell Cmdlet.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **DescribeRegions** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `DescribeRegions`.

C++

SDK untuk C++

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
Aws::EC2::EC2Client ec2Client(clientConfiguration);
```

```
Aws::EC2::Model::DescribeRegionsRequest request;
auto outcome = ec2Client.DescribeRegions(request);
bool result = true;
if (outcome.IsSuccess()) {
    std::cout << std::left <<
                std::setw(32) << "RegionName" <<
                std::setw(64) << "Endpoint" << std::endl;

    const auto &regions = outcome.GetResult().GetRegions();
    for (const auto &region: regions) {
        std::cout << std::left <<
                    std::setw(32) << region.GetRegionName() <<
                    std::setw(64) << region.GetEndpoint() << std::endl;
    }
}
else {
    std::cerr << "Failed to describe regions:" <<
                outcome.GetError().GetMessage() << std::endl;
    result = false;
}
```

- Untuk detail API, lihat [DescribeRegions](#) di Referensi AWS SDK for C++ API.

CLI

AWS CLI

Contoh 1: Untuk menjelaskan semua Wilayah yang diaktifkan

Contoh `describe-regions` berikut menjelaskan semua Wilayah yang diaktifkan untuk akun Anda.

```
aws ec2 describe-regions
```

Output:

```
{
  "Regions": [
    {
      "Endpoint": "ec2.eu-north-1.amazonaws.com",
      "RegionName": "eu-north-1",
```

```
    "OptInStatus": "opt-in-not-required"
  },
  {
    "Endpoint": "ec2.ap-south-1.amazonaws.com",
    "RegionName": "ap-south-1",
    "OptInStatus": "opt-in-not-required"
  },
  {
    "Endpoint": "ec2.eu-west-3.amazonaws.com",
    "RegionName": "eu-west-3",
    "OptInStatus": "opt-in-not-required"
  },
  {
    "Endpoint": "ec2.eu-west-2.amazonaws.com",
    "RegionName": "eu-west-2",
    "OptInStatus": "opt-in-not-required"
  },
  {
    "Endpoint": "ec2.eu-west-1.amazonaws.com",
    "RegionName": "eu-west-1",
    "OptInStatus": "opt-in-not-required"
  },
  {
    "Endpoint": "ec2.ap-northeast-3.amazonaws.com",
    "RegionName": "ap-northeast-3",
    "OptInStatus": "opt-in-not-required"
  },
  {
    "Endpoint": "ec2.ap-northeast-2.amazonaws.com",
    "RegionName": "ap-northeast-2",
    "OptInStatus": "opt-in-not-required"
  },
  {
    "Endpoint": "ec2.ap-northeast-1.amazonaws.com",
    "RegionName": "ap-northeast-1",
    "OptInStatus": "opt-in-not-required"
  },
  {
    "Endpoint": "ec2.sa-east-1.amazonaws.com",
    "RegionName": "sa-east-1",
    "OptInStatus": "opt-in-not-required"
  },
  {
    "Endpoint": "ec2.ca-central-1.amazonaws.com",
```

```
    "RegionName": "ca-central-1",
    "OptInStatus": "opt-in-not-required"
  },
  {
    "Endpoint": "ec2.ap-southeast-1.amazonaws.com",
    "RegionName": "ap-southeast-1",
    "OptInStatus": "opt-in-not-required"
  },
  {
    "Endpoint": "ec2.ap-southeast-2.amazonaws.com",
    "RegionName": "ap-southeast-2",
    "OptInStatus": "opt-in-not-required"
  },
  {
    "Endpoint": "ec2.eu-central-1.amazonaws.com",
    "RegionName": "eu-central-1",
    "OptInStatus": "opt-in-not-required"
  },
  {
    "Endpoint": "ec2.us-east-1.amazonaws.com",
    "RegionName": "us-east-1",
    "OptInStatus": "opt-in-not-required"
  },
  {
    "Endpoint": "ec2.us-east-2.amazonaws.com",
    "RegionName": "us-east-2",
    "OptInStatus": "opt-in-not-required"
  },
  {
    "Endpoint": "ec2.us-west-1.amazonaws.com",
    "RegionName": "us-west-1",
    "OptInStatus": "opt-in-not-required"
  },
  {
    "Endpoint": "ec2.us-west-2.amazonaws.com",
    "RegionName": "us-west-2",
    "OptInStatus": "opt-in-not-required"
  }
]
}
```

Untuk informasi selengkapnya, lihat [Wilayah dan Zona](#) di Panduan Pengguna Amazon EC2.

Contoh 2: Untuk menjelaskan Wilayah yang diaktifkan dengan titik akhir yang namanya berisi string tertentu

Contoh `describe-regions` berikut menjelaskan semua Wilayah yang telah Anda aktifkan yang memiliki string “us” di titik akhir.

```
aws ec2 describe-regions \  
  --filters "Name=endpoint,Values=*us*"
```

Output:

```
{  
  "Regions": [  
    {  
      "Endpoint": "ec2.us-east-1.amazonaws.com",  
      "RegionName": "us-east-1"  
    },  
    {  
      "Endpoint": "ec2.us-east-2.amazonaws.com",  
      "RegionName": "us-east-2"  
    },  
    {  
      "Endpoint": "ec2.us-west-1.amazonaws.com",  
      "RegionName": "us-west-1"  
    },  
    {  
      "Endpoint": "ec2.us-west-2.amazonaws.com",  
      "RegionName": "us-west-2"  
    }  
  ]  
}
```

Untuk informasi selengkapnya, lihat [Wilayah dan Zona](#) di Panduan Pengguna Amazon EC2.

Contoh 3: Untuk menjelaskan semua Wilayah

Contoh `describe-regions` berikut menjelaskan semua Wilayah yang tersedia, termasuk Wilayah yang dinonaktifkan.

```
aws ec2 describe-regions \  
  --all-regions
```

Output:

```
{
  "Regions": [
    {
      "Endpoint": "ec2.eu-north-1.amazonaws.com",
      "RegionName": "eu-north-1",
      "OptInStatus": "opt-in-not-required"
    },
    {
      "Endpoint": "ec2.ap-south-1.amazonaws.com",
      "RegionName": "ap-south-1",
      "OptInStatus": "opt-in-not-required"
    },
    {
      "Endpoint": "ec2.eu-west-3.amazonaws.com",
      "RegionName": "eu-west-3",
      "OptInStatus": "opt-in-not-required"
    },
    {
      "Endpoint": "ec2.eu-west-2.amazonaws.com",
      "RegionName": "eu-west-2",
      "OptInStatus": "opt-in-not-required"
    },
    {
      "Endpoint": "ec2.eu-west-1.amazonaws.com",
      "RegionName": "eu-west-1",
      "OptInStatus": "opt-in-not-required"
    },
    {
      "Endpoint": "ec2.ap-northeast-3.amazonaws.com",
      "RegionName": "ap-northeast-3",
      "OptInStatus": "opt-in-not-required"
    },
    {
      "Endpoint": "ec2.me-south-1.amazonaws.com",
      "RegionName": "me-south-1",
      "OptInStatus": "not-opted-in"
    },
    {
      "Endpoint": "ec2.ap-northeast-2.amazonaws.com",
      "RegionName": "ap-northeast-2",
      "OptInStatus": "opt-in-not-required"
    },
  ],
}
```

```
{
  "Endpoint": "ec2.ap-northeast-1.amazonaws.com",
  "RegionName": "ap-northeast-1",
  "OptInStatus": "opt-in-not-required"
},
{
  "Endpoint": "ec2.sa-east-1.amazonaws.com",
  "RegionName": "sa-east-1",
  "OptInStatus": "opt-in-not-required"
},
{
  "Endpoint": "ec2.ca-central-1.amazonaws.com",
  "RegionName": "ca-central-1",
  "OptInStatus": "opt-in-not-required"
},
{
  "Endpoint": "ec2.ap-east-1.amazonaws.com",
  "RegionName": "ap-east-1",
  "OptInStatus": "not-opted-in"
},
{
  "Endpoint": "ec2.ap-southeast-1.amazonaws.com",
  "RegionName": "ap-southeast-1",
  "OptInStatus": "opt-in-not-required"
},
{
  "Endpoint": "ec2.ap-southeast-2.amazonaws.com",
  "RegionName": "ap-southeast-2",
  "OptInStatus": "opt-in-not-required"
},
{
  "Endpoint": "ec2.eu-central-1.amazonaws.com",
  "RegionName": "eu-central-1",
  "OptInStatus": "opt-in-not-required"
},
{
  "Endpoint": "ec2.us-east-1.amazonaws.com",
  "RegionName": "us-east-1",
  "OptInStatus": "opt-in-not-required"
},
{
  "Endpoint": "ec2.us-east-2.amazonaws.com",
  "RegionName": "us-east-2",
  "OptInStatus": "opt-in-not-required"
}
```

```
    },
    {
      "Endpoint": "ec2.us-west-1.amazonaws.com",
      "RegionName": "us-west-1",
      "OptInStatus": "opt-in-not-required"
    },
    {
      "Endpoint": "ec2.us-west-2.amazonaws.com",
      "RegionName": "us-west-2",
      "OptInStatus": "opt-in-not-required"
    }
  ]
}
```

Untuk informasi selengkapnya, lihat [Wilayah dan Zona](#) di Panduan Pengguna Amazon EC2.

Contoh 4: Untuk mencantumkan nama Wilayah saja

Contoh `describe-regions` berikut menggunakan parameter `--query` untuk memfilter output dan hanya mengembalikan nama Wilayah sebagai teks.

```
aws ec2 describe-regions \
  --all-regions \
  --query "Regions[].{Name:RegionName}" \
  --output text
```

Output:

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



```
us-east-1
us-east-2
us-west-1
us-west-2
```

Untuk informasi selengkapnya, lihat [Wilayah dan Zona](#) di Panduan Pengguna Amazon EC2.

- Untuk detail API, lihat [DescribeRegions](#) di Referensi AWS CLI Perintah.

JavaScript

SDK untuk JavaScript (v3)

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
import { DescribeRegionsCommand } from "@aws-sdk/client-ec2";

import { client } from "../libs/client.js";

export const main = async () => {
  const command = new DescribeRegionsCommand({
    // By default this command will not show regions that require you to opt-in.
    // When AllRegions true even the regions that require opt-in will be
    returned.
    AllRegions: true,
    // You can omit the Filters property if you want to get all regions.
    Filters: [
      {
        Name: "region-name",
        // You can specify multiple values for a filter.
        // You can also use '*' as a wildcard. This will return all
        // of the regions that start with `us-east-`.
        Values: ["ap-southeast-4"],
      },
    ],
  });

  try {
```

```
const { Regions } = await client.send(command);
const regionsList = Regions.map((reg) => ` • ${reg.RegionName}`);
console.log("Found regions:");
console.log(regionsList.join("\n"));
} catch (err) {
  console.error(err);
}
};
```

- Untuk detail API, lihat [DescribeRegions](#) di Referensi AWS SDK for JavaScript API.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini menjelaskan wilayah yang tersedia untuk Anda.

```
Get-EC2Region
```

Output:

Endpoint	RegionName
-----	-----
ec2.eu-west-1.amazonaws.com	eu-west-1
ec2.ap-southeast-1.amazonaws.com	ap-southeast-1
ec2.ap-southeast-2.amazonaws.com	ap-southeast-2
ec2.eu-central-1.amazonaws.com	eu-central-1
ec2.ap-northeast-1.amazonaws.com	ap-northeast-1
ec2.us-east-1.amazonaws.com	us-east-1
ec2.sa-east-1.amazonaws.com	sa-east-1
ec2.us-west-1.amazonaws.com	us-west-1
ec2.us-west-2.amazonaws.com	us-west-2

- Untuk detail API, lihat [DescribeRegions](#) di Referensi AWS Tools for PowerShell Cmdlet.

Ruby

SDK untuk Ruby

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
require "aws-sdk-ec2"

# @param ec2_client [Aws::EC2::Client] An initialized EC2 client.
# @example
# list_regions_endpoints(Aws::EC2::Client.new(region: 'us-west-2'))
def list_regions_endpoints(ec2_client)
  result = ec2_client.describe_regions
  # Enable pretty printing.
  max_region_string_length = 16
  max_endpoint_string_length = 33
  # Print header.
  print "Region"
  print " " * (max_region_string_length - "Region".length)
  print " Endpoint\n"
  print "-" * max_region_string_length
  print " "
  print "-" * max_endpoint_string_length
  print "\n"
  # Print Regions and their endpoints.
  result.regions.each do |region|
    print region.region_name
    print " " * (max_region_string_length - region.region_name.length)
    print " "
    print region.endpoint
    print "\n"
  end
end

# Displays a list of Amazon Elastic Compute Cloud (Amazon EC2)
# Availability Zones available to you depending on the AWS Region
# of the Amazon EC2 client.
```

```
#
# @param ec2_client [Aws::EC2::Client] An initialized EC2 client.
# @example
# list_availability_zones(Aws::EC2::Client.new(region: 'us-west-2'))
def list_availability_zones(ec2_client)
  result = ec2_client.describe_availability_zones
  # Enable pretty printing.
  max_region_string_length = 16
  max_zone_string_length = 18
  max_state_string_length = 9
  # Print header.
  print "Region"
  print " " * (max_region_string_length - "Region".length)
  print " Zone"
  print " " * (max_zone_string_length - "Zone".length)
  print " State\n"
  print "-" * max_region_string_length
  print " "
  print "-" * max_zone_string_length
  print " "
  print "-" * max_state_string_length
  print "\n"
  # Print Regions, Availability Zones, and their states.
  result.availability_zones.each do |zone|
    print zone.region_name
    print " " * (max_region_string_length - zone.region_name.length)
    print " "
    print zone.zone_name
    print " " * (max_zone_string_length - zone.zone_name.length)
    print " "
    print zone.state
    # Print any messages for this Availability Zone.
    if zone.messages.count.positive?
      print "\n"
      puts " Messages for this zone:"
      zone.messages.each do |message|
        print "   #{message.message}\n"
      end
    end
    print "\n"
  end
end
end

# Example usage:
```

```

def run_me
  region = ""
  # Print usage information and then stop.
  if ARGV[0] == "--help" || ARGV[0] == "-h"
    puts "Usage:  ruby ec2-ruby-example-regions-availability-zones.rb REGION"
    # Replace us-west-2 with the AWS Region you're using for Amazon EC2.
    puts "Example: ruby ec2-ruby-example-regions-availability-zones.rb us-west-2"
    exit 1
  # If no values are specified at the command prompt, use these default values.
  # Replace us-west-2 with the AWS Region you're using for Amazon EC2.
  elsif ARGV.count.zero?
    region = "us-west-2"
  # Otherwise, use the values as specified at the command prompt.
  else
    region = ARGV[0]
  end

  ec2_client = Aws::EC2::Client.new(region: region)

  puts "AWS Regions for Amazon EC2 that are available to you:"
  list_regions_endpoints(ec2_client)
  puts "\n\nAmazon EC2 Availability Zones that are available to you for AWS
Region '#{region}':"
  list_availability_zones(ec2_client)
end

run_me if $PROGRAM_NAME == __FILE__

```

- Untuk detail API, lihat [DescribeRegions](#) di Referensi AWS SDK for Ruby API.

Rust

SDK untuk Rust

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```

async fn show_regions(client: &Client) -> Result<(), Error> {

```

```

let rsp = client.describe_regions().send().await?;

println!("Regions:");
for region in rsp.regions() {
    println!(" {}", region.region_name().unwrap());
}

Ok(())
}

```

- Untuk detail API, lihat [DescribeRegions](#) referensi AWS SDK for Rust API.

SAP ABAP

SDK untuk SAP ABAP

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```

TRY.
    oo_result = lo_ec2->describeregions( ) .
    oo_result is returned for testing purposes. "
    DATA(lt_regions) = oo_result->get_regions( ).
    MESSAGE 'Retrieved information about Regions.' TYPE 'I'.
    CATCH /aws1/cx_rt_service_generic INTO DATA(lo_exception).
    DATA(lv_error) = |"{ lo_exception->av_err_code }" - { lo_exception-
>av_err_msg }|.
    MESSAGE lv_error TYPE 'E'.
ENDTRY.

```

- Untuk detail API, lihat [DescribeRegions](#) di AWS SDK untuk referensi SAP ABAP API.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **DescribeRouteTables** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `DescribeRouteTables`.

CLI

AWS CLI

Untuk menggambarkan tabel rute Anda

`describe-route-tables` Contoh berikut mengambil rincian tentang tabel rute Anda

```
aws ec2 describe-route-tables
```

Output:

```
{
  "RouteTables": [
    {
      "Associations": [
        {
          "Main": true,
          "RouteTableAssociationId": "rtbassoc-0df3f54e06EXAMPLE",
          "RouteTableId": "rtb-09ba434c1bEXAMPLE"
        }
      ],
      "PropagatingVgws": [],
      "RouteTableId": "rtb-09ba434c1bEXAMPLE",
      "Routes": [
        {
          "DestinationCidrBlock": "10.0.0.0/16",
          "GatewayId": "local",
          "Origin": "CreateRouteTable",
          "State": "active"
        },
        {
          "DestinationCidrBlock": "0.0.0.0/0",
          "NatGatewayId": "nat-06c018cbd8EXAMPLE",
          "Origin": "CreateRoute",

```

```
        "State": "blackhole"
      }
    ],
    "Tags": [],
    "VpcId": "vpc-0065acced4EXAMPLE",
    "OwnerId": "111122223333"
  },
  {
    "Associations": [
      {
        "Main": true,
        "RouteTableAssociationId": "rtbassoc-9EXAMPLE",
        "RouteTableId": "rtb-a1eec7de"
      }
    ],
    "PropagatingVgws": [],
    "RouteTableId": "rtb-a1eec7de",
    "Routes": [
      {
        "DestinationCidrBlock": "172.31.0.0/16",
        "GatewayId": "local",
        "Origin": "CreateRouteTable",
        "State": "active"
      },
      {
        "DestinationCidrBlock": "0.0.0.0/0",
        "GatewayId": "igw-fEXAMPLE",
        "Origin": "CreateRoute",
        "State": "active"
      }
    ],
    "Tags": [],
    "VpcId": "vpc-3EXAMPLE",
    "OwnerId": "111122223333"
  },
  {
    "Associations": [
      {
        "Main": false,
        "RouteTableAssociationId": "rtbassoc-0b100c28b2EXAMPLE",
        "RouteTableId": "rtb-07a98f76e5EXAMPLE",
        "SubnetId": "subnet-0d3d002af8EXAMPLE"
      }
    ]
  },
],
```



```
"PropagatingVgws": [],
"RouteTableId": "rtb-07a98f76e5EXAMPLE",
"Routes": [
  {
    "DestinationCidrBlock": "10.0.0.0/16",
    "GatewayId": "local",
    "Origin": "CreateRouteTable",
    "State": "active"
  },
  {
    "DestinationCidrBlock": "0.0.0.0/0",
    "GatewayId": "igw-06cf664d80EXAMPLE",
    "Origin": "CreateRoute",
    "State": "active"
  }
],
"Tags": [],
"VpcId": "vpc-0065acced4EXAMPLE",
"OwnerId": "111122223333"
}
]
```

Untuk informasi selengkapnya, lihat [Bekerja dengan Tabel Rute](#) di Panduan Pengguna AWS VPC.

- Untuk detail API, lihat [DescribeRouteTables](#) di Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini menjelaskan semua tabel rute Anda.

```
Get-EC2RouteTable
```

Output:

```
DestinationCidrBlock    : 10.0.0.0/16
DestinationPrefixListId :
GatewayId               : local
InstanceId              :
```

```

InstanceOwnerId      :
NetworkInterfaceId  :
Origin               : CreateRouteTable
State                : active
VpcPeeringConnectionId :

DestinationCidrBlock : 0.0.0.0/0
DestinationPrefixListId :
GatewayId            : igw-1a2b3c4d
InstanceId           :
InstanceOwnerId      :
NetworkInterfaceId  :
Origin               : CreateRoute
State                : active
VpcPeeringConnectionId :

```

Contoh 2: Contoh ini mengembalikan rincian untuk tabel rute yang ditentukan.

```
Get-EC2RouteTable -RouteTableId rtb-1a2b3c4d
```

Contoh 3: Contoh ini menjelaskan tabel rute untuk VPC yang ditentukan.

```
Get-EC2RouteTable -Filter @{ Name="vpc-id"; Values="vpc-1a2b3c4d" }
```

Output:

```

Associations      : {rtbassoc-12345678}
PropagatingVgws  : {}
Routes            : {, }
RouteTableId     : rtb-1a2b3c4d
Tags              : {}
VpcId            : vpc-1a2b3c4d

```

- Untuk detail API, lihat [DescribeRouteTables](#) di Referensi AWS Tools for PowerShell Cmdlet.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan `DescribeScheduledInstanceAvailability` dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `DescribeScheduledInstanceAvailability`.

CLI

AWS CLI

Untuk menggambarkan jadwal yang tersedia

Contoh ini menjelaskan jadwal yang terjadi setiap minggu pada hari Minggu, dimulai pada tanggal yang ditentukan.

Perintah:

```
aws ec2 describe-scheduled-instance-availability --recurrence
Frequency=Weekly,Interval=1,OccurrenceDays=[1] --first-slot-start-time-range
EarliestTime=2016-01-31T00:00:00Z,LatestTime=2016-01-31T04:00:00Z
```

Output:

```
{
  "ScheduledInstanceAvailabilitySet": [
    {
      "AvailabilityZone": "us-west-2b",
      "TotalScheduledInstanceHours": 1219,
      "PurchaseToken": "eyJ2IjoiMSIsInMiOiJEsImMiOi...",
      "MinTermDurationInDays": 366,
      "AvailableInstanceCount": 20,
      "Recurrence": {
        "OccurrenceDaySet": [
          1
        ],
        "Interval": 1,
        "Frequency": "Weekly",
        "OccurrenceRelativeToEnd": false
      },
      "Platform": "Linux/UNIX",
      "FirstSlotStartTime": "2016-01-31T00:00:00Z",
```

```

    "MaxTermDurationInDays": 366,
    "SlotDurationInHours": 23,
    "NetworkPlatform": "EC2-VPC",
    "InstanceType": "c4.large",
    "HourlyPrice": "0.095"
  },
  ...
]
}

```

Untuk mempersempit hasil, Anda dapat menambahkan filter yang menentukan sistem operasi, jaringan, dan jenis instance.

Perintah:

```
--filter name=Platform, nilai = Linux/UNIX name=Network-platform, nilai = EC2-VPC
name=Instance-type, nilai = C4.large
```

- Untuk detail API, lihat [DescribeScheduledInstanceAvailability](#) di Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini menjelaskan jadwal yang terjadi setiap minggu pada hari Minggu, dimulai pada tanggal yang ditentukan.

```

Get-EC2ScheduledInstanceAvailability -Recurrence_Frequency
Weekly -Recurrence_Interval 1 -Recurrence_OccurrenceDay 1 -
FirstSlotStartTimeRange_EarliestTime 2016-01-31T00:00:00Z -
FirstSlotStartTimeRange_LatestTime 2016-01-31T04:00:00Z

```

Output:

```

AvailabilityZone           : us-west-2b
AvailableInstanceCount     : 20
FirstSlotStartTime         : 1/31/2016 8:00:00 AM
HourlyPrice                 : 0.095
InstanceType               : c4.large
MaxTermDurationInDays     : 366
MinTermDurationInDays     : 366

```

```

NetworkPlatform      : EC2-VPC
Platform             : Linux/UNIX
PurchaseToken        : eyJ2IjoiMSIsInMiOjEsImMiOi...
Recurrence           : Amazon.EC2.Model.ScheduledInstanceRecurrence
SlotDurationInHours  : 23
TotalScheduledInstanceHours : 1219
...

```

Contoh 2: Untuk mempersempit hasil, Anda dapat menambahkan filter untuk kriteria seperti sistem operasi, jaringan, dan jenis instance.

```
-Filter @{ Name="platform";Values="Linux/UNIX" },@{ Name="network-
platform";Values="EC2-VPC" },@{ Name="instance-type";Values="c4.large" }
```

- Untuk detail API, lihat [DescribeScheduledInstanceAvailability](#) di Referensi AWS Tools for PowerShell Cmdlet.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **DescribeScheduledInstances** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `DescribeScheduledInstances`.

CLI

AWS CLI

Untuk mendeskripsikan Instans Terjadwal Anda

Contoh ini menjelaskan Instance Terjadwal yang ditentukan.

Perintah:

```
aws ec2 describe-scheduled-instances --scheduled-instance-ids
sci-1234-1234-1234-1234-123456789012
```

Output:

```
{
  "ScheduledInstanceSet": [
    {
      "AvailabilityZone": "us-west-2b",
      "ScheduledInstanceId": "sci-1234-1234-1234-1234-123456789012",
      "HourlyPrice": "0.095",
      "CreateDate": "2016-01-25T21:43:38.612Z",
      "Recurrence": {
        "OccurrenceDaySet": [
          1
        ],
        "Interval": 1,
        "Frequency": "Weekly",
        "OccurrenceRelativeToEnd": false,
        "OccurrenceUnit": ""
      },
      "Platform": "Linux/UNIX",
      "TermEndDate": "2017-01-31T09:00:00Z",
      "InstanceCount": 1,
      "SlotDurationInHours": 32,
      "TermStartDate": "2016-01-31T09:00:00Z",
      "NetworkPlatform": "EC2-VPC",
      "TotalScheduledInstanceHours": 1696,
      "NextSlotStartTime": "2016-01-31T09:00:00Z",
      "InstanceType": "c4.large"
    }
  ]
}
```

Contoh ini menjelaskan semua Instans Terjadwal Anda.

Perintah:

```
aws ec2 describe-scheduled-instances
```

- Untuk detail API, lihat [DescribeScheduledInstances](#) di Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini menjelaskan Instance Terjadwal yang ditentukan.

```
Get-EC2ScheduledInstance -ScheduledInstanceId
sci-1234-1234-1234-1234-123456789012
```

Output:

```
AvailabilityZone      : us-west-2b
CreateDate            : 1/25/2016 1:43:38 PM
HourlyPrice           : 0.095
InstanceCount        : 1
InstanceType         : c4.large
NetworkPlatform      : EC2-VPC
NextSlotStartTime    : 1/31/2016 1:00:00 AM
Platform             : Linux/UNIX
PreviousSlotEndTime  :
Recurrence            : Amazon.EC2.Model.ScheduledInstanceRecurrence
ScheduledInstanceId   : sci-1234-1234-1234-1234-123456789012
SlotDurationInHours  : 32
TermEndDate          : 1/31/2017 1:00:00 AM
TermStartDate        : 1/31/2016 1:00:00 AM
TotalScheduledInstanceHours : 1696
```

Contoh 2: Contoh ini menjelaskan semua Instans Terjadwal Anda.

```
Get-EC2ScheduledInstance
```

- Untuk detail API, lihat [DescribeScheduledInstances](#) di Referensi AWS Tools for PowerShell Cmdlet.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **DescribeSecurityGroups** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `DescribeSecurityGroups`.

Contoh tindakan adalah kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Anda dapat melihat tindakan ini dalam konteks dalam contoh kode berikut:

- [Memulai instans](#)

.NET

AWS SDK for .NET

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
/// <summary>
/// Retrieve information for an Amazon EC2 security group.
/// </summary>
/// <param name="groupId">The Id of the Amazon EC2 security group.</param>
/// <returns>A list of security group information.</returns>
public async Task<List<SecurityGroup>> DescribeSecurityGroups(string groupId)
{
    var request = new DescribeSecurityGroupsRequest();
    var groupIds = new List<string> { groupId };
    request.GroupIds = groupIds;

    var response = await _amazonEC2.DescribeSecurityGroupsAsync(request);
    return response.SecurityGroups;
}

/// <summary>
/// Display the information returned by the call to
/// DescribeSecurityGroupsAsync.
/// </summary>
/// <param name="securityGroup">A list of security group information.</param>
public void DisplaySecurityGroupInfoAsync(SecurityGroup securityGroup)
{
    Console.WriteLine($"{securityGroup.GroupName}");
    Console.WriteLine("Ingress permissions:");
    securityGroup.IpPermissions.ForEach(permission =>
    {
        Console.WriteLine($"  \tFromPort: {permission.FromPort}");
        Console.WriteLine($"  \tIpProtocol: {permission.IpProtocol}");

        Console.WriteLine($"  \tIpv4Ranges: ");
        permission.Ipv4Ranges.ForEach(range =>
        { Console.WriteLine($"  \t{range.CidrIp} "); });
    });
}
```



```
        Console.WriteLine($"\\n\\tIpv6Ranges:");
        permission.Ipv6Ranges.ForEach(range =>
{ Console.Write($"{range.CidrIpv6} "); });

        Console.Write($"\\n\\tPrefixListIds: ");
        permission.PrefixListIds.ForEach(id => Console.Write($"{id.Id} "));

        Console.WriteLine($"\\n\\tTo Port: {permission.ToPort}");
    });
    Console.WriteLine("Egress permissions:");
    securityGroup.IpPermissionsEgress.ForEach(permission =>
    {
        Console.WriteLine($"\\tFromPort: {permission.FromPort}");
        Console.WriteLine($"\\tIpProtocol: {permission.IpProtocol}");

        Console.Write($"\\tIpv4Ranges: ");
        permission.Ipv4Ranges.ForEach(range =>
{ Console.Write($"{range.CidrIp} "); });

        Console.WriteLine($"\\n\\tIpv6Ranges:");
        permission.Ipv6Ranges.ForEach(range =>
{ Console.Write($"{range.CidrIpv6} "); });


        Console.Write($"\\n\\tPrefixListIds: ");
        permission.PrefixListIds.ForEach(id => Console.Write($"{id.Id} "));

        Console.WriteLine($"\\n\\tTo Port: {permission.ToPort}");
    });
}
```

- Untuk detail API, lihat [DescribeSecurityGroups](#) di Referensi AWS SDK for .NET API.

Bash

AWS CLI dengan skrip Bash

 Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
#####
# function ec2_describe_security_groups
#
# This function describes one or more Amazon Elastic Compute Cloud (Amazon EC2)
# security groups.
#
# Parameters:
#     -g security_group_id - The ID of the security group to describe
#     (optional).
#
# And:
#     0 - If successful.
#     1 - If it fails.
#####
function ec2_describe_security_groups() {
    local security_group_id response
    local option OPTARG # Required to use getopt command in a function.

    # bashsupport disable=BP5008
    function usage() {
        echo "function ec2_describe_security_groups"
        echo "Describes one or more Amazon Elastic Compute Cloud (Amazon EC2)
security groups."
        echo "  -g security_group_id - The ID of the security group to describe
(optional)."
        echo ""
    }

    # Retrieve the calling parameters.
    while getopt "g:h" option; do
        case "${option}" in
            g) security_group_id="${OPTARG}" ;;

```

```

    h)
    usage
    return 0
    ;;
\?)
    echo "Invalid parameter"
    usage
    return 1
    ;;
esac
done
export OPTIND=1

local query="SecurityGroups[*].[GroupName, GroupId, VpcId, IpPermissions[*].
[IpProtocol, FromPort, ToPort, IpRanges[*].CidrIp]]"

if [[ -n "$security_group_id" ]]; then
    response=$(aws ec2 describe-security-groups --group-ids "$security_group_id"
--query "${query}" --output text)
else
    response=$(aws ec2 describe-security-groups --query "${query}" --output text)
fi

local error_code=${?}

if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports describe-security-groups operation failed.
$response"
    return 1
fi

echo "$response"

return 0
}

```

Fungsi utilitas yang digunakan dalam contoh ini.

```

#####
# function errecho
#

```

```

# This function outputs everything sent to it to STDERR (standard error output).
#####
function errecho() {
    printf "%s\n" "$*" 1>&2
}

#####
# function aws_cli_error_log()
#
# This function is used to log the error messages from the AWS CLI.
#
# The function expects the following argument:
#     $1 - The error code returned by the AWS CLI.
#
# Returns:
#     0: - Success.
#
#####
function aws_cli_error_log() {
    local err_code=$1
    errecho "Error code : $err_code"
    if [ "$err_code" == 1 ]; then
        errecho " One or more S3 transfers failed."
    elif [ "$err_code" == 2 ]; then
        errecho " Command line failed to parse."
    elif [ "$err_code" == 130 ]; then
        errecho " Process received SIGINT."
    elif [ "$err_code" == 252 ]; then
        errecho " Command syntax invalid."
    elif [ "$err_code" == 253 ]; then
        errecho " The system environment or configuration was invalid."
    elif [ "$err_code" == 254 ]; then
        errecho " The service returned an error."
    elif [ "$err_code" == 255 ]; then
        errecho " 255 is a catch-all error."
    fi


    return 0
}

```

- Untuk detail API, lihat [DescribeSecurityGroups](#) di Referensi AWS CLI Perintah.

C++

SDK untuk C++

 Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
Aws::EC2::EC2Client ec2Client(clientConfiguration);
Aws::EC2::Model::DescribeSecurityGroupsRequest request;

if (!groupID.empty()) {
    request.AddGroupIds(groupID);
}

Aws::String nextToken;
do {
    if (!nextToken.empty()) {
        request.SetNextToken(nextToken);
    }

    auto outcome = ec2Client.DescribeSecurityGroups(request);
    if (outcome.IsSuccess()) {
        std::cout << std::left <<
            std::setw(32) << "Name" <<
            std::setw(30) << "GroupId" <<
            std::setw(30) << "VpcId" <<
            std::setw(64) << "Description" << std::endl;

        const std::vector<Aws::EC2::Model::SecurityGroup> &securityGroups =
            outcome.GetResult().GetSecurityGroups();

        for (const auto &securityGroup: securityGroups) {
            std::cout << std::left <<
                std::setw(32) << securityGroup.GetGroupName() <<
                std::setw(30) << securityGroup.GetGroupId() <<
                std::setw(30) << securityGroup.GetVpcId() <<
                std::setw(64) << securityGroup.GetDescription() <<
                std::endl;
        }
    }
}
```

```
    }
    else {
        std::cerr << "Failed to describe security groups:" <<
            outcome.GetError().GetMessage() << std::endl;
        return false;
    }

    nextToken = outcome.GetResult().GetNextToken();
} while (!nextToken.empty());
```

- Untuk detail API, lihat [DescribeSecurityGroups](#) di Referensi AWS SDK for C++ API.

CLI

AWS CLI

Contoh 1: Untuk menjelaskan grup keamanan

Contoh `describe-security-groups` berikut menjelaskan grup keamanan tertentu.

```
aws ec2 describe-security-groups \
    --group-ids sg-903004f8
```

Output:

```
{
  "SecurityGroups": [
    {
      "IpPermissionsEgress": [
        {
          "IpProtocol": "-1",
          "IpRanges": [
            {
              "CidrIp": "0.0.0.0/0"
            }
          ],
          "UserIdGroupPairs": [],
          "PrefixListIds": []
        }
      ],
      "Description": "My security group",
      "Tags": [
```

```

        {
            "Value": "SG1",
            "Key": "Name"
        }
    ],
    "IpPermissions": [
        {
            "IpProtocol": "-1",
            "IpRanges": [],
            "UserIdGroupPairs": [
                {
                    "UserId": "123456789012",
                    "GroupId": "sg-903004f8"
                }
            ],
            "PrefixListIds": []
        },
        {
            "PrefixListIds": [],
            "FromPort": 22,
            "IpRanges": [
                {
                    "Description": "Access from NY office",
                    "CidrIp": "203.0.113.0/24"
                }
            ],
            "ToPort": 22,
            "IpProtocol": "tcp",
            "UserIdGroupPairs": []
        }
    ],
    "GroupName": "MySecurityGroup",
    "VpcId": "vpc-1a2b3c4d",
    "OwnerId": "123456789012",
    "GroupId": "sg-903004f8",
}
]
}

```

Contoh 2: Untuk menjelaskan grup keamanan yang memiliki aturan khusus

`describe-security-groups` Contoh berikut menggunakan filter untuk lingkup hasil untuk kelompok keamanan yang memiliki aturan yang memungkinkan lalu lintas SSH (port 22) dan

aturan yang memungkinkan lalu lintas dari semua alamat (0.0.0.0/0). Contoh tersebut menggunakan parameter `--query` untuk hanya menampilkan nama grup keamanan. Grup keamanan harus cocok dengan semua filter yang akan dikembalikan dalam hasil; namun, satu aturan tidak harus cocok dengan semua filter. Misalnya, output mengembalikan grup keamanan dengan aturan yang memungkinkan lalu lintas SSH dari alamat IP tertentu dan aturan lain yang memungkinkan lalu lintas HTTP dari semua alamat.

```
aws ec2 describe-security-groups \
  --filters Name=ip-permission.from-port,Values=22 Name=ip-permission.to-
  port,Values=22 Name=ip-permission.cidr,Values='0.0.0.0/0' \
  --query "SecurityGroups[*].[GroupName]" \
  --output text
```

Output:

```
default
my-security-group
web-servers
launch-wizard-1
```

Contoh 3: Untuk menjelaskan grup keamanan berdasarkan tanda

Contoh `describe-security-groups` berikut menggunakan filter guna mencakup hasil untuk grup keamanan yang menyertakan `test` dalam nama grup keamanan dan yang memiliki tanda `Test=To-delete`. Contoh tersebut menggunakan parameter `--query` untuk hanya menampilkan nama dan ID grup keamanan.

```
aws ec2 describe-security-groups \
  --filters Name=group-name,Values=*test* Name=tag:Test,Values=To-delete \
  --query "SecurityGroups[*].{Name:GroupName,ID:GroupId}"
```

Output:

```
[
  {
    "Name": "testfornewinstance",
    "ID": "sg-33bb22aa"
  },
  {
    "Name": "newgrouptest",
    "ID": "sg-1a2b3c4d"
```



```
}  
]
```

Untuk contoh tambahan menggunakan filter tanda, lihat [Bekerja dengan tanda](#) di Panduan Pengguna Amazon EC2.

- Untuk detail API, lihat [DescribeSecurityGroups](#) di Referensi AWS CLI Perintah.

Java

SDK untuk Java 2.x

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
public static void describeSecurityGroups(Ec2Client ec2, String groupId) {  
    try {  
        DescribeSecurityGroupsRequest request =  
DescribeSecurityGroupsRequest.builder()  
            .groupIds(groupId)  
            .build();  
  
        // Use a paginator.  
DescribeSecurityGroupsIterable listGroups =  
ec2.describeSecurityGroupsPaginator(request);  
listGroups.stream()  
            .flatMap(r -> r.securityGroups().stream())  
            .forEach(group -> System.out  
                .println(" Group id: " +group.groupId() + " group name = " +  
group.groupName()));  
  
    } catch (Ec2Exception e) {  
        System.err.println(e.awsErrorDetails().errorMessage());  
        System.exit(1);  
    }  
}
```

- Untuk detail API, lihat [DescribeSecurityGroups](#) di Referensi AWS SDK for Java 2.x API.

JavaScript

SDK untuk JavaScript (v3)

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
import { DescribeSecurityGroupsCommand } from "@aws-sdk/client-ec2";

import { client } from "../libs/client.js";

// Log the details of a specific security group.
export const main = async () => {
  const command = new DescribeSecurityGroupsCommand({
    GroupIds: ["SECURITY_GROUP_ID"],
  });

  try {
    const { SecurityGroups } = await client.send(command);
    console.log(JSON.stringify(SecurityGroups, null, 2));
  } catch (err) {
    console.error(err);
  }
};
```

- Untuk detail API, lihat [DescribeSecurityGroups](#) di Referensi AWS SDK for JavaScript API.

Kotlin

SDK untuk Kotlin

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
suspend fun describeEC2SecurityGroups(groupId: String) {
    val request =
        DescribeSecurityGroupsRequest {
            groupIds = listOf(groupId)
        }

    Ec2Client { region = "us-west-2" }.use { ec2 ->

        val response = ec2.describeSecurityGroups(request)
        response.securityGroups?.forEach { group ->
            println("Found Security Group with id ${group.groupId}, vpc id
            ${group.vpcId} and description ${group.description}")
        }
    }
}
```

- Untuk detail API, lihat [DescribeSecurityGroups](#) di AWS SDK untuk referensi API Kotlin.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini menjelaskan grup keamanan yang ditentukan untuk VPC. Saat bekerja dengan grup keamanan milik VPC, Anda harus menggunakan ID grup keamanan (- GroupId parameter), bukan nama (- GroupName parameter), untuk mereferensikan grup.

```
Get-EC2SecurityGroup -GroupId sg-12345678
```

Output:

```
Description      : default VPC security group
GroupId          : sg-12345678
GroupName       : default
IpPermissions    : {Amazon.EC2.Model.IpPermission}
IpPermissionsEgress : {Amazon.EC2.Model.IpPermission}
OwnerId         : 123456789012
Tags            : {}
VpcId           : vpc-12345678
```

Contoh 2: Contoh ini menjelaskan grup keamanan yang ditentukan untuk EC2-Classic. Saat bekerja dengan grup keamanan untuk EC2-Classic Anda dapat menggunakan nama grup (- GroupName parameter) atau ID grup (- GroupId parameter) untuk mereferensikan grup keamanan.

```
Get-EC2SecurityGroup -GroupName my-security-group
```

Output:

```
Description      : my security group
GroupId          : sg-45678901
GroupName       : my-security-group
IpPermissions    : {Amazon.EC2.Model.IpPermission,
  Amazon.EC2.Model.IpPermission}
IpPermissionsEgress : {}
OwnerId         : 123456789012
Tags            : {}
VpcId           :
```

Contoh 3: Contoh ini mengambil semua grup keamanan untuk vpc-0fc1ff23456b789eb

```
Get-EC2SecurityGroup -Filter @{"Name"="vpc-id";"Values"="vpc-0fc1ff23456b789eb"}
```

- Untuk detail API, lihat [DescribeSecurityGroups](#) di Referensi AWS Tools for PowerShell Cmdlet.

Python

SDK untuk Python (Boto3)

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
class SecurityGroupWrapper:
    """Encapsulates Amazon Elastic Compute Cloud (Amazon EC2) security group
    actions."""
```

```
def __init__(self, ec2_resource, security_group=None):
    """
    :param ec2_resource: A Boto3 Amazon EC2 resource. This high-level
resource
                        is used to create additional high-level objects
                        that wrap low-level Amazon EC2 service actions.
    :param security_group: A Boto3 SecurityGroup object. This is a high-level
object
                        that wraps security group actions.
    """
    self.ec2_resource = ec2_resource
    self.security_group = security_group

    @classmethod
    def from_resource(cls):
        ec2_resource = boto3.resource("ec2")
        return cls(ec2_resource)

    def describe(self):
        """
        Displays information about the security group.
        """
        if self.security_group is None:
            logger.info("No security group to describe.")
            return

        try:
            print(f"Security group: {self.security_group.group_name}")
            print(f"\tID: {self.security_group.id}")
            print(f"\tVPC: {self.security_group.vpc_id}")
            if self.security_group.ip_permissions:
                print(f"Inbound permissions:")
                pp(self.security_group.ip_permissions)
        except ClientError as err:
            logger.error(
                "Couldn't get data for security group %s. Here's why: %s: %s",
                self.security_group.id,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
```

- Untuk detail API, lihat [DescribeSecurityGroups](#) di AWS SDK for Python (Boto3) Referensi API.

SAP ABAP

SDK untuk SAP ABAP

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```

TRY.
  DATA lt_group_ids TYPE /aws1/
cl_ec2groupidstrlist_w=>tt_groupidstringlist.
  APPEND NEW /aws1/cl_ec2groupidstrlist_w( iv_value = iv_group_id ) TO
  lt_group_ids.
  oo_result = lo_ec2->describesecuritygroups( it_groupids = lt_group_ids ).
  " oo_result is returned for testing purposes. "
  DATA(lt_security_groups) = oo_result->get_securitygroups( ).
  MESSAGE 'Retrieved information about security groups.' TYPE 'I'.
  CATCH /aws1/cx_rt_service_generic INTO DATA(lo_exception).
  DATA(lv_error) = |"{ lo_exception->av_err_code }" - { lo_exception-
>av_err_msg }|.
  MESSAGE lv_error TYPE 'E'.
ENDTRY.

```

- Untuk detail API, lihat [DescribeSecurityGroups](#) di AWS SDK untuk referensi SAP ABAP API.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **DescribeSnapshotAttribute** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `DescribeSnapshotAttribute`.

CLI

AWS CLI

Untuk mendeskripsikan atribut snapshot untuk snapshot

`describe-snapshot-attribute` Contoh berikut mencantumkan akun yang digunakan untuk berbagi snapshot.

```
aws ec2 describe-snapshot-attribute \  
  --snapshot-id snap-01234567890abcdef \  
  --attribute createVolumePermission
```

Output:

```
{  
  "SnapshotId": "snap-01234567890abcdef",  
  "CreateVolumePermissions": [  
    {  
      "UserId": "123456789012"  
    }  
  ]  
}
```

Untuk informasi selengkapnya, lihat [Membagikan snapshot Amazon EBS di Panduan Pengguna](#) Amazon Elastic Compute Cloud.

- Untuk detail API, lihat [DescribeSnapshotAttribute](#) di Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini menjelaskan atribut tertentu dari snapshot yang ditentukan.

```
Get-EC2SnapshotAttribute -SnapshotId snap-12345678 -Attribute ProductCodes
```

Output:

```
CreateVolumePermissions    ProductCodes    SnapshotId
```

```

-----
{}          {}          snap-12345678

```

Contoh 2: Contoh ini menjelaskan atribut tertentu dari snapshot yang ditentukan.

```

(Get-EC2SnapshotAttribute -SnapshotId snap-12345678 -Attribute
  CreateVolumePermission).CreateVolumePermissions

```

Output:

```

Group    UserId
-----
all

```

- Untuk detail API, lihat [DescribeSnapshotAttribute](#) di Referensi AWS Tools for PowerShell Cmdlet.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **DescribeSnapshots** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `DescribeSnapshots`.

CLI

AWS CLI

Contoh 1: Untuk menjelaskan snapshot

Contoh `describe-snapshots` berikut menjelaskan snapshot tertentu.

```

aws ec2 describe-snapshots \
  --snapshot-ids snap-1234567890abcdef0

```

Output:

```

{

```



```

    "Snapshots": [
      {
        "Description": "This is my snapshot",
        "Encrypted": false,
        "VolumeId": "vol-049df61146c4d7901",
        "State": "completed",
        "VolumeSize": 8,
        "StartTime": "2019-02-28T21:28:32.000Z",
        "Progress": "100%",
        "OwnerId": "012345678910",
        "SnapshotId": "snap-01234567890abcdef",
        "Tags": [
          {
            "Key": "Stack",
            "Value": "test"
          }
        ]
      }
    ]
  }
}

```

Untuk informasi selengkapnya, lihat [Snapshot Amazon EBS](#) di Panduan Pengguna Amazon EC2.

Contoh 2: Untuk menjelaskan snapshot berdasarkan filter

`describe-snapshots` Contoh berikut menggunakan filter untuk cakupan hasil ke snapshot yang dimiliki oleh AWS akun Anda yang berada dalam pending status. Contoh tersebut menggunakan parameter `--query` untuk hanya menampilkan ID snapshot dan waktu mulai snapshot.

```

aws ec2 describe-snapshots \
  --owner-ids self \
  --filters Name=status,Values=pending \
  --query "Snapshots[*].{ID:SnapshotId,Time:StartTime}"

```

Output:

```

[
  {
    "ID": "snap-1234567890abcdef0",
    "Time": "2019-08-04T12:48:18.000Z"
  }
]

```

```
    },  
    {  
      "ID": "snap-066877671789bd71b",  
      "Time": "2019-08-04T02:45:16.000Z"  
    },  
    ...  
  ]
```

Contoh `describe-snapshots` berikut menggunakan filter guna mencakup hasil untuk snapshot yang dibuat dari volume tertentu. Contoh tersebut menggunakan parameter `--query` untuk hanya menampilkan ID snapshot.

```
aws ec2 describe-snapshots \  
  --filters Name=volume-id,Values=049df61146c4d7901 \  
  --query "Snapshots[*].[SnapshotId]" \  
  --output text
```

Output:

```
snap-1234567890abcdef0  
snap-08637175a712c3fb9  
...
```

Untuk contoh tambahan menggunakan filter, lihat [Mencantumkan dan memfilter sumber daya Anda](#) di Panduan Pengguna Amazon EC2.

Contoh 3: Untuk menjelaskan snapshot berdasarkan tanda

Contoh `describe-snapshots` berikut menggunakan filter tanda guna mencakup hasil untuk snapshot yang memiliki tanda `Stack=Prod`.

```
aws ec2 describe-snapshots \  
  --filters Name=tag:Stack,Values=prod
```

Untuk contoh output `describe-snapshots`, lihat Contoh 1.

Untuk contoh tambahan menggunakan filter tanda, lihat [Bekerja dengan tanda](#) di Panduan Pengguna Amazon EC2.

Contoh 4: Untuk menjelaskan snapshot berdasarkan usia

`describe-snapshots` Contoh berikut menggunakan ekspresi JMESPath untuk menggambarkan semua snapshot yang dibuat oleh AWS akun Anda sebelum tanggal yang ditentukan. Contoh tersebut hanya menampilkan ID snapshot.

```
aws ec2 describe-snapshots \  
  --owner-ids 012345678910 \  
  --query "Snapshots[?(StartTime<='2020-03-31')].[SnapshotId]"
```

Untuk contoh tambahan menggunakan filter, lihat [Mencantumkan dan memfilter sumber daya Anda](#) di Panduan Pengguna Amazon EC2.

Contoh 5: Untuk melihat snapshot yang diarsipkan saja

Contoh `describe-snapshots` berikut hanya mencantumkan snapshot yang disimpan di tingkat arsip.

```
aws ec2 describe-snapshots \  
  --filters "Name=storage-tier,Values=archive"
```

Output:

```
{  
  "Snapshots": [  
    {  
      "Description": "Snap A",  
      "Encrypted": false,  
      "VolumeId": "vol-01234567890aaaaaa",  
      "State": "completed",  
      "VolumeSize": 8,  
      "StartTime": "2021-09-07T21:00:00.000Z",  
      "Progress": "100%",  
      "OwnerId": "123456789012",  
      "SnapshotId": "snap-01234567890aaaaaa",  
      "StorageTier": "archive",  
      "Tags": []  
    },  
  ]  
}
```

Untuk informasi selengkapnya, buka [Lihat snapshot yang diarsipkan](#) di Panduan Pengguna Amazon Elastic Compute Cloud.

- Untuk detail API, lihat [DescribeSnapshots](#) di Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini menjelaskan snapshot yang ditentukan.

```
Get-EC2Snapshot -SnapshotId snap-12345678
```

Output:

```
DataEncryptionKeyId :  
Description          : Created by CreateImage(i-1a2b3c4d) for ami-12345678 from  
vol-12345678  
Encrypted           : False  
KmsKeyId            :  
OwnerAlias          :  
OwnerId             : 123456789012  
Progress            : 100%  
SnapshotId          : snap-12345678  
StartTime           : 10/23/2014 6:01:28 AM  
State               : completed  
StateMessage        :  
Tags                : {}  
VolumeId            : vol-12345678  
VolumeSize          : 8
```

Contoh 2: Contoh ini menjelaskan snapshot yang memiliki tag 'Nama'.

```
Get-EC2Snapshot | ? { $_.Tags.Count -gt 0 -and $_.Tags.Key -eq "Name" }
```

Contoh 3: Contoh ini menjelaskan snapshot yang memiliki tag 'Nama' dengan nilai 'TestValue'.

```
Get-EC2Snapshot | ? { $_.Tags.Count -gt 0 -and $_.Tags.Key -eq "Name" -and  
$_.Tags.Value -eq "TestValue" }
```

Contoh 4: Contoh ini menjelaskan semua snapshot Anda.

```
Get-EC2Snapshot -Owner self
```

- Untuk detail API, lihat [DescribeSnapshots](#) di Referensi AWS Tools for PowerShell Cmdlet.

Rust

SDK untuk Rust

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturan dan menjalankannya di [Repositori Contoh Kode AWS](#).

Menampilkan status snapshot.

```
async fn show_state(client: &Client, id: &str) -> Result<(), Error> {
    let resp = client
        .describe_snapshots()
        .filters(Filter::builder().name("snapshot-id").values(id).build())
        .send()
        .await?;

    println!(
        "State: {}",
        resp.snapshots().first().unwrap().state().unwrap().as_ref()
    );

    Ok(())
}
```

```
async fn show_snapshots(client: &Client) -> Result<(), Error> {
    // "self" represents your account ID.
    // You can list the snapshots for any account by replacing
    // "self" with that account ID.
    let resp = client.describe_snapshots().owner_ids("self").send().await?;
    let snapshots = resp.snapshots();
    let length = snapshots.len();

    for snapshot in snapshots {
        println!(
            "ID:          {}",

```

```
        snapshot.snapshot_id().unwrap_or_default()
    );
    println!(
        "Description: {}",
        snapshot.description().unwrap_or_default()
    );
    println!("State:      {}", snapshot.state().unwrap().as_ref());
    println!();
}

println!();
println!("Found {} snapshot(s)", length);
println!();

Ok(())
}
```

- Untuk detail API, lihat [DescribeSnapshots](#) referensi AWS SDK for Rust API.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **DescribeSpotDatafeedSubscription** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `DescribeSpotDatafeedSubscription`.

CLI

AWS CLI

Untuk mendeskripsikan langganan datafeed Instance Spot untuk sebuah akun

Perintah contoh ini menjelaskan umpan data untuk akun.

Perintah:

```
aws ec2 describe-spot-datafeed-subscription
```

Output:

```
{
  "SpotDatafeedSubscription": {
    "OwnerId": "123456789012",
    "Prefix": "spotdata",
    "Bucket": "my-s3-bucket",
    "State": "Active"
  }
}
```

- Untuk detail API, lihat [DescribeSpotDatafeedSubscription](#) di Referensi AWS CLI Perintah.

PowerShell**Alat untuk PowerShell**

Contoh 1: Contoh ini menjelaskan umpan data instans Spot Anda.

```
Get-EC2SpotDatafeedSubscription
```

Output:

```
Bucket   : my-s3-bucket
Fault    :
OwnerId  : 123456789012
Prefix   : spotdata
State    : Active
```

- Untuk detail API, lihat [DescribeSpotDatafeedSubscription](#) di Referensi AWS Tools for PowerShell Cmdlet.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan `DescribeSpotFleetInstances` dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `DescribeSpotFleetInstances`.

CLI

AWS CLI

Untuk menggambarkan Instans Spot yang terkait dengan armada Spot

Perintah contoh ini mencantumkan instance Spot yang terkait dengan armada Spot yang ditentukan.

Perintah:

```
aws ec2 describe-spot-fleet-instances --spot-fleet-request-id sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE
```

Output:

```
{
  "ActiveInstances": [
    {
      "InstanceId": "i-1234567890abcdef0",
      "InstanceType": "m3.medium",
      "SpotInstanceRequestId": "sir-08b93456"
    },
    ...
  ],
  "SpotFleetRequestId": "sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE"
}
```

- Untuk detail API, lihat [DescribeSpotFleetInstances](#) di Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini menjelaskan instance yang terkait dengan permintaan armada Spot yang ditentukan.

```
Get-EC2SpotFleetInstance -SpotFleetRequestId sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE
```

Output:

InstanceId	InstanceType	SpotInstanceRequestId
-----	-----	-----
i-f089262a	c3.large	sir-12345678
i-7e8b24a4	c3.large	sir-87654321

- Untuk detail API, lihat [DescribeSpotFleetInstances](#) di Referensi AWS Tools for PowerShell Cmdlet.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **DescribeSpotFleetRequestHistory** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `DescribeSpotFleetRequestHistory`.

CLI

AWS CLI

Untuk menggambarkan sejarah armada Spot

Perintah contoh ini mengembalikan riwayat untuk armada Spot yang ditentukan mulai pada waktu yang ditentukan.

Perintah:

```
aws ec2 describe-spot-fleet-request-history --spot-fleet-request-id sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE --start-time 2015-05-26T00:00:00Z
```

Contoh keluaran berikut menunjukkan keberhasilan peluncuran dua Instans Spot untuk armada Spot.

Output:

```
{
  "HistoryRecords": [
    {
      "Timestamp": "2015-05-26T23:17:20.697Z",
```

```

    "EventInformation": {
      "EventSubType": "submitted"
    },
    "EventType": "fleetRequestChange"
  },
  {
    "Timestamp": "2015-05-26T23:17:20.873Z",
    "EventInformation": {
      "EventSubType": "active"
    },
    "EventType": "fleetRequestChange"
  },
  {
    "Timestamp": "2015-05-26T23:21:21.712Z",
    "EventInformation": {
      "InstanceId": "i-1234567890abcdef0",
      "EventSubType": "launched"
    },
    "EventType": "instanceChange"
  },
  {
    "Timestamp": "2015-05-26T23:21:21.816Z",
    "EventInformation": {
      "InstanceId": "i-1234567890abcdef1",
      "EventSubType": "launched"
    },
    "EventType": "instanceChange"
  }
],
"SpotFleetRequestId": "sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE",
"NextToken": "CpHNsscimcV5oH7bSbub03CI2Qms5+ypNpNm
+53MNIr0YcXAkp0xFlfKf91yVxSExmbtma3awYxMFzNA663ZskT0AhtJ6TCb2Z8bQC2EnZgyELbymtWPfpZ1ZbauV
+P+TfG1WxWWB/Vr5dk5d4LfdgA/DRAHUrYgxzrEXAMPLE=",
"StartTime": "2015-05-26T00:00:00Z"
}

```

- Untuk detail API, lihat [DescribeSpotFleetRequestHistory](#) di Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini menjelaskan riwayat permintaan armada Spot yang ditentukan.

```
Get-EC2SpotFleetRequestHistory -SpotFleetRequestId sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE -StartTime 2015-12-26T00:00:00Z
```

Output:

```
HistoryRecords      : {Amazon.EC2.Model.HistoryRecord,
  Amazon.EC2.Model.HistoryRecord...}
LastEvaluatedTime  : 12/26/2015 8:29:11 AM
NextToken          :
SpotFleetRequestId : sfr-088bc5f1-7e7b-451a-bd13-757f10672b93
StartTime          : 12/25/2015 8:00:00 AM
```

```
(Get-EC2SpotFleetRequestHistory -SpotFleetRequestId sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE -StartTime 2015-12-26T00:00:00Z).HistoryRecords
```

Output:

EventInformation	EventType	Timestamp
-----	-----	-----
Amazon.EC2.Model.EventInformation	fleetRequestChange	12/26/2015 8:23:33 AM
Amazon.EC2.Model.EventInformation	fleetRequestChange	12/26/2015 8:23:33 AM
Amazon.EC2.Model.EventInformation	fleetRequestChange	12/26/2015 8:23:33 AM
Amazon.EC2.Model.EventInformation	launched	12/26/2015 8:25:34 AM
Amazon.EC2.Model.EventInformation	launched	12/26/2015 8:25:05 AM

- Untuk detail API, lihat [DescribeSpotFleetRequestHistory](#) di Referensi AWS Tools for PowerShell Cmdlet.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **DescribeSpotFleetRequests** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `DescribeSpotFleetRequests`.

CLI

AWS CLI

Untuk menjelaskan permintaan armada Spot Anda

Contoh ini menjelaskan semua permintaan armada Spot Anda.

Perintah:

```
aws ec2 describe-spot-fleet-requests
```

Output:

```
{
  "SpotFleetRequestConfigs": [
    {
      "SpotFleetRequestId": "sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE",
      "SpotFleetRequestConfig": {
        "TargetCapacity": 20,
        "LaunchSpecifications": [
          {
            "EbsOptimized": false,
            "NetworkInterfaces": [
              {
                "SubnetId": "subnet-a61dafcf",
                "DeviceIndex": 0,
                "DeleteOnTermination": false,
                "AssociatePublicIpAddress": true,
                "SecondaryPrivateIpAddressCount": 0
              }
            ],
            "InstanceType": "cc2.8xlarge",
            "ImageId": "ami-1a2b3c4d"
          },
          {
            "EbsOptimized": false,
            "NetworkInterfaces": [
              {
                "SubnetId": "subnet-a61dafcf",
                "DeviceIndex": 0,
                "DeleteOnTermination": false,
                "AssociatePublicIpAddress": true,
                "SecondaryPrivateIpAddressCount": 0
              }
            ]
          }
        ]
      }
    }
  ]
}
```

```

        }
      ],
      "InstanceType": "r3.8xlarge",
      "ImageId": "ami-1a2b3c4d"
    }
  ],
  "SpotPrice": "0.05",
  "IamFleetRole": "arn:aws:iam::123456789012:role/my-spot-fleet-role"
},
"SpotFleetRequestState": "active"
},
{
  "SpotFleetRequestId": "sfr-306341ed-9739-402e-881b-ce47bEXAMPLE",
  "SpotFleetRequestConfig": {
    "TargetCapacity": 20,
    "LaunchSpecifications": [
      {
        "EbsOptimized": false,
        "NetworkInterfaces": [
          {
            "SubnetId": "subnet-6e7f829e",
            "DeviceIndex": 0,
            "DeleteOnTermination": false,
            "AssociatePublicIpAddress": true,
            "SecondaryPrivateIpAddressCount": 0
          }
        ],
        "InstanceType": "m3.medium",
        "ImageId": "ami-1a2b3c4d"
      }
    ],
    "SpotPrice": "0.05",
    "IamFleetRole": "arn:aws:iam::123456789012:role/my-spot-fleet-role"
  },
  "SpotFleetRequestState": "active"
}
]
}

```

Untuk menjelaskan permintaan armada Spot

Contoh ini menjelaskan permintaan armada Spot yang ditentukan.

Perintah:

```
aws ec2 describe-spot-fleet-requests --spot-fleet-request-ids sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE
```

Output:

```
{
  "SpotFleetRequestConfigs": [
    {
      "SpotFleetRequestId": "sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE",
      "SpotFleetRequestConfig": {
        "TargetCapacity": 20,
        "LaunchSpecifications": [
          {
            "EbsOptimized": false,
            "NetworkInterfaces": [
              {
                "SubnetId": "subnet-a61dafcf",
                "DeviceIndex": 0,
                "DeleteOnTermination": false,
                "AssociatePublicIpAddress": true,
                "SecondaryPrivateIpAddressCount": 0
              }
            ],
            "InstanceType": "cc2.8xlarge",
            "ImageId": "ami-1a2b3c4d"
          },
          {
            "EbsOptimized": false,
            "NetworkInterfaces": [
              {
                "SubnetId": "subnet-a61dafcf",
                "DeviceIndex": 0,
                "DeleteOnTermination": false,
                "AssociatePublicIpAddress": true,
                "SecondaryPrivateIpAddressCount": 0
              }
            ],
            "InstanceType": "r3.8xlarge",
            "ImageId": "ami-1a2b3c4d"
          }
        ],
        "SpotPrice": "0.05",
        "IamFleetRole": "arn:aws:iam::123456789012:role/my-spot-fleet-role"
      }
    }
  ]
}
```

```

    },
    "SpotFleetRequestState": "active"
  }
]
}

```

- Untuk detail API, lihat [DescribeSpotFleetRequests](#) di Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini menjelaskan permintaan armada Spot yang ditentukan.

```
Get-EC2SpotFleetRequest -SpotFleetRequestId sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE | format-list
```

Output:

```

ConfigData           : Amazon.EC2.Model.SpotFleetRequestConfigData
CreateTime           : 12/26/2015 8:23:33 AM
SpotFleetRequestId   : sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE
SpotFleetRequestState : active

```

Contoh 2: Contoh ini menjelaskan semua permintaan armada Spot Anda.

```
Get-EC2SpotFleetRequest
```

- Untuk detail API, lihat [DescribeSpotFleetRequests](#) di Referensi AWS Tools for PowerShell Cmdlet.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **DescribeSpotInstanceRequests** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `DescribeSpotInstanceRequests`.

CLI

AWS CLI

Contoh 1: Untuk mendeskripsikan permintaan Instans Spot

`describe-spot-instance-requests` Contoh berikut menjelaskan permintaan Instans Spot yang ditentukan.

```
aws ec2 describe-spot-instance-requests \  
  --spot-instance-request-ids sir-08b93456
```

Output:

```
{  
  "SpotInstanceRequests": [  
    {  
      "CreateTime": "2018-04-30T18:14:55.000Z",  
      "InstanceId": "i-1234567890abcdef1",  
      "LaunchSpecification": {  
        "InstanceType": "t2.micro",  
        "ImageId": "ami-003634241a8fcdec0",  
        "KeyName": "my-key-pair",  
        "SecurityGroups": [  
          {  
            "GroupName": "default",  
            "GroupId": "sg-e38f24a7"  
          }  
        ],  
        "BlockDeviceMappings": [  
          {  
            "DeviceName": "/dev/sda1",  
            "Ebs": {  
              "DeleteOnTermination": true,  
              "SnapshotId": "snap-0e54a519c999adbbd",  
              "VolumeSize": 8,  
              "VolumeType": "standard",  
              "Encrypted": false  
            }  
          }  
        ],  
        "NetworkInterfaces": [  
          {
```



```

        "DeleteOnTermination": true,
        "DeviceIndex": 0,
        "SubnetId": "subnet-049df61146c4d7901"
    }
  ],
  "Placement": {
    "AvailabilityZone": "us-east-2b",
    "Tenancy": "default"
  },
  "Monitoring": {
    "Enabled": false
  }
},
"LaunchedAvailabilityZone": "us-east-2b",
"ProductDescription": "Linux/UNIX",
"SpotInstanceRequestId": "sir-08b93456",
"SpotPrice": "0.010000"
"State": "active",
"Status": {
  "Code": "fulfilled",
  "Message": "Your Spot request is fulfilled.",
  "UpdateTime": "2018-04-30T18:16:21.000Z"
},
"Tags": [],
"Type": "one-time",
"InstanceInterruptionBehavior": "terminate"
}
]
}

```

Contoh 2: Untuk menjelaskan permintaan Instans Spot berdasarkan filter

`describe-spot-instance-requests` Contoh berikut menggunakan filter untuk membuat cakupan hasil ke permintaan Instance Spot dengan jenis instans yang ditentukan di Availability Zone yang ditentukan. Contoh menggunakan `--query` parameter untuk menampilkan hanya ID instance.

```

aws ec2 describe-spot-instance-requests \
  --filters Name=launch.instance-type,Values=m3.medium Name=launched-
  availability-zone,Values=us-east-2a \
  --query "SpotInstanceRequests[*].[InstanceId]" \
  --output text

```

Output:

```
i-057750d42936e468a
i-001efd250faaa6ffa
i-027552a73f021f3bd
...
```

Untuk contoh tambahan menggunakan filter, lihat [Mencantumkan dan memfilter sumber daya Anda](#) di Panduan Pengguna Amazon Elastic Compute Cloud.

Contoh 3: Untuk mendeskripsikan permintaan Instans Spot berdasarkan tag

`describe-spot-instance-requests` Contoh berikut menggunakan filter tag untuk cakupan hasil ke permintaan Spot Instance yang memiliki tag `cost-center=cc123`.

```
aws ec2 describe-spot-instance-requests \
  --filters Name=tag:cost-center,Values=cc123
```

Untuk contoh output `describe-spot-instance-requests`, lihat Contoh 1.

Untuk contoh tambahan menggunakan filter tanda, lihat [Bekerja dengan tanda](#) di Panduan Pengguna Amazon EC2.

- Untuk detail API, lihat [DescribeSpotInstanceRequests](#) di Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini menjelaskan permintaan instance Spot yang ditentukan.

```
Get-EC2SpotInstanceRequest -SpotInstanceRequestId sir-12345678
```

Output:

```
ActualBlockHourlyPrice      :
AvailabilityZoneGroup       :
BlockDurationMinutes        : 0
CreateTime                  : 4/8/2015 2:51:33 PM
Fault                       :
InstanceId                   : i-12345678
```

```
LaunchedAvailabilityZone : us-west-2b
LaunchGroup              :
LaunchSpecification      : Amazon.EC2.Model.LaunchSpecification
ProductDescription       : Linux/UNIX
SpotInstanceRequestId   : sir-12345678
SpotPrice                : 0.020000
State                   : active
Status                  : Amazon.EC2.Model.SpotInstanceStatus
Tags                    : {Name}
Type                    : one-time
```

Contoh 2: Contoh ini menjelaskan semua permintaan instans Spot Anda.

```
Get-EC2SpotInstanceRequest
```

- Untuk detail API, lihat [DescribeSpotInstanceRequests](#) di Referensi AWS Tools for PowerShell Cmdlet.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **DescribeSpotPriceHistory** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `DescribeSpotPriceHistory`.

CLI

AWS CLI

Untuk menggambarkan sejarah harga Spot

Perintah contoh ini mengembalikan riwayat Harga Spot untuk instance `m1.xlarge` untuk hari tertentu di bulan Januari.

Perintah:

```
aws ec2 describe-spot-price-history --instance-types m1.xlarge --start-time
2014-01-06T07:08:09 --end-time 2014-01-06T08:09:10
```

Output:

```
{
  "SpotPriceHistory": [
    {
      "Timestamp": "2014-01-06T07:10:55.000Z",
      "ProductDescription": "SUSE Linux",
      "InstanceType": "m1.xlarge",
      "SpotPrice": "0.087000",
      "AvailabilityZone": "us-west-1b"
    },
    {
      "Timestamp": "2014-01-06T07:10:55.000Z",
      "ProductDescription": "SUSE Linux",
      "InstanceType": "m1.xlarge",
      "SpotPrice": "0.087000",
      "AvailabilityZone": "us-west-1c"
    },
    {
      "Timestamp": "2014-01-06T05:42:36.000Z",
      "ProductDescription": "SUSE Linux (Amazon VPC)",
      "InstanceType": "m1.xlarge",
      "SpotPrice": "0.087000",
      "AvailabilityZone": "us-west-1a"
    },
    ...
  ]
}
```

Untuk menggambarkan sejarah harga Spot untuk Linux/Unix Amazon VPC

Perintah contoh ini mengembalikan histori Harga Spot untuk instans VPC Amazon Linux/Unix m1.xlarge untuk hari tertentu di bulan Januari.

Perintah:

```
aws ec2 describe-spot-price-history --instance-types m1.xlarge --product-
description "Linux/UNIX (Amazon VPC)" --start-time 2014-01-06T07:08:09 --end-time
2014-01-06T08:09:10
```

Output:

```
{
  "SpotPriceHistory": [
    {
```

```

    "Timestamp": "2014-01-06T04:32:53.000Z",
    "ProductDescription": "Linux/UNIX (Amazon VPC)",
    "InstanceType": "m1.xlarge",
    "SpotPrice": "0.080000",
    "AvailabilityZone": "us-west-1a"
  },
  {
    "Timestamp": "2014-01-05T11:28:26.000Z",
    "ProductDescription": "Linux/UNIX (Amazon VPC)",
    "InstanceType": "m1.xlarge",
    "SpotPrice": "0.080000",
    "AvailabilityZone": "us-west-1c"
  }
]
}

```

- Untuk detail API, lihat [DescribeSpotPriceHistory](#) di Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini mendapatkan 10 entri terakhir dalam riwayat harga Spot untuk jenis instans tertentu dan Availability Zone. Perhatikan bahwa nilai yang ditentukan untuk AvailabilityZone parameter - harus valid untuk nilai wilayah yang diberikan ke parameter - Region cmdlet (tidak ditampilkan dalam contoh) atau ditetapkan sebagai default di shell. Perintah contoh ini mengasumsikan wilayah default 'us-west-2' telah disetel di lingkungan.

```

Get-EC2SpotPriceHistory -InstanceType c3.large -AvailabilityZone us-west-2a -
MaxResult 10

```

Output:

```

AvailabilityZone    : us-west-2a
InstanceType       : c3.large
Price              : 0.017300
ProductDescription : Linux/UNIX (Amazon VPC)
Timestamp          : 12/25/2015 7:39:49 AM

AvailabilityZone    : us-west-2a
InstanceType       : c3.large
Price              : 0.017200

```

```
ProductDescription : Linux/UNIX (Amazon VPC)
Timestamp          : 12/25/2015 7:38:29 AM

AvailabilityZone   : us-west-2a
InstanceType      : c3.large
Price             : 0.017300
ProductDescription : Linux/UNIX (Amazon VPC)
Timestamp          : 12/25/2015 6:57:13 AM
...
```

- Untuk detail API, lihat [DescribeSpotPriceHistory](#) di Referensi AWS Tools for PowerShell Cmdlet.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **DescribeSubnets** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `DescribeSubnets`.

Contoh tindakan adalah kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Anda dapat melihat tindakan ini dalam konteks dalam contoh kode berikut:

- [Membangun dan mengelola layanan yang tangguh](#)

.NET

AWS SDK for .NET

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
/// <summary>
/// Get all the subnets for a Vpc in a set of availability zones.
/// </summary>
/// <param name="vpcId">The Id of the Vpc.</param>
```

```
/// <param name="availabilityZones">The list of availability zones.</param>
/// <returns>The collection of subnet objects.</returns>
public async Task<List<Subnet>> GetAllVpcSubnetsForZones(string vpcId,
List<string> availabilityZones)
{
    var subnets = new List<Subnet>();
    var subnetPaginator = _amazonEc2.Paginators.DescribeSubnets(
        new DescribeSubnetsRequest()
        {
            Filters = new List<Amazon.EC2.Model.Filter>()
            {
                new ("vpc-id", new List<string>() { vpcId}),
                new ("availability-zone", availabilityZones),
                new ("default-for-az", new List<string>() { "true" })
            }
        });

    // Get the entire list using the paginator.
    await foreach (var subnet in subnetPaginator.Subnets)
    {
        subnets.Add(subnet);
    }

    return subnets;
}
```

- Untuk detail API, lihat [DescribeSubnets](#) di Referensi AWS SDK for .NET API.

CLI

AWS CLI

Contoh 1: Untuk menjelaskan semua subnet Anda

Contoh `describe-subnets` berikut menampilkan detail subnet Anda.

```
aws ec2 describe-subnets
```

Output:

```
{
  "Subnets": [
```

```

    {
      "AvailabilityZone": "us-east-1d",
      "AvailabilityZoneId": "use1-az2",
      "AvailableIpAddressCount": 4089,
      "CidrBlock": "172.31.80.0/20",
      "DefaultForAz": true,
      "MapPublicIpOnLaunch": false,
      "MapCustomerOwnedIpOnLaunch": true,
      "State": "available",
      "SubnetId": "subnet-0bb1c79de3EXAMPLE",
      "VpcId": "vpc-0ee975135dEXAMPLE",
      "OwnerId": "111122223333",
      "AssignIpv6AddressOnCreation": false,
      "Ipv6CidrBlockAssociationSet": [],
      "CustomerOwnedIpv4Pool": "pool-2EXAMPLE",
      "SubnetArn": "arn:aws:ec2:us-east-2:111122223333:subnet/
subnet-0bb1c79de3EXAMPLE",
      "EnableDns64": false,
      "Ipv6Native": false,
      "PrivateDnsNameOptionsOnLaunch": {
        "HostnameType": "ip-name",
        "EnableResourceNameDnsARecord": false,
        "EnableResourceNameDnsAAAARecord": false
      }
    },
    {
      "AvailabilityZone": "us-east-1d",
      "AvailabilityZoneId": "use1-az2",
      "AvailableIpAddressCount": 4089,
      "CidrBlock": "172.31.80.0/20",
      "DefaultForAz": true,
      "MapPublicIpOnLaunch": true,
      "MapCustomerOwnedIpOnLaunch": false,
      "State": "available",
      "SubnetId": "subnet-8EXAMPLE",
      "VpcId": "vpc-3EXAMPLE",
      "OwnerId": "111122223333",
      "AssignIpv6AddressOnCreation": false,
      "Ipv6CidrBlockAssociationSet": [],
      "Tags": [
        {
          "Key": "Name",
          "Value": "MySubnet"
        }
      ]
    }
  ]
}

```



```

    ],
    "SubnetArn": "arn:aws:ec2:us-east-1:111122223333:subnet/
subnet-8EXAMPLE",
    "EnableDns64": false,
    "Ipv6Native": false,
    "PrivateDnsNameOptionsOnLaunch": {
      "HostnameType": "ip-name",
      "EnableResourceNameDnsARecord": false,
      "EnableResourceNameDnsAAAARecord": false
    }
  }
]
}

```

Untuk informasi selengkapnya, lihat [Bekerja dengan VPC dan Subnet](#) di Panduan Pengguna VPC AWS .

Contoh 2: Untuk menjelaskan subnet dari VPC tertentu

Contoh `describe-subnets` berikut menggunakan filter guna mengambil detail untuk subnet dari VPC tertentu.

```

aws ec2 describe-subnets \
  --filters "Name=vpc-id,Values=vpc-3EXAMPLE"

```

Output:

```

{
  "Subnets": [
    {
      "AvailabilityZone": "us-east-1d",
      "AvailabilityZoneId": "use1-az2",
      "AvailableIpAddressCount": 4089,
      "CidrBlock": "172.31.80.0/20",
      "DefaultForAz": true,
      "MapPublicIpOnLaunch": true,
      "MapCustomerOwnedIpOnLaunch": false,
      "State": "available",
      "SubnetId": "subnet-8EXAMPLE",
      "VpcId": "vpc-3EXAMPLE",
      "OwnerId": "111122223333",
      "AssignIpv6AddressOnCreation": false,
      "Ipv6CidrBlockAssociationSet": [],

```

```

    "Tags": [
      {
        "Key": "Name",
        "Value": "MySubnet"
      }
    ],
    "SubnetArn": "arn:aws:ec2:us-east-1:111122223333:subnet/
subnet-8EXAMPLE",
    "EnableDns64": false,
    "Ipv6Native": false,
    "PrivateDnsNameOptionsOnLaunch": {
      "HostnameType": "ip-name",
      "EnableResourceNameDnsARecord": false,
      "EnableResourceNameDnsAAAARecord": false
    }
  }
]
}

```

Untuk informasi selengkapnya, lihat [Bekerja dengan VPC dan Subnet](#) di Panduan Pengguna VPC AWS .

Contoh 3: Untuk menjelaskan subnet dengan tanda tertentu

Contoh `describe-subnets` berikut menggunakan filter untuk mengambil detail subnet tersebut dengan tanda `CostCenter=123` dan parameter `--query` untuk menampilkan ID subnet dari subnet dengan tanda ini.

```

aws ec2 describe-subnets \
  --filters "Name=tag:CostCenter,Values=123" \
  --query "Subnets[*].SubnetId" \
  --output text

```

Output:

```

subnet-0987a87c8b37348ef
subnet-02a95061c45f372ee
subnet-03f720e7de2788d73

```

Untuk informasi selengkapnya, lihat [Bekerja dengan VPC dan Subnet](#) di Panduan Pengguna Amazon VPC.

- Untuk detail API, lihat [DescribeSubnets](#) di Referensi AWS CLI Perintah.

JavaScript

SDK untuk JavaScript (v3)

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
const client = new EC2Client({});
const { Subnets } = await client.send(
  new DescribeSubnetsCommand({
    Filters: [
      { Name: "vpc-id", Values: [state.defaultVpc] },
      { Name: "availability-zone", Values: state.availabilityZoneNames },
      { Name: "default-for-az", Values: ["true"] },
    ],
  }),
);
```

- Untuk detail API, lihat [DescribeSubnets](#) di Referensi AWS SDK for JavaScript API.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini menjelaskan subnet yang ditentukan.

```
Get-EC2Subnet -SubnetId subnet-1a2b3c4d
```

Output:

```
AvailabilityZone      : us-west-2c
AvailableIpAddressCount : 251
CidrBlock             : 10.0.0.0/24
DefaultForAz         : False
MapPublicIpOnLaunch  : False
State                 : available
SubnetId              : subnet-1a2b3c4d
```

```
Tags          : {}
VpcId         : vpc-12345678
```

Contoh 2: Contoh ini menjelaskan semua subnet Anda.

```
Get-EC2Subnet
```

- Untuk detail API, lihat [DescribeSubnets](#) di Referensi AWS Tools for PowerShell Cmdlet.

Python

SDK untuk Python (Boto3)

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
class AutoScaler:
    """
    Encapsulates Amazon EC2 Auto Scaling and EC2 management actions.
    """

    def __init__(
        self,
        resource_prefix,
        inst_type,
        ami_param,
        autoscaling_client,
        ec2_client,
        ssm_client,
        iam_client,
    ):
        """
        :param resource_prefix: The prefix for naming AWS resources that are
        created by this class.
        :param inst_type: The type of EC2 instance to create, such as t3.micro.
        :param ami_param: The Systems Manager parameter used to look up the AMI
        that is
                               created.
```

```

:param autoscaling_client: A Boto3 EC2 Auto Scaling client.
:param ec2_client: A Boto3 EC2 client.
:param ssm_client: A Boto3 Systems Manager client.
:param iam_client: A Boto3 IAM client.
"""

self.inst_type = inst_type
self.ami_param = ami_param
self.autoscaling_client = autoscaling_client
self.ec2_client = ec2_client
self.ssm_client = ssm_client
self.iam_client = iam_client
self.launch_template_name = f"{resource_prefix}-template"
self.group_name = f"{resource_prefix}-group"
self.instance_policy_name = f"{resource_prefix}-pol"
self.instance_role_name = f"{resource_prefix}-role"
self.instance_profile_name = f"{resource_prefix}-prof"
self.bad_creds_policy_name = f"{resource_prefix}-bc-pol"
self.bad_creds_role_name = f"{resource_prefix}-bc-role"
self.bad_creds_profile_name = f"{resource_prefix}-bc-prof"
self.key_pair_name = f"{resource_prefix}-key-pair"

def get_subnets(self, vpc_id, zones):
    """
    Gets the default subnets in a VPC for a specified list of Availability
    Zones.

    :param vpc_id: The ID of the VPC to look up.
    :param zones: The list of Availability Zones to look up.
    :return: The list of subnets found.
    """
    try:
        response = self.ec2_client.describe_subnets(
            Filters=[
                {"Name": "vpc-id", "Values": [vpc_id]},
                {"Name": "availability-zone", "Values": zones},
                {"Name": "default-for-az", "Values": ["true"]},
            ]
        )
        subnets = response["Subnets"]
        log.info("Found %s subnets for the specified zones.", len(subnets))
    except ClientError as err:
        raise AutoScalerError(f"Couldn't get subnets: {err}")
    else:

```

```
return subnets
```

- Untuk detail API, lihat [DescribeSubnets](#) di AWS SDK for Python (Boto3) Referensi API.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **DescribeTags** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `DescribeTags`.

CLI

AWS CLI

Contoh 1: Untuk mendeskripsikan semua tag untuk satu sumber daya

`describe-tags` Contoh berikut menjelaskan tag untuk contoh tertentu.

```
aws ec2 describe-tags \
  --filters "Name=resource-id,Values=i-1234567890abcdef8"
```

Output:

```
{
  "Tags": [
    {
      "ResourceType": "instance",
      "ResourceId": "i-1234567890abcdef8",
      "Value": "Test",
      "Key": "Stack"
    },
    {
      "ResourceType": "instance",
      "ResourceId": "i-1234567890abcdef8",
      "Value": "Beta Server",
      "Key": "Name"
    }
  ]
}
```

```
]
}
```

Contoh 2: Untuk mendeskripsikan semua tag untuk jenis sumber daya

`describe-tags` Contoh berikut menjelaskan tag untuk volume Anda.

```
aws ec2 describe-tags \
  --filters "Name=resource-type,Values=volume"
```

Output:

```
{
  "Tags": [
    {
      "ResourceType": "volume",
      "ResourceId": "vol-1234567890abcdef0",
      "Value": "Project1",
      "Key": "Purpose"
    },
    {
      "ResourceType": "volume",
      "ResourceId": "vol-049df61146c4d7901",
      "Value": "Logs",
      "Key": "Purpose"
    }
  ]
}
```

Contoh 3: Untuk mendeskripsikan semua tag Anda

`describe-tags` Contoh berikut menjelaskan tag untuk semua sumber daya Anda.

```
aws ec2 describe-tags
```

Contoh 4: Untuk mendeskripsikan tag untuk sumber daya Anda berdasarkan kunci tag

`describe-tags` Contoh berikut menjelaskan tag untuk sumber daya Anda yang memiliki tag dengan kunci `Stack`.

```
aws ec2 describe-tags \
```

```
--filters Name=key,Values=Stack
```

Output:

```
{
  "Tags": [
    {
      "ResourceType": "volume",
      "ResourceId": "vol-027552a73f021f3b",
      "Value": "Production",
      "Key": "Stack"
    },
    {
      "ResourceType": "instance",
      "ResourceId": "i-1234567890abcdef8",
      "Value": "Test",
      "Key": "Stack"
    }
  ]
}
```

Contoh 5: Untuk mendeskripsikan tag untuk sumber daya Anda berdasarkan kunci tag dan nilai tag

`describe-tags` Contoh berikut menjelaskan tag untuk sumber daya Anda yang memiliki `tagStack=Test`.

```
aws ec2 describe-tags \
  --filters Name=key,Values=Stack Name=value,Values=Test
```

Output:

```
{
  "Tags": [
    {
      "ResourceType": "image",
      "ResourceId": "ami-3ac336533f021f3bd",
      "Value": "Test",
      "Key": "Stack"
    },
    {
      "ResourceType": "instance",
```



```

        "ResourceId": "i-1234567890abcdef8",
        "Value": "Test",
        "Key": "Stack"
    }
]
}

```

`describe-tags` Contoh berikut menggunakan sintaks alternatif untuk menggambarkan sumber daya dengan tag `Stack=Test`.

```

aws ec2 describe-tags \
  --filters "Name=tag:Stack,Values=Test"

```

`describe-tags` Contoh berikut menjelaskan tag untuk semua instance Anda yang memiliki tag dengan kunci `Purpose` dan tanpa nilai.

```

aws ec2 describe-tags \
  --filters "Name=resource-type,Values=instance" "Name=key,Values=Purpose"
  "Name=value,Values="

```

Output:

```

{
  "Tags": [
    {
      "ResourceType": "instance",
      "ResourceId": "i-1234567890abcdef5",
      "Value": null,
      "Key": "Purpose"
    }
  ]
}

```

- Untuk detail API, lihat [DescribeTags](#) di Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini mengambil tag untuk 'image' tipe sumber daya

```
Get-EC2Tag -Filter @{"Name"="resource-type";Values="image"}
```

Output:

Key	ResourceId	ResourceType	Value
---	-----	-----	----
Name	ami-0a123b4ccb567a8ea	image	Win7-Imported
auto-delete	ami-0a123b4ccb567a8ea	image	never

Contoh 2: Contoh ini mengambil semua tag untuk semua sumber daya dan mengelompokkannya berdasarkan jenis sumber daya

```
Get-EC2Tag | Group-Object resourcetype
```

Output:

Count	Name	Group
-----	----	-----
9	subnet	{Amazon.EC2.Model.TagDescription, Amazon.EC2.Model.TagDescription, Amazon.EC2.Model.TagDescription, Amazon.EC2.Model.TagDescription...}
53	instance	{Amazon.EC2.Model.TagDescription, Amazon.EC2.Model.TagDescription, Amazon.EC2.Model.TagDescription, Amazon.EC2.Model.TagDescription...}
3	route-table	{Amazon.EC2.Model.TagDescription, Amazon.EC2.Model.TagDescription, Amazon.EC2.Model.TagDescription}
5	security-group	{Amazon.EC2.Model.TagDescription, Amazon.EC2.Model.TagDescription, Amazon.EC2.Model.TagDescription, Amazon.EC2.Model.TagDescription...}
30	volume	{Amazon.EC2.Model.TagDescription, Amazon.EC2.Model.TagDescription, Amazon.EC2.Model.TagDescription, Amazon.EC2.Model.TagDescription...}
1	internet-gateway	{Amazon.EC2.Model.TagDescription}
3	network-interface	{Amazon.EC2.Model.TagDescription, Amazon.EC2.Model.TagDescription, Amazon.EC2.Model.TagDescription}
4	elastic-ip	{Amazon.EC2.Model.TagDescription, Amazon.EC2.Model.TagDescription, Amazon.EC2.Model.TagDescription, Amazon.EC2.Model.TagDescription}
1	dhcp-options	{Amazon.EC2.Model.TagDescription}
2	image	{Amazon.EC2.Model.TagDescription, Amazon.EC2.Model.TagDescription}

```
3 vpc {Amazon.EC2.Model.TagDescription,
Amazon.EC2.Model.TagDescription, Amazon.EC2.Model.TagDescription}
```

Contoh 3: Contoh ini menampilkan semua sumber daya dengan tag 'auto-delete' dengan nilai 'no' untuk wilayah tertentu

```
Get-EC2Tag -Region eu-west-1 -Filter @{{Name="tag:auto-delete";Values="no"}}
```

Output:

Key	ResourceId	ResourceType	Value
auto-delete	i-0f1bce234d5dd678b	instance	no
auto-delete	vol-01d234aa5678901a2	volume	no
auto-delete	vol-01234bfb5def6f7b8	volume	no
auto-delete	vol-01ccb23f4c5e67890	volume	no

Contoh 4: Contoh ini memperoleh semua sumber daya dengan tag 'hapus otomatis' dengan nilai 'no' dan filter lebih lanjut di pipa berikutnya untuk mengurai hanya jenis sumber daya 'instance' dan akhirnya membuat tag 'ThisInstance' untuk setiap sumber daya instance dengan nilai menjadi id instance itu sendiri

```
Get-EC2Tag -Region eu-west-1 -Filter @{{Name="tag:auto-delete";Values="no"}
| Where-Object ResourceType -eq "instance" | ForEach-Object {New-EC2Tag -
ResourceId $_.ResourceId -Tag @{{Key="ThisInstance";Value=$_.ResourceId}}
```

Contoh 5: Contoh ini mengambil tag untuk semua sumber daya instance serta kunci 'Nama' dan menampilkannya dalam format tabel

```
Get-EC2Tag -Filter @{{Name="resource-
type";Values="instance"},@{{Name="key";Values="Name"}} | Select-Object ResourceId,
@{{Name="Name-Tag";Expression={$PSItem.Value}}} | Format-Table -AutoSize
```

Output:

ResourceId	Name-Tag
i-012e3cb4df567e1aa	jump1
i-01c23a45d6fc7a89f	repro-3

- Untuk detail API, lihat [DescribeTags](#) di Referensi AWS Tools for PowerShell Cmdlet.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **DescribeVolumeAttribute** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `DescribeVolumeAttribute`.

CLI

AWS CLI

Untuk menggambarkan atribut volume

Perintah contoh ini menjelaskan `autoEnableIo` atribut volume dengan `vol-049df61146c4d7901`.

Perintah:

```
aws ec2 describe-volume-attribute --volume-id vol-049df61146c4d7901 --attribute autoEnableIO
```

Output:

```
{
  "AutoEnableIO": {
    "Value": false
  },
  "VolumeId": "vol-049df61146c4d7901"
}
```

- Untuk detail API, lihat [DescribeVolumeAttribute](#) di Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini menjelaskan atribut tertentu dari volume yang ditentukan.

```
Get-EC2VolumeAttribute -VolumeId vol-12345678 -Attribute AutoEnableIO
```

Output:

AutoEnableIO	ProductCodes	VolumeId
-----	-----	-----
False	{}	vol-12345678

- Untuk detail API, lihat [DescribeVolumeAttribute](#) di Referensi AWS Tools for PowerShell Cmdlet.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **DescribeVolumeStatus** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `DescribeVolumeStatus`.

CLI

AWS CLI

Untuk menggambarkan status satu volume

Perintah contoh ini menjelaskan status untuk volume `vol-1234567890abcdef0`.

Perintah:

```
aws ec2 describe-volume-status --volume-ids vol-1234567890abcdef0
```

Output:

```
{
  "VolumeStatuses": [
    {
      "VolumeStatus": {
        "Status": "ok",
        "Details": [
          {
            "Status": "passed",
```

```

        "Name": "io-enabled"
      },
      {
        "Status": "not-applicable",
        "Name": "io-performance"
      }
    ]
  },
  "AvailabilityZone": "us-east-1a",
  "VolumeId": "vol-1234567890abcdef0",
  "Actions": [],
  "Events": []
}
]
}

```

Untuk menggambarkan status volume yang terganggu

Perintah contoh ini menjelaskan status untuk semua volume yang terganggu. Dalam contoh output ini, tidak ada volume yang terganggu.

Perintah:

```
aws ec2 describe-volume-status --filters Name=volume-status.status,Values=impaired
```

Output:

```
{
  "VolumeStatuses": []
}
```

Jika Anda memiliki volume dengan pemeriksaan status gagal (status terganggu), lihat Bekerja dengan Volume yang Terganggu di Panduan Pengguna Amazon EC2.

- Untuk detail API, lihat [DescribeVolumeStatus](#) di Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini menjelaskan status volume yang ditentukan.

```
Get-EC2VolumeStatus -VolumeId vol-12345678
```

Output:

```
Actions          : {}
AvailabilityZone  : us-west-2a
Events           : {}
VolumeId         : vol-12345678
VolumeStatus     : Amazon.EC2.Model.VolumeStatusInfo
```

```
(Get-EC2VolumeStatus -VolumeId vol-12345678).VolumeStatus
```

Output:

Details	Status
-----	-----
{io-enabled, io-performance}	ok

```
(Get-EC2VolumeStatus -VolumeId vol-12345678).VolumeStatus.Details
```

Output:

Name	Status
----	-----
io-enabled	passed
io-performance	not-applicable

- Untuk detail API, lihat [DescribeVolumeStatus](#) di Referensi AWS Tools for PowerShell Cmdlet.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **DescribeVolumes** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `DescribeVolumes`.

CLI

AWS CLI

Contoh 1: Untuk mendeskripsikan volume

`describe-volumes` Contoh berikut menjelaskan volume yang ditentukan di Wilayah saat ini.

```
aws ec2 describe-volumes \
  --volume-ids vol-049df61146c4d7901 vol-1234567890abcdef0
```

Output:

```
{
  "Volumes": [
    {
      "AvailabilityZone": "us-east-1a",
      "Attachments": [
        {
          "AttachTime": "2013-12-18T22:35:00.000Z",
          "InstanceId": "i-1234567890abcdef0",
          "VolumeId": "vol-049df61146c4d7901",
          "State": "attached",
          "DeleteOnTermination": true,
          "Device": "/dev/sda1"
        }
      ],
      "Encrypted": true,
      "KmsKeyId": "arn:aws:kms:us-east-2a:123456789012:key/8c5b2c63-
b9bc-45a3-a87a-5513eEXAMPLE",
      "VolumeType": "gp2",
      "VolumeId": "vol-049df61146c4d7901",
      "State": "in-use",
      "Iops": 100,
      "SnapshotId": "snap-1234567890abcdef0",
      "CreateTime": "2019-12-18T22:35:00.084Z",
      "Size": 8
    },
    {
      "AvailabilityZone": "us-east-1a",
      "Attachments": [],
      "Encrypted": false,
      "VolumeType": "gp2",
```



```

        "VolumeId": "vol-1234567890abcdef0",
        "State": "available",
        "Iops": 300,
        "SnapshotId": "",
        "CreateTime": "2020-02-27T00:02:41.791Z",
        "Size": 100
    }
]
}

```

Contoh 2: Untuk menggambarkan volume yang dilampirkan ke instance tertentu

`describe-volumes` Contoh berikut menjelaskan semua volume yang keduanya dilampirkan ke instance tertentu dan diatur untuk menghapus ketika instance berakhir.

```

aws ec2 describe-volumes \
  --region us-east-1 \
  --filters Name=attachment.instance-id,Values=i-1234567890abcdef0
  Name=attachment.delete-on-termination,Values=true

```

Untuk contoh output `describe-volumes`, lihat Contoh 1.

Contoh 3: Untuk menjelaskan volume yang tersedia di Availability Zone tertentu

`describe-volumes` Contoh berikut menjelaskan semua volume yang memiliki status `available` dan berada di Availability Zone yang ditentukan.

```

aws ec2 describe-volumes \
  --filters Name=status,Values=available Name=availability-zone,Values=us-
east-1a

```

Untuk contoh output `describe-volumes`, lihat Contoh 1.

Contoh 4: Untuk menggambarkan volume berdasarkan tag

`describe-volumes` Contoh berikut menjelaskan semua volume yang memiliki kunci tag `Name` dan nilai yang dimulai dengan `Test`. Output kemudian disaring dengan kueri yang hanya menampilkan tag dan ID volume.

```

aws ec2 describe-volumes \
  --filters Name=tag:Name,Values=Test* \

```

```
--query "Volumes[*].{ID:VolumeId,Tag:Tags}"
```

Output:

```
[
  {
    "Tag": [
      {
        "Value": "Test2",
        "Key": "Name"
      }
    ],
    "ID": "vol-1234567890abcdef0"
  },
  {
    "Tag": [
      {
        "Value": "Test1",
        "Key": "Name"
      }
    ],
    "ID": "vol-049df61146c4d7901"
  }
]
```

Untuk contoh tambahan menggunakan filter tanda, lihat [Bekerja dengan tanda](#) di Panduan Pengguna Amazon EC2.

- Untuk detail API, lihat [DescribeVolumes](#) di Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini menjelaskan volume EBS yang ditentukan.

```
Get-EC2Volume -VolumeId vol-12345678
```

Output:

```
Attachments      : {}
AvailabilityZone  : us-west-2c
```

```

CreateTime      : 7/17/2015 4:35:19 PM
Encrypted       : False
Iops            : 90
KmsKeyId       :
Size           : 30
SnapshotId     : snap-12345678
State          : in-use
Tags           : {}
VolumeId       : vol-12345678
VolumeType     : standard

```

Contoh 2: Contoh ini menjelaskan volume EBS Anda yang memiliki status 'tersedia'.

```
Get-EC2Volume -Filter @{ Name="status"; Values="available" }
```

Output:

```

Attachments    : {}
AvailabilityZone : us-west-2c
CreateTime     : 12/21/2015 2:31:29 PM
Encrypted      : False
Iops          : 60
KmsKeyId      :
Size         : 20
SnapshotId   : snap-12345678
State       : available
Tags       : {}
VolumeId   : vol-12345678
VolumeType : gp2
...

```

Contoh 3: Contoh ini menjelaskan semua volume EBS Anda.

```
Get-EC2Volume
```

- Untuk detail API, lihat [DescribeVolumes](#) di Referensi AWS Tools for PowerShell Cmdlet.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan `DescribeVpcAttribute` dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `DescribeVpcAttribute`.

CLI

AWS CLI

Untuk menggambarkan `enableDnsSupport` atribut

Contoh ini menjelaskan `enableDnsSupport` atribut. Atribut ini menunjukkan apakah resolusi DNS diaktifkan untuk VPC. Jika atribut ini `true`, server DNS Amazon menyelesaikan nama host DNS untuk instance Anda ke alamat IP yang sesuai; jika tidak, tidak.

Perintah:

```
aws ec2 describe-vpc-attribute --vpc-id vpc-a01106c2 --attribute enableDnsSupport
```

Output:

```
{
  "VpcId": "vpc-a01106c2",
  "EnableDnsSupport": {
    "Value": true
  }
}
```

Untuk menggambarkan `enableDnsHostnames` atribut

Contoh ini menjelaskan `enableDnsHostnames` atribut. Atribut ini menunjukkan apakah instance yang diluncurkan di VPC mendapatkan nama host DNS. Jika atribut ini `true`, instance di VPC mendapatkan nama host DNS; jika tidak, mereka tidak.

Perintah:

```
aws ec2 describe-vpc-attribute --vpc-id vpc-a01106c2 --attribute
enableDnsHostnames
```

Output:

```
{
  "VpcId": "vpc-a01106c2",
  "EnableDnsHostnames": {
    "Value": true
  }
}
```

- Untuk detail API, lihat [DescribeVpcAttribute](#) di Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini menjelaskan atribut `enableDnsSupport` ".

```
Get-EC2VpcAttribute -VpcId vpc-12345678 -Attribute enableDnsSupport
```

Output:

```
EnableDnsSupport
-----
True
```

Contoh 2: Contoh ini menjelaskan atribut `enableDnsHostnames` ".

```
Get-EC2VpcAttribute -VpcId vpc-12345678 -Attribute enableDnsHostnames
```

Output:

```
EnableDnsHostnames
-----
True
```

- Untuk detail API, lihat [DescribeVpcAttribute](#) di Referensi AWS Tools for PowerShell Cmdlet.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **DescribeVpcClassicLink** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `DescribeVpcClassicLink`.

CLI

AWS CLI

Untuk menggambarkan ClassicLink status VPC Anda

Contoh ini mencantumkan ClassicLink status `vpc-88888888`.

Perintah:

```
aws ec2 describe-vpc-classic-link --vpc-id vpc-88888888
```

Output:

```
{
  "Vpcs": [
    {
      "ClassicLinkEnabled": true,
      "VpcId": "vpc-88888888",
      "Tags": [
        {
          "Value": "classiclinkvpc",
          "Key": "Name"
        }
      ]
    }
  ]
}
```

Contoh ini hanya mencantumkan VPC yang diaktifkan untuk Classiclink (nilai filter `is-classic-link-enabled` disetel ke) `true`

Perintah:

```
aws ec2 describe-vpc-classic-link --filter "Name=is-classic-link-enabled,Values=true"
```

- Untuk detail API, lihat [DescribeVpcClassicLink](#) di Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh di atas mengembalikan semua VPC dengan ClassicLinkEnabled statusnya untuk wilayah tersebut

```
Get-EC2VpcClassicLink -Region eu-west-1
```

Output:

```
ClassicLinkEnabled Tags    VpcId
-----
False              {Name} vpc-0fc1ff23f45b678eb
False              {}      vpc-01e23c4a5d6db78e9
False              {Name} vpc-0123456b078b9d01f
False              {}      vpc-12cf3b4f
False              {Name} vpc-0b12d3456a7e8901d
```

- Untuk detail API, lihat [DescribeVpcClassicLink](#) di Referensi AWS Tools for PowerShell Cmdlet.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **DescribeVpcClassicLinkDnsSupport** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `DescribeVpcClassicLinkDnsSupport`.

CLI

AWS CLI

Untuk menggambarkan dukungan ClassicLink DNS untuk VPC Anda

Contoh ini menjelaskan status dukungan ClassicLink DNS dari semua VPC Anda.

Perintah:

```
aws ec2 describe-vpc-classic-link-dns-support
```

Output:

```
{
  "Vpcs": [
    {
      "VpcId": "vpc-88888888",
      "ClassicLinkDnsSupported": true
    },
    {
      "VpcId": "vpc-1a2b3c4d",
      "ClassicLinkDnsSupported": false
    }
  ]
}
```

- Untuk detail API, lihat [DescribeVpcClassicLinkDnsSupport](#) di Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini menjelaskan status dukungan ClassicLink DNS VPC untuk wilayah eu-west-1

```
Get-EC2VpcClassicLinkDnsSupport -VpcId vpc-0b12d3456a7e8910d -Region eu-west-1
```

Output:

ClassicLinkDnsSupported	VpcId
False	vpc-0b12d3456a7e8910d
False	vpc-12cf3b4f

- Untuk detail API, lihat [DescribeVpcClassicLinkDnsSupport](#) di Referensi AWS Tools for PowerShell Cmdlet.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **DescribeVpcEndpointServices** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `DescribeVpcEndpointServices`.

CLI

AWS CLI

Contoh 1: Untuk mendeskripsikan semua layanan titik akhir VPC

Contoh "describe-vpc-endpoint-services" berikut mencantumkan semua layanan titik akhir VPC untuk suatu Wilayah. AWS

```
aws ec2 describe-vpc-endpoint-services
```

Output:

```
{
  "ServiceDetails": [
    {
      "ServiceType": [
        {
          "ServiceType": "Gateway"
        }
      ],
      "AcceptanceRequired": false,
      "ServiceName": "com.amazonaws.us-east-1.dynamodb",
      "VpcEndpointPolicySupported": true,
      "Owner": "amazon",
      "AvailabilityZones": [
        "us-east-1a",
        "us-east-1b",
        "us-east-1c",
        "us-east-1d",
        "us-east-1e",
        "us-east-1f"
      ],
      "BaseEndpointDnsNames": [
        "dynamodb.us-east-1.amazonaws.com"
      ]
    }
  ]
}
```

```
]
},
{
  "ServiceType": [
    {
      "ServiceType": "Interface"
    }
  ],
  "PrivateDnsName": "ec2.us-east-1.amazonaws.com",
  "ServiceName": "com.amazonaws.us-east-1.ec2",
  "VpcEndpointPolicySupported": false,
  "Owner": "amazon",
  "AvailabilityZones": [
    "us-east-1a",
    "us-east-1b",
    "us-east-1c",
    "us-east-1d",
    "us-east-1e",
    "us-east-1f"
  ],
  "AcceptanceRequired": false,
  "BaseEndpointDnsNames": [
    "ec2.us-east-1.vpce.amazonaws.com"
  ]
},
{
  "ServiceType": [
    {
      "ServiceType": "Interface"
    }
  ],
  "PrivateDnsName": "ssm.us-east-1.amazonaws.com",
  "ServiceName": "com.amazonaws.us-east-1.ssm",
  "VpcEndpointPolicySupported": true,
  "Owner": "amazon",
  "AvailabilityZones": [
    "us-east-1a",
    "us-east-1b",
    "us-east-1c",
    "us-east-1d",
    "us-east-1e"
  ],
  "AcceptanceRequired": false,
  "BaseEndpointDnsNames": [
```

```

        "ssm.us-east-1.vpce.amazonaws.com"
    ]
}
],
"ServiceNames": [
    "com.amazonaws.us-east-1.dynamodb",
    "com.amazonaws.us-east-1.ec2",
    "com.amazonaws.us-east-1.ec2messages",
    "com.amazonaws.us-east-1.elasticloadbalancing",
    "com.amazonaws.us-east-1.kinesis-streams",
    "com.amazonaws.us-east-1.s3",
    "com.amazonaws.us-east-1.ssm"
]
}

```

Untuk informasi selengkapnya, [lihat Melihat nama AWS layanan yang tersedia](#) di Panduan Pengguna untuk AWS PrivateLink.

Contoh 2: Untuk menjelaskan detail tentang layanan endpoint

Contoh "describe-vpc-endpoint-services" berikut mencantumkan rincian service endpoint antarmuka Amazon S3

```

aws ec2 describe-vpc-endpoint-services \
  --filter "Name=service-type,Values=Interface" Name=service-
name,Values=com.amazonaws.us-east-1.s3

```

Output:

```

{
  "ServiceDetails": [
    {
      "ServiceName": "com.amazonaws.us-east-1.s3",
      "ServiceId": "vpce-svc-081d84efcdEXAMPLE",
      "ServiceType": [
        {
          "ServiceType": "Interface"
        }
      ],
      "AvailabilityZones": [
        "us-east-1a",
        "us-east-1b",

```

```

        "us-east-1c",
        "us-east-1d",
        "us-east-1e",
        "us-east-1f"
    ],
    "Owner": "amazon",
    "BaseEndpointDnsNames": [
        "s3.us-east-1.vpce.amazonaws.com"
    ],
    "VpcEndpointPolicySupported": true,
    "AcceptanceRequired": false,
    "ManagesVpcEndpoints": false,
    "Tags": []
  }
],
"ServiceNames": [
  "com.amazonaws.us-east-1.s3"
]
}

```

Untuk informasi selengkapnya, [lihat Melihat nama AWS layanan yang tersedia](#) di Panduan Pengguna untuk AWS PrivateLink.

- Untuk detail API, lihat [DescribeVpcEndpointServices](#) di Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini menjelaskan layanan titik akhir VPC EC2 dengan filter yang diberikan, dalam hal ini com.amazonaws.eu-west-1.ecs. Selanjutnya, itu juga memperluas ServiceDetails properti dan menampilkan detailnya

```

Get-EC2VpcEndpointService -Region eu-west-1 -MaxResult 5 -Filter @{"Name"="service-name";Values="com.amazonaws.eu-west-1.ecs"} | Select-Object -ExpandProperty ServiceDetails

```

Output:

```

AcceptanceRequired      : False
AvailabilityZones       : {eu-west-1a, eu-west-1b, eu-west-1c}
BaseEndpointDnsNames   : {ecs.eu-west-1.vpce.amazonaws.com}

```

```
Owner           : amazon
PrivateDnsName  : ecs.eu-west-1.amazonaws.com
ServiceName     : com.amazonaws.eu-west-1.ecs
ServiceType     : {Amazon.EC2.Model.ServiceTypeDetail}
VpcEndpointPolicySupported : False
```

Contoh 2: Contoh ini mengambil semua layanan Titik Akhir VPC EC2 dan mengembalikan “ssm” yang cocok ServiceNames

```
Get-EC2VpcEndpointService -Region eu-west-1 | Select-Object -ExpandProperty
  Servicenames | Where-Object { -match "ssm"}
```

Output:

```
com.amazonaws.eu-west-1.ssm
com.amazonaws.eu-west-1.ssmmessages
```

- Untuk detail API, lihat [DescribeVpcEndpointServices](#) di Referensi AWS Tools for PowerShell Cmdlet.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **DescribeVpcEndpoints** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `DescribeVpcEndpoints`.

CLI

AWS CLI

Untuk mendeskripsikan titik akhir VPC Anda

`describe-vpc-endpoints` Contoh berikut menampilkan detail untuk semua titik akhir VPC Anda.

```
aws ec2 describe-vpc-endpoints
```

Output:

```

{
  "VpcEndpoints": [
    {
      "PolicyDocument": "{\"Version\":\"2008-10-17\",\"Statement\":
[[{\"Effect\":\"Allow\",\"Principal\":\"*\",\"Action\":\"*\",\"Resource\":\"*
\"}]]\",
      "VpcId": "vpc-aabb1122",
      "NetworkInterfaceIds": [],
      "SubnetIds": [],
      "PrivateDnsEnabled": true,
      "State": "available",
      "ServiceName": "com.amazonaws.us-east-1.dynamodb",
      "RouteTableIds": [
        "rtb-3d560345"
      ],
      "Groups": [],
      "VpcEndpointId": "vpce-032a826a",
      "VpcEndpointType": "Gateway",
      "CreationTimestamp": "2017-09-05T20:41:28Z",
      "DnsEntries": [],
      "OwnerId": "123456789012"
    },
    {
      "PolicyDocument": "{\n  \"Statement\": [\n    {\n      \"Action\":
\\*\\\", \n      \"Effect\": \"Allow\", \n      \"Principal\": \\*\\\", \n
\\\"Resource\": \\*\\\" \n    } \n  ] \n}",
      "VpcId": "vpc-1a2b3c4d",
      "NetworkInterfaceIds": [
        "eni-2ec2b084",
        "eni-1b4a65cf"
      ],
      "SubnetIds": [
        "subnet-d6fcaa8d",
        "subnet-7b16de0c"
      ],
      "PrivateDnsEnabled": false,
      "State": "available",
      "ServiceName": "com.amazonaws.us-east-1.elasticloadbalancing",
      "RouteTableIds": [],
      "Groups": [
        {
          "GroupName": "default",

```

```
        "GroupId": "sg-54e8bf31"
      }
    ],
    "VpcEndpointId": "vpce-0f89a33420c1931d7",
    "VpcEndpointType": "Interface",
    "CreationTimestamp": "2017-09-05T17:55:27.583Z",
    "DnsEntries": [
      {
        "HostedZoneId": "Z7HUB22UULQXV",
        "DnsName": "vpce-0f89a33420c1931d7-
bluzidnv.elasticloadbalancing.us-east-1.vpce.amazonaws.com"
      },
      {
        "HostedZoneId": "Z7HUB22UULQXV",
        "DnsName": "vpce-0f89a33420c1931d7-bluzidnv-us-
east-1b.elasticloadbalancing.us-east-1.vpce.amazonaws.com"
      },
      {
        "HostedZoneId": "Z7HUB22UULQXV",
        "DnsName": "vpce-0f89a33420c1931d7-bluzidnv-us-
east-1a.elasticloadbalancing.us-east-1.vpce.amazonaws.com"
      }
    ],
    "OwnerId": "123456789012"
  },
  {
    "VpcEndpointId": "vpce-aabbaabbaabbaabba",
    "VpcEndpointType": "GatewayLoadBalancer",
    "VpcId": "vpc-111122223333aabbcc",
    "ServiceName": "com.amazonaws.vpce.us-east-1.vpce-
svc-123123a1c43abc123",
    "State": "available",
    "SubnetIds": [
      "subnet-0011aabbcc2233445"
    ],
    "RequesterManaged": false,
    "NetworkInterfaceIds": [
      "eni-01010120203030405"
    ],
    "CreationTimestamp": "2020-11-11T08:06:03.522Z",
    "Tags": [],
    "OwnerId": "123456789012"
  }
]
```

```
}

```

Untuk informasi selengkapnya, lihat [Titik akhir VPC](#) di Panduan Pengguna Amazon VPC.

- Untuk detail API, lihat [DescribeVpcEndpoints](#) di Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini menjelaskan satu atau beberapa titik akhir VPC Anda untuk wilayah eu-west-1. Kemudian pipa output ke perintah berikutnya, yang memilih VpcEndpointId properti dan mengembalikan array VPC ID sebagai string array

```
Get-EC2VpcEndpoint -Region eu-west-1 | Select-Object -ExpandProperty
  VpcEndpointId

```

Output:

```
vpce-01a2ab3f4f5cc6f7d
vpce-01d2b345a6787890b
vpce-0012e34d567890e12
vpce-0c123db4567890123

```

Contoh 2: Contoh ini menjelaskan semua titik akhir vpc untuk wilayah eu-west-1 dan memilih,, dan properti untuk menyajikannya dalam format tabel VpcEndpointId VpcId ServiceName PrivateDnsEnabled

```
Get-EC2VpcEndpoint -Region eu-west-1 | Select-Object VpcEndpointId, VpcId,
  ServiceName, PrivateDnsEnabled | Format-Table -AutoSize

```

Output:

```
VpcEndpointId      VpcId              ServiceName
-----
PrivateDnsEnabled
-----
vpce-02a2ab2f2f2cc2f2d vpc-0fc6ff46f65b039eb com.amazonaws.eu-west-1.ssm
                True
vpce-01d1b111a1114561b vpc-0fc6ff46f65b039eb com.amazonaws.eu-west-1.ec2
                True

```



```
vpce-0011e23d45167e838 vpc-0fc6ff46f65b039eb com.amazonaws.eu-west-1.ec2messages
    True
vpce-0c123db4567890123 vpc-0fc6ff46f65b039eb com.amazonaws.eu-west-1.ssmmessages
    True
```

Contoh 3: Contoh ini mengeksport dokumen kebijakan untuk VPC Endpoint vpce-01a2ab3f4f5cc6f7d ke dalam file json

```
Get-EC2VpcEndpoint -Region eu-west-1 -VpcEndpointId vpce-01a2ab3f4f5cc6f7d |
    Select-Object -expand PolicyDocument | Out-File vpce_policyDocument.json
```

- Untuk detail API, lihat [DescribeVpcEndpoints](#) di Referensi AWS Tools for PowerShell Cmdlet.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **DescribeVpcs** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `DescribeVpcs`.

Contoh tindakan adalah kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Anda dapat melihat tindakan ini dalam konteks dalam contoh kode berikut:

- [Membangun dan mengelola layanan yang tangguh](#)

.NET

AWS SDK for .NET

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
/// <summary>
/// Get the default VPC for the account.
```

```
/// </summary>
/// <returns>The default VPC object.</returns>
public async Task<Vpc> GetDefaultVpc()
{
    var vpcResponse = await _amazonEc2.DescribeVpcsAsync(
        new DescribeVpcsRequest()
        {
            Filters = new List<Amazon.EC2.Model.Filter>()
            {
                new ("is-default", new List<string>() { "true" })
            }
        });
    return vpcResponse.Vpcs[0];
}
```

- Untuk detail API, lihat [DescribeVpcs](#) di Referensi AWS SDK for .NET API.

CLI

AWS CLI

Contoh 1: Untuk menjelaskan semua VPC Anda

Contoh `describe-vpcs` berikut mengambil detail tentang VPC Anda.

```
aws ec2 describe-vpcs
```

Output:

```
{
  "Vpcs": [
    {
      "CidrBlock": "30.1.0.0/16",
      "DhcpOptionsId": "dopt-19edf471",
      "State": "available",
      "VpcId": "vpc-0e9801d129EXAMPLE",
      "OwnerId": "111122223333",
      "InstanceTenancy": "default",
      "CidrBlockAssociationSet": [
        {
          "AssociationId": "vpc-cidr-assoc-062c64cfafEXAMPLE",
          "CidrBlock": "30.1.0.0/16",
```

```

        "CidrBlockState": {
            "State": "associated"
        }
    ],
    "IsDefault": false,
    "Tags": [
        {
            "Key": "Name",
            "Value": "Not Shared"
        }
    ]
},
{
    "CidrBlock": "10.0.0.0/16",
    "DhcpOptionsId": "dopt-19edf471",
    "State": "available",
    "VpcId": "vpc-06e4ab6c6cEXAMPLE",
    "OwnerId": "222222222222",
    "InstanceTenancy": "default",
    "CidrBlockAssociationSet": [
        {
            "AssociationId": "vpc-cidr-assoc-00b17b4eddEXAMPLE",
            "CidrBlock": "10.0.0.0/16",
            "CidrBlockState": {
                "State": "associated"
            }
        }
    ],
    "IsDefault": false,
    "Tags": [
        {
            "Key": "Name",
            "Value": "Shared VPC"
        }
    ]
}
]
}

```

Contoh 2: Untuk menjelaskan VPC tertentu

Contoh `describe-vpcs` berikut mengambil detail untuk VPC tertentu.

```
aws ec2 describe-vpcs \  
  --vpc-ids vpc-06e4ab6c6cEXAMPLE
```

Output:

```
{  
  "Vpcs": [  
    {  
      "CidrBlock": "10.0.0.0/16",  
      "DhcpOptionsId": "dopt-19edf471",  
      "State": "available",  
      "VpcId": "vpc-06e4ab6c6cEXAMPLE",  
      "OwnerId": "111122223333",  
      "InstanceTenancy": "default",  
      "CidrBlockAssociationSet": [  
        {  
          "AssociationId": "vpc-cidr-assoc-00b17b4eddEXAMPLE",  
          "CidrBlock": "10.0.0.0/16",  
          "CidrBlockState": {  
            "State": "associated"  
          }  
        }  
      ],  
      "IsDefault": false,  
      "Tags": [  
        {  
          "Key": "Name",  
          "Value": "Shared VPC"  
        }  
      ]  
    }  
  ]  
}
```

- Untuk detail API, lihat [DescribeVpcs](#) di Referensi AWS CLI Perintah.

JavaScript

SDK untuk JavaScript (v3)

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
const client = new EC2Client({});
const { Vpcs } = await client.send(
  new DescribeVpcsCommand({
    Filters: [{ Name: "is-default", Values: ["true"] }],
  }),
);
```

- Untuk detail API, lihat [DescribeVpcs](#) di Referensi AWS SDK for JavaScript API.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini menjelaskan VPC yang ditentukan.

```
Get-EC2Vpc -VpcId vpc-12345678
```

Output:

```
CidrBlock      : 10.0.0.0/16
DhcpOptionsId  : dopt-1a2b3c4d
InstanceTenancy : default
IsDefault      : False
State          : available
Tags           : {Name}
VpcId          : vpc-12345678
```

Contoh 2: Contoh ini menjelaskan VPC default (hanya ada satu per wilayah). Jika akun Anda mendukung EC2-Classic di wilayah ini, tidak ada VPC default.

```
Get-EC2Vpc -Filter @{"Name"="isDefault"; Values="true"}
```

Output:

```
CidrBlock      : 172.31.0.0/16
DhcpOptionsId  : dopt-12345678
InstanceTenancy : default
IsDefault      : True
State          : available
Tags           : {}
VpcId          : vpc-45678901
```

Contoh 3: Contoh ini menjelaskan VPC yang cocok dengan filter yang ditentukan (yaitu, memiliki CIDR yang cocok dengan nilai '10.0.0.0/16' dan berada dalam keadaan 'tersedia').

```
Get-EC2Vpc -Filter @{"Name"="cidr";
  Values="10.0.0.0/16"},@{"Name"="state";Values="available"}
```

Contoh 4: Contoh ini menjelaskan semua VPC Anda.

```
Get-EC2Vpc
```

- Untuk detail API, lihat [DescribeVpcs](#) di Referensi AWS Tools for PowerShell Cmdlet.

Python

SDK untuk Python (Boto3)

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
class AutoScaler:
    """
    Encapsulates Amazon EC2 Auto Scaling and EC2 management actions.
    """
```

```
def __init__(
    self,
    resource_prefix,
    inst_type,
    ami_param,
    autoscaling_client,
    ec2_client,
    ssm_client,
    iam_client,
):
    """
    :param resource_prefix: The prefix for naming AWS resources that are
    created by this class.
    :param inst_type: The type of EC2 instance to create, such as t3.micro.
    :param ami_param: The Systems Manager parameter used to look up the AMI
    that is
        created.
    :param autoscaling_client: A Boto3 EC2 Auto Scaling client.
    :param ec2_client: A Boto3 EC2 client.
    :param ssm_client: A Boto3 Systems Manager client.
    :param iam_client: A Boto3 IAM client.
    """
    self.inst_type = inst_type
    self.ami_param = ami_param
    self.autoscaling_client = autoscaling_client
    self.ec2_client = ec2_client
    self.ssm_client = ssm_client
    self.iam_client = iam_client
    self.launch_template_name = f"{resource_prefix}-template"
    self.group_name = f"{resource_prefix}-group"
    self.instance_policy_name = f"{resource_prefix}-pol"
    self.instance_role_name = f"{resource_prefix}-role"
    self.instance_profile_name = f"{resource_prefix}-prof"
    self.bad_creds_policy_name = f"{resource_prefix}-bc-pol"
    self.bad_creds_role_name = f"{resource_prefix}-bc-role"
    self.bad_creds_profile_name = f"{resource_prefix}-bc-prof"
    self.key_pair_name = f"{resource_prefix}-key-pair"

def get_default_vpc(self):
    """
    Gets the default VPC for the account.

    :return: Data about the default VPC.
```

```

"""
try:
    response = self.ec2_client.describe_vpcs(
        Filters=[{"Name": "is-default", "Values": ["true"]}])
except ClientError as err:
    raise AutoScalerError(f"Couldn't get default VPC: {err}")
else:
    return response["Vpcs"][0]

```

- Untuk detail API, lihat [DescribeVpcs](#) di AWS SDK for Python (Boto3) Referensi API.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **DescribeVpnConnections** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `DescribeVpnConnections`.

CLI

AWS CLI

Contoh 1: Untuk menggambarkan koneksi VPN Anda

`describe-vpn-connections` Contoh berikut menjelaskan semua koneksi VPN Site-to-Site Anda.

```
aws ec2 describe-vpn-connections
```

Output:

```
{
  "VpnConnections": [
    {
      "CustomerGatewayConfiguration": "...configuration information...",
      "CustomerGatewayId": "cgw-01234567abcde1234",

```



```

    "Category": "VPN",
    "State": "available",
    "Type": "ipsec.1",
    "VpnConnectionId": "vpn-1122334455aabbccd",
    "TransitGatewayId": "tgw-00112233445566aab",
    "Options": {
      "EnableAcceleration": false,
      "StaticRoutesOnly": true,
      "LocalIpv4NetworkCidr": "0.0.0.0/0",
      "RemoteIpv4NetworkCidr": "0.0.0.0/0",
      "TunnelInsideIpVersion": "ipv4"
    },
    "Routes": [],
    "Tags": [
      {
        "Key": "Name",
        "Value": "CanadaVPN"
      }
    ],
    "VgwTelemetry": [
      {
        "AcceptedRouteCount": 0,
        "LastStatusChange": "2020-07-29T10:35:11.000Z",
        "OutsideIpAddress": "203.0.113.3",
        "Status": "DOWN",
        "StatusMessage": ""
      },
      {
        "AcceptedRouteCount": 0,
        "LastStatusChange": "2020-09-02T09:09:33.000Z",
        "OutsideIpAddress": "203.0.113.5",
        "Status": "UP",
        "StatusMessage": ""
      }
    ]
  }
]
}

```

Untuk informasi selengkapnya, lihat [Cara kerja AWS VPN Site-to-Site di Panduan Pengguna VPN AWS Site-to-Site](#).

Contoh 2: Untuk menggambarkan koneksi VPN Anda yang tersedia

`describe-vpn-connections` Contoh berikut menjelaskan koneksi VPN Site-to-Site Anda dengan status. `available`

```
aws ec2 describe-vpn-connections \  
  --filters "Name=state,Values=available"
```

Untuk informasi selengkapnya, lihat [Cara kerja AWS VPN Site-to-Site di Panduan Pengguna VPN AWS Site-to-Site](#).

- Untuk detail API, lihat [DescribeVpnConnections](#) di Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini menjelaskan koneksi VPN yang ditentukan.

```
Get-EC2VpnConnection -VpnConnectionId vpn-12345678
```

Output:

```
CustomerGatewayConfiguration : [XML document]  
CustomerGatewayId           : cgw-1a2b3c4d  
Options                      : Amazon.EC2.Model.VpnConnectionOptions  
Routes                      : {Amazon.EC2.Model.VpnStaticRoute}  
State                        : available  
Tags                        : {}  
Type                        : ipsec.1  
VgwTelemetry                : {Amazon.EC2.Model.VgwTelemetry,  
  Amazon.EC2.Model.VgwTelemetry}  
VpnConnectionId            : vpn-12345678  
VpnGatewayId               : vgw-1a2b3c4d
```

Contoh 2: Contoh ini menjelaskan koneksi VPN apa pun yang statusnya tertunda atau tersedia.

```
$filter = New-Object Amazon.EC2.Model.Filter  
$filter.Name = "state"  
$filter.Values = @( "pending", "available" )  
  
Get-EC2VpnConnection -Filter $filter
```

Contoh 3: Contoh ini menjelaskan semua koneksi VPN Anda.

```
Get-EC2VpnConnection
```

- Untuk detail API, lihat [DescribeVpnConnections](#) di Referensi AWS Tools for PowerShell Cmdlet.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **DescribeVpnGateways** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `DescribeVpnGateways`.

CLI

AWS CLI

Untuk menggambarkan gateway pribadi virtual Anda

Contoh ini menjelaskan gateway pribadi virtual Anda.

Perintah:

```
aws ec2 describe-vpn-gateways
```

Output:

```
{
  "VpnGateways": [
    {
      "State": "available",
      "Type": "ipsec.1",
      "VpnGatewayId": "vgw-f211f09b",
      "VpcAttachments": [
        {
          "State": "attached",
          "VpcId": "vpc-98eb5ef5"
        }
      ]
    }
  ]
}
```

```
    },
    {
      "State": "available",
      "Type": "ipsec.1",
      "VpnGatewayId": "vgw-9a4cacf3",
      "VpcAttachments": [
        {
          "State": "attaching",
          "VpcId": "vpc-a01106c2"
        }
      ]
    }
  ]
}
```

- Untuk detail API, lihat [DescribeVpnGateways](#) di Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini menjelaskan gateway pribadi virtual yang ditentukan.

```
Get-EC2VpnGateway -VpnGatewayId vgw-1a2b3c4d
```

Output:

```
AvailabilityZone :
State            : available
Tags             : {}
Type             : ipsec.1
VpcAttachments  : {vpc-12345678}
VpnGatewayId    : vgw-1a2b3c4d
```

Contoh 2: Contoh ini menjelaskan gateway pribadi virtual yang statusnya tertunda atau tersedia.

```
$filter = New-Object Amazon.EC2.Model.Filter
$filter.Name = "state"
$filter.Values = @( "pending", "available" )
```

```
Get-EC2VpnGateway -Filter $filter
```

Contoh 3: Contoh ini menjelaskan semua gateway pribadi virtual Anda.

```
Get-EC2VpnGateway
```

- Untuk detail API, lihat [DescribeVpnGateways](#) di Referensi AWS Tools for PowerShell Cmdlet.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **DetachInternetGateway** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `DetachInternetGateway`.

CLI

AWS CLI

Untuk melepaskan gateway internet dari VPC Anda

`detach-internet-gateway` Contoh berikut melepaskan gateway internet yang ditentukan dari VPC tertentu.

```
aws ec2 detach-internet-gateway \  
  --internet-gateway-id igw-0d0fb496b3EXAMPLE \  
  --vpc-id vpc-0a60eb65b4EXAMPLE
```

Perintah ini tidak menghasilkan output.

Untuk informasi lebih lanjut, lihat [Gateway internet](#) di Panduan Pengguna Amazon VPC.

- Untuk detail API, lihat [DetachInternetGateway](#) di Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini melepaskan gateway Internet yang ditentukan dari VPC yang ditentukan.

```
Dismount-EC2InternetGateway -InternetGatewayId igw-1a2b3c4d -VpcId vpc-12345678
```

- Untuk detail API, lihat [DetachInternetGateway](#) di Referensi AWS Tools for PowerShell Cmdlet.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **DetachNetworkInterface** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `DetachNetworkInterface`.

CLI

AWS CLI

Untuk melepaskan antarmuka jaringan dari instans Anda

Contoh ini melepaskan antarmuka jaringan yang ditentukan dari contoh yang ditentukan. Jika perintah berhasil, tidak ada output yang akan ditampilkan.

Perintah:

```
aws ec2 detach-network-interface --attachment-id eni-attach-66c4350a
```

- Untuk detail API, lihat [DetachNetworkInterface](#) di Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini menghapus lampiran yang ditentukan antara antarmuka jaringan dan instance.

```
Dismount-EC2NetworkInterface -AttachmentId eni-attach-1a2b3c4d -Force
```

- Untuk detail API, lihat [DetachNetworkInterface](#) di Referensi AWS Tools for PowerShell Cmdlet.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **DetachVolume** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `DetachVolume`.

CLI

AWS CLI

Untuk melepaskan volume dari sebuah instance

Perintah contoh ini melepaskan volume (`vol-049df61146c4d7901`) dari instance yang dilampirkan.

Perintah:

```
aws ec2 detach-volume --volume-id vol-1234567890abcdef0
```

Output:

```
{
  "AttachTime": "2014-02-27T19:23:06.000Z",
  "InstanceId": "i-1234567890abcdef0",
  "VolumeId": "vol-049df61146c4d7901",
  "State": "detaching",
  "Device": "/dev/sdb"
}
```

- Untuk detail API, lihat [DetachVolume](#) di Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini melepaskan volume yang ditentukan.

```
Dismount-EC2Volume -VolumeId vol-12345678
```

Output:

```
AttachTime      : 12/22/2015 1:53:58 AM
DeleteOnTermination : False
Device          : /dev/sdh
InstanceId      : i-1a2b3c4d
State           : detaching
VolumeId       : vol-12345678
```

Contoh 2: Anda juga dapat menentukan ID instans dan nama perangkat untuk memastikan bahwa Anda melepaskan volume yang benar.

```
Dismount-EC2Volume -VolumeId vol-12345678 -InstanceId i-1a2b3c4d -Device /dev/sdh
```

- Untuk detail API, lihat [DetachVolume](#) di Referensi AWS Tools for PowerShell Cmdlet.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **DetachVpnGateway** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `DetachVpnGateway`.

CLI

AWS CLI

Untuk melepaskan gateway pribadi virtual dari VPC Anda

Contoh ini melepaskan gateway pribadi virtual yang ditentukan dari VPC yang ditentukan. Jika perintah berhasil, tidak ada output yang akan ditampilkan.

Perintah:

```
aws ec2 detach-vpn-gateway --vpn-gateway-id vgw-9a4cacf3 --vpc-id vpc-a01106c2
```

- Untuk detail API, lihat [DetachVpnGateway](#) di Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini melepaskan gateway pribadi virtual yang ditentukan dari VPC yang ditentukan.

```
Dismount-EC2VpnGateway -VpnGatewayId vgw-1a2b3c4d -VpcId vpc-12345678
```

- Untuk detail API, lihat [DetachVpnGateway](#) di Referensi AWS Tools for PowerShell Cmdlet.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **DisableVgwRoutePropagation** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `DisableVgwRoutePropagation`.

CLI

AWS CLI

Untuk menonaktifkan propagasi rute

Contoh ini menonaktifkan gateway pribadi virtual yang ditentukan dari menyebarkan rute statis ke tabel rute yang ditentukan. Jika perintah berhasil, tidak ada output yang akan ditampilkan.

Perintah:

```
aws ec2 disable-vgw-route-propagation --route-table-id rtb-22574640 --gateway-id vgw-9a4cacf3
```

- Untuk detail API, lihat [DisableVgwRoutePropagation](#) di Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini menonaktifkan VGW dari menyebarkan rute secara otomatis ke tabel perutean yang ditentukan.

```
Disable-EC2VgwRoutePropagation -RouteTableId rtb-12345678 -GatewayId vgw-1a2b3c4d
```

- Untuk detail API, lihat [DisableVgwRoutePropagation](#) di Referensi AWS Tools for PowerShell Cmdlet.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **DisableVpcClassicLink** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `DisableVpcClassicLink`.

CLI

AWS CLI

Untuk menonaktifkan `ClassicLink` untuk VPC

Contoh ini menonaktifkan `ClassicLink` untuk `vpc-88888888`.

Perintah:

```
aws ec2 disable-vpc-classic-link --vpc-id vpc-88888888
```

Output:

```
{
  "Return": true
}
```

- Untuk detail API, lihat [DisableVpcClassicLink](#) di Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini menonaktifkan EC2 untuk `VpcClassicLink` `vpc-01e23c4a5d6db78e9`. Ia mengembalikan `True` atau `False`

```
Disable-EC2VpcClassicLink -VpcId vpc-01e23c4a5d6db78e9
```

- Untuk detail API, lihat [DisableVpcClassicLink](#) di Referensi AWS Tools for PowerShell Cmdlet.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **DisableVpcClassicLinkDnsSupport** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `DisableVpcClassicLinkDnsSupport`.

CLI

AWS CLI

Untuk menonaktifkan dukungan ClassicLink DNS untuk VPC

Contoh ini menonaktifkan dukungan ClassicLink DNS untuk `vpc-88888888`

Perintah:

```
aws ec2 disable-vpc-classic-link-dns-support --vpc-id vpc-88888888
```

Output:

```
{
  "Return": true
}
```

- Untuk detail API, lihat [DisableVpcClassicLinkDnsSupport](#) di Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini menonaktifkan dukungan ClassicLink DNS untuk `vpc-0b12d3456a7e8910d`

```
Disable-EC2VpcClassicLinkDnsSupport -VpcId vpc-0b12d3456a7e8910d
```

- Untuk detail API, lihat [DisableVpcClassicLinkDnsSupport](#) di Referensi AWS Tools for PowerShell Cmdlet.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **DisassociateAddress** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `DisassociateAddress`.

Contoh tindakan adalah kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Anda dapat melihat tindakan ini dalam konteks dalam contoh kode berikut:

- [Memulai instans](#)

.NET

AWS SDK for .NET

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
/// <summary>
/// Disassociate an Elastic IP address from an EC2 instance.
/// </summary>
/// <param name="associationId">The association Id.</param>
/// <returns>A Boolean value indicating the success of the action.</returns>
public async Task<bool> DisassociateIp(string associationId)
{
    var response = await _amazonEC2.DisassociateAddressAsync(
        new DisassociateAddressRequest { AssociationId = associationId });
    return response.HttpStatusCode == HttpStatusCode.OK;
}
```

- Untuk detail API, lihat [DisassociateAddress](#) di Referensi AWS SDK for .NET API.

Bash

AWS CLI dengan skrip Bash

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
#####
# function ec2_disassociate_address
#
# This function disassociates an Elastic IP address from an Amazon Elastic
# Compute Cloud (Amazon EC2) instance.
#
# Parameters:
#     -a association_id - The association ID that represents the association of
#     the Elastic IP address with an instance.
#
# And:
#     0 - If successful.
#     1 - If it fails.
#
#####
function ec2_disassociate_address() {
    local association_id response

    # Function to display usage information
    function usage() {
        echo "function ec2_disassociate_address"
        echo "Disassociates an Elastic IP address from an Amazon Elastic Compute
        Cloud (Amazon EC2) instance."
        echo " -a association_id - The association ID that represents the
        association of the Elastic IP address with an instance."
        echo ""
    }
}
```

```

# Parse the command-line arguments
while getopts "a:h" option; do
  case "${option}" in
    a) association_id="${OPTARG}" ;;
    h)
      usage
      return 0
      ;;
    \?)
      echo "Invalid parameter"
      usage
      return 1
      ;;
  esac
done
export OPTIND=1

# Validate the input parameters
if [[ -z "$association_id" ]]; then
  errecho "ERROR: You must provide an association ID with the -a parameter."
  return 1
fi

response=$(aws ec2 disassociate-address \
  --association-id "$association_id") || {
  aws_cli_error_log ${?}
  errecho "ERROR: AWS reports disassociate-address operation failed."
  errecho "$response"
  return 1
}

return 0
}

```

Fungsi utilitas yang digunakan dalam contoh ini.

```

#####
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
#####
function errecho() {

```

```

printf "%s\n" "$*" 1>&2
}

#####
# function aws_cli_error_log()
#
# This function is used to log the error messages from the AWS CLI.
#
# The function expects the following argument:
#     $1 - The error code returned by the AWS CLI.
#
# Returns:
#     0: - Success.
#
#####
function aws_cli_error_log() {
    local err_code=$1
    errecho "Error code : $err_code"
    if [ "$err_code" == 1 ]; then
        errecho " One or more S3 transfers failed."
    elif [ "$err_code" == 2 ]; then
        errecho " Command line failed to parse."
    elif [ "$err_code" == 130 ]; then
        errecho " Process received SIGINT."
    elif [ "$err_code" == 252 ]; then
        errecho " Command syntax invalid."
    elif [ "$err_code" == 253 ]; then
        errecho " The system environment or configuration was invalid."
    elif [ "$err_code" == 254 ]; then
        errecho " The service returned an error."
    elif [ "$err_code" == 255 ]; then
        errecho " 255 is a catch-all error."
    fi

    return 0
}

```

- Untuk detail API, lihat [DisassociateAddress](#) di Referensi AWS CLI Perintah.

CLI

AWS CLI

Untuk melepaskan kaitan alamat IP Elastis di EC2-Classic

Contoh ini melepaskan kaitan alamat IP Elastis dari instans di EC2-Classic. Jika perintah berhasil, tidak ada output yang akan ditampilkan.

Perintah:

```
aws ec2 disassociate-address --public-ip 198.51.100.0
```

Untuk melepaskan kaitan alamat IP Elastis di EC2-VPC

Contoh ini melepaskan kaitan alamat IP Elastis dari instans di VPC. Jika perintah berhasil, tidak ada output yang akan ditampilkan.

Perintah:

```
aws ec2 disassociate-address --association-id eipassoc-2bebb745
```

- Untuk detail API, lihat [DisassociateAddress](#) di Referensi AWS CLI Perintah.

Java

SDK untuk Java 2.x

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
public static void disassociateAddress(Ec2Client ec2, String associationId) {  
    try {  
        DisassociateAddressRequest addressRequest =  
DisassociateAddressRequest.builder()  
            .associationId(associationId)  
            .build();
```



```
        ec2.disassociateAddress(addressRequest);
        System.out.println("You successfully disassociated the address!");

    } catch (Ec2Exception e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
}
```

- Untuk detail API, lihat [DisassociateAddress](#) di Referensi AWS SDK for Java 2.x API.

JavaScript

SDK untuk JavaScript (v3)

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
import { DisassociateAddressCommand } from "@aws-sdk/client-ec2";

import { client } from "../libs/client.js";

// Disassociate an Elastic IP address from an instance.
export const main = async () => {
    const command = new DisassociateAddressCommand({
        // You can also use PublicIp, but that is for EC2 classic which is being
        // retired.
        AssociationId: "ASSOCIATION_ID",
    });

    try {
        await client.send(command);
        console.log("Successfully disassociated address");
    } catch (err) {
        console.error(err);
    }
};
```

- Untuk detail API, lihat [DisassociateAddress](#) di Referensi AWS SDK for JavaScript API.

Kotlin

SDK untuk Kotlin

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
suspend fun disassociateAddressSc(associationIdVal: String?) {
    val addressRequest =
        DisassociateAddressRequest {
            associationId = associationIdVal
        }
    Ec2Client { region = "us-west-2" }.use { ec2 ->
        ec2.disassociateAddress(addressRequest)
        println("You successfully disassociated the address!")
    }
}
```

- Untuk detail API, lihat [DisassociateAddress](#) di AWS SDK untuk referensi API Kotlin.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini memisahkan alamat IP Elastis yang ditentukan dari instance yang ditentukan dalam VPC.

```
Unregister-EC2Address -AssociationId eipassoc-12345678
```

Contoh 2: Contoh ini memisahkan alamat IP Elastis yang ditentukan dari instance yang ditentukan di EC2-Classic.

```
Unregister-EC2Address -PublicIp 203.0.113.17
```

- Untuk detail API, lihat [DisassociateAddress](#) di Referensi AWS Tools for PowerShell Cmdlet.

Python

SDK untuk Python (Boto3)

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
class ElasticIpWrapper:
    """Encapsulates Amazon Elastic Compute Cloud (Amazon EC2) Elastic IP address
    actions."""

    def __init__(self, ec2_resource, elastic_ip=None):
        """
        :param ec2_resource: A Boto3 Amazon EC2 resource. This high-level
        resource
                               is used to create additional high-level objects
                               that wrap low-level Amazon EC2 service actions.
        :param elastic_ip: A Boto3 VpcAddress object. This is a high-level object
        that
                               wraps Elastic IP actions.
        """
        self.ec2_resource = ec2_resource
        self.elastic_ip = elastic_ip

    @classmethod
    def from_resource(cls):
        ec2_resource = boto3.resource("ec2")
        return cls(ec2_resource)

    def disassociate(self):
        """
        Removes an association between an Elastic IP address and an instance.
        When the
```

```
association is removed, the instance is assigned a new public IP address.
"""
if self.elastic_ip is None:
    logger.info("No Elastic IP to disassociate.")
    return

try:
    self.elastic_ip.association.delete()
except ClientError as err:
    logger.error(
        "Couldn't disassociate Elastic IP %s from its instance. Here's
why: %s: %s",
        self.elastic_ip.allocation_id,
        err.response["Error"]["Code"],
        err.response["Error"]["Message"],
    )
    raise
```

- Untuk detail API, lihat [DisassociateAddress](#) di AWS SDK for Python (Boto3) Referensi API.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **DisassociateRouteTable** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `DisassociateRouteTable`.

CLI

AWS CLI

Untuk memisahkan tabel rute

Contoh ini memisahkan tabel rute yang ditentukan dari subnet yang ditentukan. Jika perintah berhasil, tidak ada output yang akan ditampilkan.

Perintah:

```
aws ec2 disassociate-route-table --association-id rtbassoc-781d0d1a
```

- Untuk detail API, lihat [DisassociateRouteTable](#) di Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini menghapus asosiasi tertentu antara tabel rute dan subnet.

```
Unregister-EC2RouteTable -AssociationId rtbassoc-1a2b3c4d
```

- Untuk detail API, lihat [DisassociateRouteTable](#) di Referensi AWS Tools for PowerShell Cmdlet.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **EnableVgwRoutePropagation** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `EnableVgwRoutePropagation`.

CLI

AWS CLI

Untuk mengaktifkan propagasi rute

Contoh ini memungkinkan gateway pribadi virtual yang ditentukan untuk menyebarkan rute statis ke tabel rute yang ditentukan. Jika perintah berhasil, tidak ada output yang akan ditampilkan.

Perintah:

```
aws ec2 enable-vgw-route-propagation --route-table-id rtb-22574640 --gateway-id vgw-9a4cacf3
```

- Untuk detail API, lihat [EnableVgwRoutePropagation](#) di Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini memungkinkan VGW yang ditentukan untuk menyebarkan rute secara otomatis ke tabel perutean yang ditentukan.

```
Enable-EC2VgwRoutePropagation -RouteTableId rtb-12345678 -GatewayId vgw-1a2b3c4d
```

- Untuk detail API, lihat [EnableVgwRoutePropagation](#) di Referensi AWS Tools for PowerShell Cmdlet.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **EnableVolumeIo** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `EnableVolumeIo`.

CLI

AWS CLI

Untuk mengaktifkan I/O untuk volume

Contoh ini memungkinkan I/O pada `volumevol-1234567890abcdef0`.

Perintah:

```
aws ec2 enable-volume-io --volume-id vol-1234567890abcdef0
```

Output:

```
{
  "Return": true
}
```

- Untuk detail API, lihat [EnableVolumeIo](#) di Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini memungkinkan operasi I/O untuk volume yang ditentukan, jika operasi I/O dinonaktifkan.

```
Enable-EC2VolumeIO -VolumeId vol-12345678
```

- Untuk detail API, lihat [EnableVolumeIO](#) di Referensi AWS Tools for PowerShell Cmdlet.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **EnableVpcClassicLink** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `EnableVpcClassicLink`.

CLI

AWS CLI

Untuk mengaktifkan VPC untuk ClassicLink

Contoh ini memungkinkan vpc-88888888 untuk ClassicLink

Perintah:

```
aws ec2 enable-vpc-classic-link --vpc-id vpc-88888888
```

Output:

```
{
  "Return": true
}
```

- Untuk detail API, lihat [EnableVpcClassicLink](#) di Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini memungkinkan VPC `vpc-0123456b789b0d12f` untuk ClassicLink

```
Enable-EC2VpcClassicLink -VpcId vpc-0123456b789b0d12f
```

Output:

```
True
```

- Untuk detail API, lihat [EnableVpcClassicLink](#) di Referensi AWS Tools for PowerShell Cmdlet.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **EnableVpcClassicLinkDnsSupport** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `EnableVpcClassicLinkDnsSupport`.

CLI

AWS CLI

Untuk mengaktifkan dukungan ClassicLink DNS untuk VPC

Contoh ini memungkinkan dukungan ClassicLink DNS untuk `vpc-88888888`.

Perintah:

```
aws ec2 enable-vpc-classic-link-dns-support --vpc-id vpc-88888888
```

Output:

```
{
  "Return": true
}
```


- Untuk detail API, lihat [EnableVpcClassicLinkDnsSupport](#) di Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini memungkinkan vpc-0b12d3456a7e8910d untuk mendukung resolusi nama host DNS untuk ClassicLink

```
Enable-EC2VpcClassicLinkDnsSupport -VpcId vpc-0b12d3456a7e8910d -Region eu-west-1
```

- Untuk detail API, lihat [EnableVpcClassicLinkDnsSupport](#) di Referensi AWS Tools for PowerShell Cmdlet.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **GetConsoleOutput** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `GetConsoleOutput`.

CLI

AWS CLI

Contoh 1: Untuk mendapatkan output konsol

`get-console-output` Contoh berikut mendapatkan output konsol untuk instance Linux tertentu.

```
aws ec2 get-console-output \  
  --instance-id i-1234567890abcdef0
```

Output:

```
{  
  "InstanceId": "i-1234567890abcdef0",  
  "Timestamp": "2013-07-25T21:23:53.000Z",
```

```
"Output": "..."  
}
```

Untuk informasi selengkapnya, lihat [Output konsol instans](#) di Panduan Pengguna Amazon EC2.

Contoh 2: Untuk mendapatkan keluaran konsol terbaru

get-console-output Contoh berikut mendapatkan output konsol terbaru untuk instance Linux yang ditentukan.

```
aws ec2 get-console-output \  
  --instance-id i-1234567890abcdef0 \  
  --latest \  
  --output text
```

Output:

```
i-1234567890abcdef0 [ 0.000000] Command line: root=LABEL=/ console=tty1  
console=ttyS0 selinux=0 nvme_core.io_timeout=4294967295  
[ 0.000000] x86/fpu: Supporting XSAVE feature 0x001: 'x87 floating point  
registers'  
[ 0.000000] x86/fpu: Supporting XSAVE feature 0x002: 'SSE registers'  
[ 0.000000] x86/fpu: Supporting XSAVE feature 0x004: 'AVX registers'  
...  
Cloud-init v. 0.7.6 finished at Wed, 09 May 2018 19:01:13 +0000. Datasource  
DataSourceEc2. Up 21.50 seconds  
Amazon Linux AMI release 2018.03  
Kernel 4.14.26-46.32.amzn1.x
```

Untuk informasi selengkapnya, lihat [Output konsol instans](#) di Panduan Pengguna Amazon EC2.

- Untuk detail API, lihat [GetConsoleOutput](#) di Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini mendapatkan output konsol untuk instance Linux yang ditentukan. Output konsol dikodekan.

```
Get-EC2ConsoleOutput -InstanceId i-0e19abcd47c123456
```

Output:

InstanceId	Output
-----	-----
i-0e194d3c47c123637	WyAgICAwLjAwMDAwMF0gQ29tbW...bGU9dHR5UzAgc2Vs

Contoh 2: Contoh ini menyimpan output konsol yang dikodekan dalam variabel dan kemudian menerjemahkannya.

```
$Output_encoded = (Get-EC2ConsoleOutput -InstanceId i-0e19abcd47c123456).Output
[System.Text.Encoding]::UTF8.GetString([System.Convert]::FromBase64String($Output_encoded))
```

- Untuk detail API, lihat [GetConsoleOutput](#) di Referensi AWS Tools for PowerShell Cmdlet.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **GetHostReservationPurchasePreview** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `GetHostReservationPurchasePreview`.

CLI

AWS CLI

Untuk mendapatkan pratinjau pembelian untuk Reservasi Tuan Rumah Khusus

Contoh ini memberikan pratinjau biaya untuk Reservasi Host Khusus yang ditentukan untuk Host Khusus yang ditentukan di akun Anda.

Perintah:

```
aws ec2 get-host-reservation-purchase-preview --offering-id hro-03f707bf363b6b324
--host-id-set h-013abcd2a00cbd123
```

Output:

```
{
  "TotalHourlyPrice": "1.499",
  "Purchase": [
    {
      "HourlyPrice": "1.499",
      "InstanceFamily": "m4",
      "PaymentOption": "NoUpfront",
      "HostIdSet": [
        "h-013abcd2a00cbd123"
      ],
      "UpfrontPrice": "0.000",
      "Duration": 31536000
    }
  ],
  "TotalUpfrontPrice": "0.000"
}
```

- Untuk detail API, lihat [GetHostReservationPurchasePreview](#) di Referensi AWS CLI Perintah.

PowerShell**Alat untuk PowerShell**

Contoh 1: Contoh ini menampilkan pratinjau pembelian reservasi dengan konfigurasi yang cocok dengan Host Khusus Anda h-01e23f4cd567890f1

```
Get-EC2HostReservationPurchasePreview -OfferingId hro-0c1f23456789d0ab -HostIdSet
h-01e23f4cd567890f1
```

Output:

```
CurrencyCode Purchase TotalHourlyPrice TotalUpfrontPrice
-----
                {}          1.307                0.000
```

- Untuk detail API, lihat [GetHostReservationPurchasePreview](#) di Referensi AWS Tools for PowerShell Cmdlet.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **GetPasswordData** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `GetPasswordData`.

CLI

AWS CLI

Untuk mendapatkan kata sandi terenkripsi

Contoh ini mendapatkan kata sandi terenkripsi.

Perintah:

```
aws ec2 get-password-data --instance-id i-1234567890abcdef0
```

Output:

```
{
  "InstanceId": "i-1234567890abcdef0",
  "Timestamp": "2013-08-07T22:18:38.000Z",
  "PasswordData": "gS1JFq+VpcZXqy+iktXMF6NyxQ4qCrT4+ga0uN0enX1MmgXPTj7XEXAMPLE
UQ+YeFfb+L1U4C4AKv652Ux1iRB3CPTYP7WmU3TUnhsuBd+p6LVk7T2lKUmL60Xbk6WPW1VYYm/TRPB1
e1DQ7PY4an/DgZT4mwcpRFigzhniQgDDe01InvSDcwoUTwNs0Y1S8ouri2W4n5GNlriM3Q0AnNVe1Vz/
53TkDtxbNoU606M1gK9zUWSxqEgwbvV2j8c5rP0WCuaMWSF14ziDu4bd7q+4RSyi8NUsVWnKZ4aEZffu
DPGzKrF5yL1f3etP2L4ZR6CvG7K1hx7VK0QVN32Dajw=="
}
```

Untuk mendapatkan kata sandi yang didekripsi

Contoh ini mendapatkan kata sandi yang didekripsi.

Perintah:

```
aws ec2 get-password-data --instance-id i-1234567890abcdef0 --priv-launch-key C:
\Keys\MyKeyPair.pem
```

Output:

```
{
  "InstanceId": "i-1234567890abcdef0",
  "Timestamp": "2013-08-30T23:18:05.000Z",
  "PasswordData": "&ViJ652e*u"
}
```

- Untuk detail API, lihat [GetPasswordData](#) di Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini mendekripsi kata sandi yang ditetapkan Amazon EC2 ke akun Administrator untuk instance Windows yang ditentukan. Sebagai file PEM ditentukan, pengaturan sakelar `-Decrypt` secara otomatis diasumsikan.

```
Get-EC2PasswordData -InstanceId i-12345678 -PemFile C:\path\my-key-pair.pem
```

Output:

```
mYZ(PA9?C)Q
```

Contoh 2: (PowerShell Hanya Windows) Memeriksa instance untuk menentukan nama keypair yang digunakan untuk meluncurkan instance dan kemudian mencoba menemukan data keypair yang sesuai di penyimpanan konfigurasi Toolkit for Visual AWS Studio. Jika data keypair ditemukan, kata sandi didekripsi.

```
Get-EC2PasswordData -InstanceId i-12345678 -Decrypt
```

Output:

```
mYZ(PA9?C)Q
```

Contoh 3: Mengembalikan data kata sandi terenkripsi untuk contoh.

```
Get-EC2PasswordData -InstanceId i-12345678
```

Output:

```
iVz3BAK/WAXV.....dqt8WeMA==
```

- Untuk detail API, lihat [GetPasswordData](#) di Referensi AWS Tools for PowerShell Cmdlet.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **ImportImage** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `ImportImage`.

CLI

AWS CLI

Untuk mengimpor file gambar VM sebagai AMI

`import-image` Contoh berikut mengimpor OVA tertentu.

```
aws ec2 import-image \  
  --disk-containers Format=ova,UserBucket="{S3Bucket=my-import-bucket,S3Key=vms/  
my-server-vm.ova}"
```

Output:

```
{  
  "ImportTaskId": "import-ami-1234567890abcdef0",  
  "Progress": "2",  
  "SnapshotDetails": [  
    {  
      "DiskImageSize": 0.0,  
      "Format": "ova",  
      "UserBucket": {  
        "S3Bucket": "my-import-bucket",  
        "S3Key": "vms/my-server-vm.ova"  
      }  
    }  
  ],  
  "Status": "active",  
  "StatusMessage": "pending"
```

```
}
```

- Untuk detail API, lihat [ImportImage](#) di Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini mengimpor image mesin virtual disk tunggal dari bucket Amazon S3 yang ditentukan ke Amazon EC2 dengan token idempotensi. Contoh ini mengharuskan Peran Layanan Impor VM dengan nama default 'vmimport' ada, dengan kebijakan yang mengizinkan akses Amazon EC2 ke bucket yang ditentukan, seperti yang dijelaskan dalam topik Prerequisites Impor VM. Untuk menggunakan peran kustom, tentukan nama peran menggunakan - **RoleName** parameter.

```
$container = New-Object Amazon.EC2.Model.ImageDiskContainer
$container.Format="VMDK"
$container.UserBucket = New-Object Amazon.EC2.Model.UserBucket
$container.UserBucket.S3Bucket = "myVirtualMachineImages"
$container.UserBucket.S3Key = "Win_2008_Server_Standard_SP2_64-bit-disk1.vmdk"

$params = @{
    "ClientToken"="idempotencyToken"
    "Description"="Windows 2008 Standard Image Import"
    "Platform"="Windows"
    "LicenseType"="AWS"
}

Import-EC2Image -DiskContainer $container @params
```

Output:

```
Architecture      :
Description       : Windows 2008 Standard Image
Hypervisor        :
ImageId           :
ImportTaskId      : import-ami-abcdefgh
LicenseType       : AWS
Platform          : Windows
Progress          : 2
SnapshotDetails   : {}
```



```
Status      : active
StatusMessage : pending
```

- Untuk detail API, lihat [ImportImage](#) di Referensi AWS Tools for PowerShell Cmdlet.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **ImportKeyPair** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `ImportKeyPair`.

CLI

AWS CLI

Untuk mengimpor kunci publik

Pertama, buat key pair dengan alat pilihan Anda. Misalnya, gunakan perintah `ssh-keygen` ini:

Perintah:

```
ssh-keygen -t rsa -C "my-key" -f ~/.ssh/my-key
```

Output:

```
Generating public/private rsa key pair.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/ec2-user/.ssh/my-key.
Your public key has been saved in /home/ec2-user/.ssh/my-key.pub.
...
```

Perintah contoh ini mengimpor kunci publik yang ditentukan.

Perintah:

```
aws ec2 import-key-pair --key-name "my-key" --public-key-material fileb://~/.ssh/my-key.pub
```

Output:

```
{
  "KeyName": "my-key",
  "KeyFingerprint": "1f:51:ae:28:bf:89:e9:d8:1f:25:5d:37:2d:7d:b8:ca"
}
```

- Untuk detail API, lihat [ImportKeyPair](#) di Referensi AWS CLI Perintah.

PowerShell**Alat untuk PowerShell**

Contoh 1: Contoh ini mengimpor kunci publik ke EC2. Baris pertama menyimpan isi file kunci publik (*.pub) dalam variabel. **\$publickey** Selanjutnya, contoh mengkonversi format UTF8 dari file kunci publik ke string Base64-encoded, dan menyimpan string dikonversi dalam variabel. **\$pkbase64** Pada baris terakhir, kunci publik yang dikonversi diimpor ke EC2. Cmdlet mengembalikan sidik jari kunci dan nama sebagai hasil.

```
$publickey=[Io.File]::ReadAllText("C:\Users\TestUser\.ssh\id_rsa.pub")
$pkbase64 =
[System.Convert]::ToBase64String([System.Text.Encoding]::UTF8.GetBytes($publickey))
Import-EC2KeyPair -KeyName Example-user-key -PublicKey $pkbase64
```

Output:

```
KeyFingerprint                                KeyName
-----
do:d0:15:8f:79:97:12:be:00:fd:df:31:z3:b1:42:z1 Example-user-key
```

- Untuk detail API, lihat [ImportKeyPair](#) di Referensi AWS Tools for PowerShell Cmdlet.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan `ImportSnapshot` dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `ImportSnapshot`.

CLI

AWS CLI

Untuk mengimpor snapshot

`import-snapshot` Contoh berikut mengimpor disk yang ditentukan sebagai snapshot.

```
aws ec2 import-snapshot \  
  --description "My server VMDK" \  
  --disk-container Format=VMDK,UserBucket={S3Bucket=my-import-bucket,S3Key=vms/  
my-server-vm.vmdk}
```

Output:

```
{  
  "Description": "My server VMDK",  
  "ImportTaskId": "import-snap-1234567890abcdef0",  
  "SnapshotTaskDetail": {  
    "Description": "My server VMDK",  
    "DiskImageSize": "0.0",  
    "Format": "VMDK",  
    "Progress": "3",  
    "Status": "active",  
    "StatusMessage": "pending"  
    "UserBucket": {  
      "S3Bucket": "my-import-bucket",  
      "S3Key": "vms/my-server-vm.vmdk"  
    }  
  }  
}
```

- Untuk detail API, lihat [ImportSnapshot](#) di Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini mengimpor image disk VM format 'VMDK' ke snapshot Amazon EBS. Contoh ini memerlukan Peran Layanan Impor VM dengan nama default 'vmimport', dengan kebijakan yang mengizinkan akses Amazon EC2 ke bucket yang ditentukan, seperti

yang dijelaskan dalam topik di <http://docs.aws.amazon.com/2/latest/VM VM Import Prerequisites.html>. AWSEC WindowsGuide ImportPrerequisites Untuk menggunakan peran kustom, tentukan nama peran menggunakan **-RoleName** parameter.

```
$parms = @{
    "ClientToken"="idempotencyToken"
    "Description"="Disk Image Import"
    "DiskContainer_Description" = "Data disk"
    "DiskContainer_Format" = "VMDK"
    "DiskContainer_S3Bucket" = "myVirtualMachineImages"
    "DiskContainer_S3Key" = "datadiskimage.vmdk"
}

Import-EC2Snapshot @parms
```

Output:

Description	ImportTaskId	SnapshotTaskDetail
-----	-----	-----
Disk Image Import	import-snap-abcdefgh	Amazon.EC2.Model.SnapshotTaskDetail

- Untuk detail API, lihat [ImportSnapshot](#) di Referensi AWS Tools for PowerShell Cmdlet.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **ModifyCapacityReservation** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `ModifyCapacityReservation`.

CLI

AWS CLI

Contoh 1: Untuk mengubah jumlah instans yang dicadangkan oleh reservasi kapasitas yang ada

`modify-capacity-reservation` Contoh berikut mengubah jumlah contoh yang kapasitas cadangan kapasitas reservasi.

```
aws ec2 modify-capacity-reservation \
  --capacity-reservation-id cr-1234abcd56EXAMPLE \
  --instance-count 5
```

Output:

```
{
  "Return": true
}
```

Contoh 2: Untuk mengubah tanggal dan waktu akhir reservasi kapasitas yang ada

`modify-capacity-reservation` Contoh berikut memodifikasi reservasi kapasitas yang ada untuk berakhir pada tanggal dan waktu yang ditentukan.

```
aws ec2 modify-capacity-reservation \
  --capacity-reservation-id cr-1234abcd56EXAMPLE \
  --end-date-type limited \
  --end-date 2019-08-31T23:59:59Z
```

Untuk informasi selengkapnya, lihat [Memodifikasi Reservasi Kapasitas](#) di Panduan Pengguna Amazon Elastic Compute Cloud untuk Instans Linux.

- Untuk detail API, lihat [ModifyCapacityReservation](#) di Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini memodifikasi CapacityReservationId `cr-0c1f2345db6f7cdba` dengan mengubah hitungan instane menjadi 1

```
Edit-EC2CapacityReservation -CapacityReservationId cr-0c1f2345db6f7cdba -
InstanceCount 1
```

Output:

```
True
```

- Untuk detail API, lihat [ModifyCapacityReservation](#) di Referensi AWS Tools for PowerShell Cmdlet.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **ModifyHosts** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `ModifyHosts`.

CLI

AWS CLI

Contoh 1: Untuk mengaktifkan penempatan otomatis untuk Host Khusus

`modify-hosts` Contoh berikut memungkinkan penempatan otomatis untuk Host Khusus sehingga menerima peluncuran instans yang tidak ditargetkan yang cocok dengan konfigurasi tipe instance-nya.

```
aws ec2 modify-hosts \
  --host-id h-06c2f189b4EXAMPLE \
  --auto-placement on
```

Output:

```
{
  "Successful": [
    "h-06c2f189b4EXAMPLE"
  ],
  "Unsuccessful": []
}
```

Contoh 2: Untuk mengaktifkan pemulihan host untuk Host Khusus

`modify-hosts` Contoh berikut memungkinkan pemulihan host untuk Host Khusus yang ditentukan.

```
aws ec2 modify-hosts \
  --host-id h-06c2f189b4EXAMPLE \
  --host-recovery on
```

Output:

```
{
  "Successful": [
    "h-06c2f189b4EXAMPLE"
  ],
  "Unsuccessful": []
}
```

Untuk informasi selengkapnya, lihat [Memodifikasi Penempatan Otomatis Host Khusus](#) di Panduan Pengguna Amazon Elastic Compute Cloud untuk Instans Linux.

- Untuk detail API, lihat [ModifyHosts](#) di Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini memodifikasi AutoPlacement pengaturan ke off untuk host khusus h-01e23f4cd567890f3

```
Edit-EC2Host -HostId h-03e09f8cd681609f3 -AutoPlacement off
```

Output:

```
Successful          Unsuccessful
-----
{h-01e23f4cd567890f3} {}
```

- Untuk detail API, lihat [ModifyHosts](#) di Referensi AWS Tools for PowerShell Cmdlet.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **ModifyIdFormat** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `ModifyIdFormat`.

CLI

AWS CLI

Untuk mengaktifkan format ID yang lebih panjang untuk sumber daya

`modify-id-format` Contoh berikut memungkinkan format ID yang lebih panjang untuk jenis instance sumber daya.

```
aws ec2 modify-id-format \  
  --resource instance \  
  --use-long-ids
```

Untuk menonaktifkan format ID yang lebih panjang untuk sumber daya

`modify-id-format` Contoh berikut menonaktifkan format ID yang lebih panjang untuk jenis instance sumber daya.

```
aws ec2 modify-id-format \  
  --resource instance \  
  --no-use-long-ids
```

`modify-id-format` Contoh berikut memungkinkan format ID yang lebih panjang untuk semua jenis sumber daya yang didukung yang berada dalam periode keikutsertaannya.

```
aws ec2 modify-id-format \  
  --resource all-current \  
  --use-long-ids
```

- Untuk detail API, lihat [ModifyIdFormat](#) di Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini memungkinkan format ID yang lebih panjang untuk jenis sumber daya yang ditentukan.


```
Edit-EC2IdFormat -Resource instance -UseLongId $true
```

Contoh 2: Contoh ini menonaktifkan format ID yang lebih panjang untuk jenis sumber daya yang ditentukan.

```
Edit-EC2IdFormat -Resource instance -UseLongId $false
```

- Untuk detail API, lihat [ModifyIdFormat](#) di Referensi AWS Tools for PowerShell Cmdlet.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **ModifyImageAttribute** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `ModifyImageAttribute`.

CLI

AWS CLI

Contoh 1: Untuk membuat AMI publik

`modify-instance-attribute` Contoh berikut membuat AMI yang ditentukan publik.

```
aws ec2 modify-image-attribute \  
  --image-id ami-5731123e \  
  --launch-permission "Add=[{Group=all}]"
```

Perintah ini tidak menghasilkan output.

Contoh 2: Untuk membuat AMI pribadi

`modify-instance-attribute` Contoh berikut membuat AMI yang ditentukan menjadi pribadi.

```
aws ec2 modify-image-attribute \  
  --image-id ami-5731123e \  
  --launch-permission "Add=[{Group=all}]"
```

```
--launch-permission "Remove=[{Group=all}]"
```

Perintah ini tidak menghasilkan output.

Contoh 3: Untuk memberikan izin peluncuran ke AWS akun

`modify-instance-attribute` Contoh berikut memberikan izin peluncuran ke akun yang ditentukan AWS .

```
aws ec2 modify-image-attribute \  
  --image-id ami-5731123e \  
  --launch-permission "Add=[{UserId=123456789012}]"
```

Perintah ini tidak menghasilkan output.

Contoh 4: Untuk menghapus izin peluncuran dari AWS akun

`modify-instance-attribute` Contoh berikut menghapus izin peluncuran dari AWS akun yang ditentukan.

```
aws ec2 modify-image-attribute \  
  --image-id ami-5731123e \  
  --launch-permission "Remove=[{UserId=123456789012}]"
```

- Untuk detail API, lihat [ModifyImageAttribute](#) di Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini memperbarui deskripsi untuk AMI yang ditentukan.

```
Edit-EC2ImageAttribute -ImageId ami-12345678 -Description "New description"
```

Contoh 2: Contoh ini membuat AMI menjadi publik (misalnya, jadi siapa pun Akun AWS dapat menggunakannya).

```
Edit-EC2ImageAttribute -ImageId ami-12345678 -Attribute launchPermission -  
OperationType add -UserGroup all
```

Contoh 3: Contoh ini menjadikan AMI pribadi (misalnya, sehingga hanya Anda sebagai pemilik yang dapat menggunakannya).

```
Edit-EC2ImageAttribute -ImageId ami-12345678 -Attribute launchPermission -
OperationType remove -UserGroup all
```

Contoh 4: Contoh ini memberikan izin peluncuran ke yang ditentukan Akun AWS.

```
Edit-EC2ImageAttribute -ImageId ami-12345678 -Attribute launchPermission -
OperationType add -UserId 111122223333
```

Contoh 5: Contoh ini menghapus izin peluncuran dari yang ditentukan Akun AWS.

```
Edit-EC2ImageAttribute -ImageId ami-12345678 -Attribute launchPermission -
OperationType remove -UserId 111122223333
```

- Untuk detail API, lihat [ModifyImageAttribute](#) di Referensi AWS Tools for PowerShell Cmdlet.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **ModifyInstanceAttribute** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `ModifyInstanceAttribute`.

CLI

AWS CLI

Contoh 1: Untuk memodifikasi jenis instance

`modify-instance-attribute` Contoh berikut memodifikasi jenis instance dari instance tertentu. Instans harus berada dalam status `stopped`.

```
aws ec2 modify-instance-attribute \
  --instance-id i-1234567890abcdef0 \
  --instance-type "{\"Value\": \"m1.small\"}"
```

Perintah ini tidak menghasilkan output.

Contoh 2: Untuk mengaktifkan jaringan yang disempurnakan pada sebuah instance

`modify-instance-attribute` Contoh berikut memungkinkan jaringan ditingkatkan untuk contoh tertentu. Instans harus berada dalam status `stopped`.

```
aws ec2 modify-instance-attribute \  
  --instance-id i-1234567890abcdef0 \  
  --sriov-net-support simple
```

Perintah ini tidak menghasilkan output.

Contoh 3: Untuk memodifikasi `sourceDestCheck` atribut

`modify-instance-attribute` Contoh berikut menetapkan `sourceDestCheck` atribut dari contoh yang ditentukan untuk `true`. Instans harus dalam VPC.

```
aws ec2 modify-instance-attribute --instance-id i-1234567890abcdef0 --source-  
dest-check "{\"Value\": true}"
```

Perintah ini tidak menghasilkan output.

Contoh 4: Untuk memodifikasi `deleteOnTermination` atribut volume root

`modify-instance-attribute` Contoh berikut menetapkan `deleteOnTermination` atribut untuk volume root instance yang didukung Amazon EBS yang ditentukan. `false` Secara default, atribut ini adalah `true` untuk volume root.

Perintah:

```
aws ec2 modify-instance-attribute \  
  --instance-id i-1234567890abcdef0 \  
  --block-device-mappings "[{\"DeviceName\": \"/dev/sda1\", \"Ebs\":  
{\"DeleteOnTermination\": false}}]"
```

Perintah ini tidak menghasilkan output.

Contoh 5: Untuk memodifikasi data pengguna yang dilampirkan ke sebuah instance

`modify-instance-attribute` Contoh berikut menambahkan isi file `UserData.txt` sebagai `UserData` untuk contoh yang ditentukan.

Isi file asli `UserData.txt`:

```
#!/bin/bash
yum update -y
service httpd start
chkconfig httpd on
```

Isi file harus dikodekan base64. Perintah pertama mengonversi file teks ke base64 dan menyimpannya sebagai file baru.

Versi perintah Linux/macOS:

```
base64 UserData.txt > UserData.base64.txt
```

Perintah ini tidak menghasilkan output.

Versi perintah Windows:

```
certutil -encode UserData.txt tmp.b64 && findstr /v /c:- tmp.b64 >
UserData.base64.txt
```

Output:

```
Input Length = 67
Output Length = 152
CertUtil: -encode command completed successfully.
```

Sekarang Anda dapat mereferensikan file itu dalam perintah CLI yang berikut:

```
aws ec2 modify-instance-attribute \
  --instance-id=i-09b5a14dbca622e76 \
  --attribute userData --value file://UserData.base64.txt
```

Perintah ini tidak menghasilkan output.

Untuk informasi selengkapnya, lihat [Data Pengguna dan AWS CLI](#) di Panduan Pengguna EC2.

- Untuk detail API, lihat [ModifyInstanceAttribute](#) di Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini memodifikasi jenis instance dari instance yang ditentukan.

```
Edit-EC2InstanceAttribute -InstanceId i-12345678 -InstanceType m3.medium
```

Contoh 2: Contoh ini memungkinkan peningkatan jaringan untuk instance tertentu, dengan menentukan "sederhana" sebagai nilai dari parameter dukungan jaringan virtualisasi I/O root tunggal (SR-IOV), `-SriovNetSupport`

```
Edit-EC2InstanceAttribute -InstanceId i-12345678 -SriovNetSupport "simple"
```

Contoh 3: Contoh ini memodifikasi grup keamanan untuk instance tertentu. Instans harus dalam VPC. Anda harus menentukan ID dari setiap grup keamanan, bukan nama.

```
Edit-EC2InstanceAttribute -InstanceId i-12345678 -Group @( "sg-12345678",  
"sg-45678901" )
```

Contoh 4: Contoh ini memungkinkan optimasi EBS I/O untuk instance tertentu. Fitur ini tidak tersedia dengan semua jenis instance. Biaya penggunaan tambahan berlaku saat menggunakan instans yang dioptimalkan EBS.

```
Edit-EC2InstanceAttribute -InstanceId i-12345678 -EbsOptimized $true
```

Contoh 5: Contoh ini memungkinkan pemeriksaan sumber/tujuan untuk contoh yang ditentukan. Untuk instance NAT untuk melakukan NAT, nilainya harus 'salah'.

```
Edit-EC2InstanceAttribute -InstanceId i-12345678 -SourceDestCheck $true
```

Contoh 6: Contoh ini menonaktifkan penghentian untuk contoh yang ditentukan.

```
Edit-EC2InstanceAttribute -InstanceId i-12345678 -DisableApiTermination $true
```

Contoh 7: Contoh ini mengubah instance tertentu sehingga berakhir ketika shutdown dimulai dari instance.

```
Edit-EC2InstanceAttribute -InstanceId i-12345678 -  
InstanceInitiatedShutdownBehavior terminate
```

- Untuk detail API, lihat [ModifyInstanceAttribute](#) di Referensi AWS Tools for PowerShell Cmdlet.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **ModifyInstanceCreditSpecification** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `ModifyInstanceCreditSpecification`.

CLI

AWS CLI

Untuk memodifikasi opsi kredit untuk penggunaan CPU dari sebuah instance

Contoh ini memodifikasi opsi kredit untuk penggunaan CPU dari instance yang ditentukan di wilayah tertentu menjadi “tidak terbatas”. Opsi kredit yang valid adalah “standar” dan “tidak terbatas”.

Perintah:

```
aws ec2 modify-instance-credit-specification --instance-credit-specification  
"InstanceId=i-1234567890abcdef0,CpuCredits=unlimited"
```

Output:

```
{  
  "SuccessfulInstanceCreditSpecifications": [  
    {  
      "InstanceId": "i-1234567890abcdef0"  
    }  
  ],  
  "UnsuccessfulInstanceCreditSpecifications": []  
}
```

```
}
```

- Untuk detail API, lihat [ModifyInstanceCreditSpecification](#) di Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell

Contoh 1: Ini memungkinkan kredit tak terbatas T2 misalnya i-01234567890abcdef.

```
$Credit = New-Object -TypeName  
    Amazon.EC2.Model.InstanceCreditSpecificationRequest  
$Credit.InstanceId = "i-01234567890abcdef"  
$Credit.CpuCredits = "unlimited"  
Edit-EC2InstanceCreditSpecification -InstanceCreditSpecification $Credit
```

- Untuk detail API, lihat [ModifyInstanceCreditSpecification](#) di Referensi AWS Tools for PowerShell Cmdlet.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **ModifyNetworkInterfaceAttribute** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `ModifyNetworkInterfaceAttribute`.

CLI

AWS CLI

Untuk memodifikasi atribut lampiran antarmuka jaringan

Perintah contoh ini memodifikasi attachment atribut antarmuka jaringan yang ditentukan.

Perintah:

```
aws ec2 modify-network-interface-attribute --network-interface-id eni-686ea200 --  
attachment AttachmentId=eni-attach-43348162,DeleteOnTermination=false
```


Untuk memodifikasi atribut deskripsi antarmuka jaringan

Perintah contoh ini memodifikasi `description` atribut antarmuka jaringan yang ditentukan.

Perintah:

```
aws ec2 modify-network-interface-attribute --network-interface-id eni-686ea200 --description "My description"
```

Untuk memodifikasi atribut `groupSet` dari antarmuka jaringan

Perintah contoh ini memodifikasi `groupSet` atribut antarmuka jaringan yang ditentukan.

Perintah:

```
aws ec2 modify-network-interface-attribute --network-interface-id eni-686ea200 --groups sg-903004f8 sg-1a2b3c4d
```

Untuk memodifikasi `sourceDestCheck` atribut antarmuka jaringan

Perintah contoh ini memodifikasi `sourceDestCheck` atribut antarmuka jaringan yang ditentukan.

Perintah:

```
aws ec2 modify-network-interface-attribute --network-interface-id eni-686ea200 --no-source-dest-check
```

- Untuk detail API, lihat [ModifyNetworkInterfaceAttribute](#) di Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini memodifikasi antarmuka jaringan yang ditentukan sehingga lampiran yang ditentukan dihapus pada penghentian.

```
Edit-EC2NetworkInterfaceAttribute -NetworkInterfaceId eni-1a2b3c4d -Attachment_AttachmentId eni-attach-1a2b3c4d -Attachment_DeleteOnTermination $true
```

Contoh 2: Contoh ini memodifikasi deskripsi antarmuka jaringan yang ditentukan.

```
Edit-EC2NetworkInterfaceAttribute -NetworkInterfaceId eni-1a2b3c4d -Description "my description"
```

Contoh 3: Contoh ini memodifikasi grup keamanan untuk antarmuka jaringan yang ditentukan.

```
Edit-EC2NetworkInterfaceAttribute -NetworkInterfaceId eni-1a2b3c4d -Groups sg-1a2b3c4d
```

Contoh 4: Contoh ini menonaktifkan pemeriksaan sumber/tujuan untuk antarmuka jaringan yang ditentukan.

```
Edit-EC2NetworkInterfaceAttribute -NetworkInterfaceId eni-1a2b3c4d -SourceDestCheck $false
```

- Untuk detail API, lihat [ModifyNetworkInterfaceAttribute](#) di Referensi AWS Tools for PowerShell Cmdlet.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **ModifyReservedInstances** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `ModifyReservedInstances`.

CLI

AWS CLI

Untuk memodifikasi Instans Cadangan

Perintah contoh ini memindahkan Instans Cadangan ke Availability Zone lain di wilayah yang sama.

Perintah:

```
aws ec2 modify-reserved-instances --reserved-instances-ids b847fa93-e282-4f55-b59a-1342f5bd7c02 --target-configurations AvailabilityZone=us-west-1c,Platform=EC2-Classical,InstanceCount=10
```

Output:

```
{
  "ReservedInstancesModificationId": "rimod-d3ed4335-b1d3-4de6-ab31-0f13aaf46687"
}
```

Untuk memodifikasi platform jaringan Instans Cadangan

Perintah contoh ini mengonversi EC2-Classic Reserved Instances ke EC2-VPC.

Perintah:

```
aws ec2 modify-reserved-instances --reserved-instances-ids f127bd27-
edb7-44c9-a0eb-0d7e09259af0 --target-configurations AvailabilityZone=us-
west-1c,Platform=EC2-VPC,InstanceCount=5
```

Output:

```
{
  "ReservedInstancesModificationId": "rimod-82fa9020-668f-4fb6-945d-61537009d291"
}
```

Untuk informasi selengkapnya, lihat Memodifikasi Instans Cadangan Anda di Panduan Pengguna Amazon EC2.

Untuk mengubah ukuran instans dari Instans Cadangan

Perintah contoh ini memodifikasi Instans Cadangan yang memiliki 10 instance Linux/Unix m1.small di us-west-1c sehingga 8 instance m1.small menjadi 2 instance m1.large, dan 2 m1.small sisanya menjadi 1 m1.medium instance di Availability Zone yang sama. Perintah:

```
aws ec2 modify-reserved-instances --reserved-instances-
ids 1ba8e2e3-3556-4264-949e-63ee671405a9 --target-
configurations AvailabilityZone=us-west-1c,Platform=EC2-
Classic,InstanceCount=2,InstanceType=m1.large AvailabilityZone=us-
west-1c,Platform=EC2-Classic,InstanceCount=1,InstanceType=m1.medium
```

Output:

```
{
  "ReservedInstancesModificationId": "rimod-acc5f240-080d-4717-
b3e3-1c6b11fa00b6"
}
```

```
}
```

Untuk informasi selengkapnya, lihat [Memodifikasi Ukuran Instans Reservasi Anda](#) di Panduan Pengguna Amazon EC2.

- Untuk detail API, lihat [ModifyReservedInstances](#) di Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini memodifikasi Availability Zone, jumlah instans, dan platform untuk instans Cadangan yang ditentukan.

```
$config = New-Object Amazon.EC2.Model.ReservedInstancesConfiguration
$config.AvailabilityZone = "us-west-2a"
$config.InstanceCount = 1
$config.Platform = "EC2-VPC"

Edit-EC2ReservedInstance `
-ReservedInstancesId @"FE32132D-70D5-4795-B400-AE435EXAMPLE",
"0CC556F3-7AB8-4C00-B0E5-98666EXAMPLE" `
-TargetConfiguration $config
```

- Untuk detail API, lihat [ModifyReservedInstances](#) di Referensi AWS Tools for PowerShell Cmdlet.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **ModifySnapshotAttribute** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `ModifySnapshotAttribute`.

CLI

AWS CLI

Contoh 1: Untuk memodifikasi atribut snapshot

`modify-snapshot-attribute` Contoh berikut memperbarui `createVolumePermission` atribut untuk snapshot yang ditentukan, menghapus izin volume untuk pengguna tertentu.

```
aws ec2 modify-snapshot-attribute \  
  --snapshot-id snap-1234567890abcdef0 \  
  --attribute createVolumePermission \  
  --operation-type remove \  
  --user-ids 123456789012
```

Contoh 2: Untuk membuat snapshot publik

`modify-snapshot-attribute` Contoh berikut membuat snapshot yang ditentukan publik.

```
aws ec2 modify-snapshot-attribute \  
  --snapshot-id snap-1234567890abcdef0 \  
  --attribute createVolumePermission \  
  --operation-type add \  
  --group-names all
```

- Untuk detail API, lihat [ModifySnapshotAttribute](#) di Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini membuat snapshot yang ditentukan publik dengan menyetel `CreateVolumePermission` atributnya.

```
Edit-EC2SnapshotAttribute -SnapshotId snap-12345678 -Attribute  
CreateVolumePermission -OperationType Add -GroupName all
```

- Untuk detail API, lihat [ModifySnapshotAttribute](#) di Referensi AWS Tools for PowerShell Cmdlet.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **ModifySpotFleetRequest** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `ModifySpotFleetRequest`.

CLI

AWS CLI

Untuk mengubah permintaan armada Spot

Perintah contoh ini memperbarui kapasitas target permintaan armada Spot yang ditentukan.

Perintah:

```
aws ec2 modify-spot-fleet-request --target-capacity 20 --spot-fleet-request-id
sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE
```

Output:

```
{
  "Return": true
}
```

Perintah contoh ini mengurangi kapasitas target permintaan armada Spot yang ditentukan tanpa menghentikan Instans Spot apa pun sebagai hasilnya.

Perintah:

```
aws ec2 modify-spot-fleet-request --target-capacity 10 --excess-capacity-
termination-policy NoTermination --spot-fleet-request-ids sfr-73fbd2ce-
aa30-494c-8788-1cee4EXAMPLE
```

Output:

```
{
  "Return": true
}
```

- Untuk detail API, lihat [ModifySpotFleetRequest](#) di Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini memperbarui kapasitas target permintaan armada Spot yang ditentukan.

```
Edit-EC2SpotFleetRequest -SpotFleetRequestId sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE -TargetCapacity 10
```

Output:

```
True
```

- Untuk detail API, lihat [ModifySpotFleetRequest](#) di Referensi AWS Tools for PowerShell Cmdlet.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **ModifySubnetAttribute** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `ModifySubnetAttribute`.

CLI

AWS CLI

Untuk mengubah perilaku pengalamatan IPv4 publik subnet

Contoh ini memodifikasi subnet-1a2b3c4d untuk menentukan bahwa semua instance yang diluncurkan ke subnet ini diberi alamat IPv4 publik. Jika perintah berhasil, tidak ada output yang akan ditampilkan.

Perintah:

```
aws ec2 modify-subnet-attribute --subnet-id subnet-1a2b3c4d --map-public-ip-on-launch
```

Untuk mengubah perilaku pengalamatan IPv6 subnet

Contoh ini memodifikasi subnet-1a2b3c4d untuk menentukan bahwa semua instance yang diluncurkan ke subnet ini diberi alamat IPv6 dari kisaran subnet.

Perintah:

```
aws ec2 modify-subnet-attribute --subnet-id subnet-1a2b3c4d --assign-ipv6-address-on-creation
```

Untuk informasi selengkapnya, lihat Pengalamatan IP di VPC Anda di AWS Panduan Pengguna Virtual Private Cloud.

- Untuk detail API, lihat [ModifySubnetAttribute](#) di Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini memungkinkan pengalamatan IP publik untuk subnet yang ditentukan.

```
Edit-EC2SubnetAttribute -SubnetId subnet-1a2b3c4d -MapPublicIpOnLaunch $true
```

Contoh 2: Contoh ini menonaktifkan pengalamatan IP publik untuk subnet yang ditentukan.

```
Edit-EC2SubnetAttribute -SubnetId subnet-1a2b3c4d -MapPublicIpOnLaunch $false
```

- Untuk detail API, lihat [ModifySubnetAttribute](#) di Referensi AWS Tools for PowerShell Cmdlet.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **ModifyVolumeAttribute** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `ModifyVolumeAttribute`.

CLI

AWS CLI

Untuk memodifikasi atribut volume

Contoh ini menetapkan `autoEnableIo` atribut volume dengan ID `vol-1234567890abcdef0` ke `true`. Jika perintah berhasil, tidak ada output yang akan ditampilkan.

Perintah:

```
aws ec2 modify-volume-attribute --volume-id vol-1234567890abcdef0 --auto-enable-io
```

- Untuk detail API, lihat [ModifyVolumeAttribute](#) di Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini memodifikasi atribut tertentu dari volume yang ditentukan. Operasi I/O untuk volume secara otomatis dilanjutkan setelah ditangguhkan karena data yang berpotensi tidak konsisten.

```
Edit-EC2VolumeAttribute -VolumeId vol-12345678 -AutoEnableIO $true
```

- Untuk detail API, lihat [ModifyVolumeAttribute](#) di Referensi AWS Tools for PowerShell Cmdlet.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **ModifyVpcAttribute** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `ModifyVpcAttribute`.

CLI

AWS CLI

Untuk memodifikasi `enableDnsSupport` atribut

Contoh ini memodifikasi `enableDnsSupport` atribut. Atribut ini menunjukkan apakah resolusi DNS diaktifkan untuk VPC. Jika atribut `inittrue`, server DNS Amazon menyelesaikan nama

host DNS untuk instance Anda ke alamat IP yang sesuai; jika tidak, tidak. Jika perintah berhasil, tidak ada output yang akan ditampilkan.

Perintah:

```
aws ec2 modify-vpc-attribute --vpc-id vpc-a01106c2 --enable-dns-support "{\"Value\":false}"
```

Untuk memodifikasi enableDnsHostnames atribut

Contoh ini memodifikasi enableDnsHostnames atribut. Atribut ini menunjukkan apakah instance yang diluncurkan di VPC mendapatkan nama host DNS. Jika atribut ini true, instance di VPC mendapatkan nama host DNS; jika tidak, mereka tidak. Jika perintah berhasil, tidak ada output yang akan ditampilkan.

Perintah:

```
aws ec2 modify-vpc-attribute --vpc-id vpc-a01106c2 --enable-dns-hostnames "{\"Value\":false}"
```

- Untuk detail API, lihat [ModifyVpcAttribute](#) di Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini memungkinkan dukungan untuk nama host DNS untuk VPC yang ditentukan.

```
Edit-EC2VpcAttribute -VpcId vpc-12345678 -EnableDnsHostnames $true
```

Contoh 2: Contoh ini menonaktifkan dukungan untuk nama host DNS untuk VPC yang ditentukan.

```
Edit-EC2VpcAttribute -VpcId vpc-12345678 -EnableDnsHostnames $false
```

Contoh 3: Contoh ini memungkinkan dukungan untuk resolusi DNS untuk VPC yang ditentukan.

```
Edit-EC2VpcAttribute -VpcId vpc-12345678 -EnableDnsSupport $true
```

Contoh 4: Contoh ini menonaktifkan dukungan untuk resolusi DNS untuk VPC yang ditentukan.

```
Edit-EC2VpcAttribute -VpcId vpc-12345678 -EnableDnsSupport $false
```

- Untuk detail API, lihat [ModifyVpcAttribute](#) di Referensi AWS Tools for PowerShell Cmdlet.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **MonitorInstances** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `MonitorInstances`.

C++

SDK untuk C++

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
Aws::EC2::EC2Client ec2Client(clientConfiguration);
Aws::EC2::Model::MonitorInstancesRequest request;
request.AddInstanceIds(instanceId);
request.SetDryRun(true);

auto dry_run_outcome = ec2Client.MonitorInstances(request);
if (dry_run_outcome.IsSuccess()) {
    std::cerr
        << "Failed dry run to enable monitoring on instance. A dry run
should trigger an error."
        <<
        std::endl;
```

```

        return false;
    }
    else if (dry_run_outcome.GetError().GetErrorType()
        != Aws::EC2::EC2Errors::DRY_RUN_OPERATION) {
        std::cerr << "Failed dry run to enable monitoring on instance " <<
            instanceId << ": " << dry_run_outcome.GetError().GetMessage()
<<
            std::endl;
        return false;
    }

    request.SetDryRun(false);
    auto monitorInstancesOutcome = ec2Client.MonitorInstances(request);
    if (!monitorInstancesOutcome.IsSuccess()) {
        std::cerr << "Failed to enable monitoring on instance " <<
            instanceId << ": " <<
            monitorInstancesOutcome.GetError().GetMessage() << std::endl;
    }
    else {
        std::cout << "Successfully enabled monitoring on instance " <<
            instanceId << std::endl;
    }
}

```

- Untuk detail API, lihat [MonitorInstances](#) di Referensi AWS SDK for C++ API.

CLI

AWS CLI

Untuk mengaktifkan pemantauan yang mendetail untuk instans

Perintah contoh ini mengaktifkan pemantauan yang mendetail untuk instans tertentu.

Perintah:

```
aws ec2 monitor-instances --instance-ids i-1234567890abcdef0
```

Output:

```
{
  "InstanceMonitorings": [
```

```
{
  "InstanceId": "i-1234567890abcdef0",
  "Monitoring": {
    "State": "pending"
  }
}
]
```

- Untuk detail API, lihat [MonitorInstances](#) di Referensi AWS CLI Perintah.

JavaScript

SDK untuk JavaScript (v3)

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
import { MonitorInstancesCommand } from "@aws-sdk/client-ec2";

import { client } from "../libs/client.js";

// Turn on detailed monitoring for the selected instance.
// By default, metrics are sent to Amazon CloudWatch every 5 minutes.
// For a cost you can enable detailed monitoring which sends metrics every
minute.
export const main = async () => {
  const command = new MonitorInstancesCommand({
    InstanceIds: ["INSTANCE_ID"],
  });

  try {
    const { InstanceMonitorings } = await client.send(command);
    const instancesBeingMonitored = InstanceMonitorings.map(
      (im) =>
        ` • Detailed monitoring state for ${im.InstanceId} is
${im.Monitoring.State}.`,
    );
    console.log("Monitoring status:");
```

```

    console.log(instancesBeingMonitored.join("\n"));
  } catch (err) {
    console.error(err);
  }
};

```

- Untuk detail API, lihat [MonitorInstances](#) di Referensi AWS SDK for JavaScript API.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini memungkinkan pemantauan terperinci untuk contoh yang ditentukan.

```
Start-EC2InstanceMonitoring -InstanceId i-12345678
```

Output:

```

InstanceId      Monitoring
-----
i-12345678     Amazon.EC2.Model.Monitoring

```

- Untuk detail API, lihat [MonitorInstances](#) di Referensi AWS Tools for PowerShell Cmdlet.

SAP ABAP

SDK untuk SAP ABAP

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```

DATA lt_instance_ids TYPE /aws1/
cl_ec2instidstringlist_w=>tt_instanceidstringlist.
APPEND NEW /aws1/cl_ec2instidstringlist_w( iv_value = iv_instance_id ) TO
lt_instance_ids.

```

```

"Perform dry run"
TRY.
    " DryRun is set to true. This checks for the required permissions to
    monitor the instance without actually making the request. "
    lo_ec2->monitorinstances(
        it_instanceids = lt_instance_ids
        iv_dryrun = abap_true
    ).
CATCH /aws1/cx_rt_service_generic INTO DATA(lo_exception).
    " If the error code returned is `DryRunOperation`, then you have the
    required permissions to monitor this instance. "
    IF lo_exception->av_err_code = 'DryRunOperation'.
        MESSAGE 'Dry run to enable detailed monitoring completed.' TYPE 'I'.
        " DryRun is set to false to enable detailed monitoring. "
        lo_ec2->monitorinstances(
            it_instanceids = lt_instance_ids
            iv_dryrun = abap_false
        ).
        MESSAGE 'Detailed monitoring enabled.' TYPE 'I'.
        " If the error code returned is `UnauthorizedOperation`, then you don't
        have the required permissions to monitor this instance. "
        ELSEIF lo_exception->av_err_code = 'UnauthorizedOperation'.
            MESSAGE 'Dry run to enable detailed monitoring failed. User does not
            have the permissions to monitor the instance.' TYPE 'E'.
        ELSE.
            DATA(lv_error) = |"{ lo_exception->av_err_code }" - { lo_exception-
            >av_err_msg }|.
            MESSAGE lv_error TYPE 'E'.
        ENDIF.
    ENDTRY.

```

- Untuk detail API, lihat [MonitorInstances](#) di AWS SDK untuk referensi SAP ABAP API.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **MoveAddressToVpc** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `MoveAddressToVpc`.

CLI

AWS CLI

Untuk memindahkan alamat ke EC2-VPC

Contoh ini memindahkan alamat IP Elastis 54.123.4.56 ke platform EC2-VPC.

Perintah:

```
aws ec2 move-address-to-vpc --public-ip 54.123.4.56
```

Output:

```
{
  "Status": "MoveInProgress"
}
```

- Untuk detail API, lihat [MoveAddressToVpc](#) di Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini memindahkan instans EC2 dengan alamat IP publik 12.345.67.89 ke platform EC2-VPC di wilayah AS Timur (Virginia Utara).

```
Move-EC2AddressToVpc -PublicIp 12.345.67.89 -Region us-east-1
```

Contoh 2: Contoh ini menyalurkan hasil Get-EC2Instance perintah ke Move-EC2AddressToVpc cmdlet. Get-EC2InstancePerintah mendapat instance yang ditentukan oleh ID instance, kemudian mengembalikan properti alamat IP publik dari instance tersebut.

```
(Get-EC2Instance -Instance i-12345678).Instances.PublicIpAddress | Move-EC2AddressToVpc
```

- Untuk detail API, lihat [MoveAddressToVpc](#) di Referensi AWS Tools for PowerShell Cmdlet.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **PurchaseHostReservation** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `PurchaseHostReservation`.

CLI

AWS CLI

Untuk membeli Reservasi Tuan Rumah Khusus

Contoh ini membeli penawaran Reservasi Host Khusus yang ditentukan untuk Host Khusus yang ditentukan di akun Anda.

Perintah:

```
aws ec2 purchase-host-reservation --offering-id hro-03f707bf363b6b324 --host-id-set h-013abcd2a00cbd123
```

Output:

```
{
  "TotalHourlyPrice": "1.499",
  "Purchase": [
    {
      "HourlyPrice": "1.499",
      "InstanceFamily": "m4",
      "PaymentOption": "NoUpfront",
      "HostIdSet": [
        "h-013abcd2a00cbd123"
      ],
      "HostReservationId": "hr-0d418a3a4ffc669ae",
      "UpfrontPrice": "0.000",
      "Duration": 31536000
    }
  ],
  "TotalUpfrontPrice": "0.000"
}
```

- Untuk detail API, lihat [PurchaseHostReservation](#) di Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini membeli reservasi yang menawarkan hro-0c1f23456789d0ab dengan konfigurasi yang cocok dengan Host Khusus Anda h-01e23f4cd567890f1

```
New-EC2HostReservation -OfferingId hro-0c1f23456789d0ab HostIdSet
h-01e23f4cd567890f1
```

Output:

```
ClientToken      :
CurrencyCode     :
Purchase         : {hr-0123f4b5d67bedc89}
TotalHourlyPrice : 1.307
TotalUpfrontPrice : 0.000
```

- Untuk detail API, lihat [PurchaseHostReservation](#) di Referensi AWS Tools for PowerShell Cmdlet.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **PurchaseScheduledInstances** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `PurchaseScheduledInstances`.

CLI

AWS CLI

Untuk membeli Instance Terjadwal

Contoh ini membeli Instance Terjadwal.

Perintah:

```
aws ec2 purchase-scheduled-instances --purchase-requests file://purchase-
request.json
```

Permintaan pembelian.json:

```
[
  {
    "PurchaseToken": "eyJ2IjoiMSIsInMiOjEsImMiOi...",
    "InstanceCount": 1
  }
]
```

Output:

```
{
  "ScheduledInstanceSet": [
    {
      "AvailabilityZone": "us-west-2b",
      "ScheduledInstanceId": "sci-1234-1234-1234-1234-123456789012",
      "HourlyPrice": "0.095",
      "CreateDate": "2016-01-25T21:43:38.612Z",
      "Recurrence": {
        "OccurrenceDaySet": [
          1
        ],
        "Interval": 1,
        "Frequency": "Weekly",
        "OccurrenceRelativeToEnd": false,
        "OccurrenceUnit": ""
      },
      "Platform": "Linux/UNIX",
      "TermEndDate": "2017-01-31T09:00:00Z",
      "InstanceCount": 1,
      "SlotDurationInHours": 32,
      "TermStartDate": "2016-01-31T09:00:00Z",
      "NetworkPlatform": "EC2-VPC",
      "TotalScheduledInstanceHours": 1696,
      "NextSlotStartTime": "2016-01-31T09:00:00Z",
      "InstanceType": "c4.large"
    }
  ]
}
```

- Untuk detail API, lihat [PurchaseScheduledInstances](#) di Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini membeli Instance Terjadwal.

```
$request = New-Object Amazon.EC2.Model.PurchaseRequest
$request.InstanceCount = 1
$request.PurchaseToken = "eyJ2IjoiMSIsInMiOjEsImMiOi..."
New-EC2ScheduledInstancePurchase -PurchaseRequest $request
```

Output:

```
AvailabilityZone      : us-west-2b
CreateDate            : 1/25/2016 1:43:38 PM
HourlyPrice           : 0.095
InstanceCount        : 1
InstanceType         : c4.large
NetworkPlatform      : EC2-VPC
NextSlotStartTime    : 1/31/2016 1:00:00 AM
Platform             : Linux/UNIX
PreviousSlotEndTime  :
Recurrence           : Amazon.EC2.Model.ScheduledInstanceRecurrence
ScheduledInstanceId  : sci-1234-1234-1234-1234-123456789012
SlotDurationInHours  : 32
TermEndDate          : 1/31/2017 1:00:00 AM
TermStartDate        : 1/31/2016 1:00:00 AM
TotalScheduledInstanceHours : 1696
```

- Untuk detail API, lihat [PurchaseScheduledInstances](#) di Referensi AWS Tools for PowerShell Cmdlet.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **RebootInstances** dengan AWS SDK atau CLI


Contoh kode berikut menunjukkan cara menggunakan `RebootInstances`.

Contoh tindakan adalah kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Anda dapat melihat tindakan ini dalam konteks dalam contoh kode berikut:

- [Membangun dan mengelola layanan yang tangguh](#)

.NET

AWS SDK for .NET

 Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
/// <summary>
/// Reboot EC2 instances.
/// </summary>
/// <param name="ec2InstanceId">The instance Id of the instances that will be
rebooted.</param>
/// <returns>Async task.</returns>
public async Task RebootInstances(string ec2InstanceId)
{
    var request = new RebootInstancesRequest
    {
        InstanceIds = new List<string> { ec2InstanceId },
    };

    var response = await _amazonEC2.RebootInstancesAsync(request);
    if (response.HttpStatusCode == System.Net.HttpStatusCode.OK)
    {
        Console.WriteLine("Instances successfully rebooted.");
    }
    else
    {
        Console.WriteLine("Could not reboot one or more instances.");
    }
}
```

Ganti profil untuk instans, boot ulang, dan mulai ulang server web.

```
/// <summary>
/// Replace the profile associated with a running instance. After the profile
is replaced, the instance
/// is rebooted to ensure that it uses the new profile. When the instance is
ready, Systems Manager is
/// used to restart the Python web server.
/// </summary>
/// <param name="instanceId">The Id of the instance to update.</param>
/// <param name="credsProfileName">The name of the new profile to associate
with the specified instance.</param>
/// <param name="associationId">The Id of the existing profile association
for the instance.</param>
/// <returns>Async task.</returns>
public async Task ReplaceInstanceProfile(string instanceId, string
credsProfileName, string associationId)
{
    await _amazonEc2.ReplaceIamInstanceProfileAssociationAsync(
        new ReplaceIamInstanceProfileAssociationRequest()
        {
            AssociationId = associationId,
            IamInstanceProfile = new IamInstanceProfileSpecification()
            {
                Name = credsProfileName
            }
        });
    // Allow time before resetting.
    Thread.Sleep(25000);
    var instanceReady = false;
    var retries = 5;
    while (retries-- > 0 && !instanceReady)
    {
        await _amazonEc2.RebootInstancesAsync(
            new RebootInstancesRequest(new List<string>() { instanceId }));
        Thread.Sleep(10000);

        var instancesPaginator =
        _amazonSsm.Paginators.DescribeInstanceInformation(
            new DescribeInstanceInformationRequest());
        // Get the entire list using the paginator.
        await foreach (var instance in
instancesPaginator.InstanceInformationList)
        {
            instanceReady = instance.InstanceId == instanceId;
        }
    }
}
```

```

        if (instanceReady)
        {
            break;
        }
    }
}
Console.WriteLine($"Sending restart command to instance {instanceId}");
await _amazonSsm.SendCommandAsync(
    new SendCommandRequest()
    {
        InstanceIds = new List<string>() { instanceId },
        DocumentName = "AWS-RunShellScript",
        Parameters = new Dictionary<string, List<string>>()
        {
            {"commands", new List<string>() { "cd / && sudo python3
server.py 80" }}
        }
    });
Console.WriteLine($"Restarted the web server on instance {instanceId}");
}

```

- Untuk detail API, lihat [RebootInstances](#) di Referensi AWS SDK for .NET API.

C++

SDK untuk C++

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```

Aws::EC2::EC2Client ec2Client(clientConfiguration);

Aws::EC2::Model::RebootInstancesRequest request;
request.AddInstanceIds(instanceId);
request.SetDryRun(true);

auto dry_run_outcome = ec2Client.RebootInstances(request);
if (dry_run_outcome.IsSuccess()) {

```

```
        std::cerr
            << "Failed dry run to reboot on instance. A dry run should
trigger an error."
            <<
            std::endl;
        return false;
    }
    else if (dry_run_outcome.GetError().GetErrorType()
        != Aws::EC2::EC2Errors::DRY_RUN_OPERATION) {
        std::cout << "Failed dry run to reboot instance " << instanceId << ": "
            << dry_run_outcome.GetError().GetMessage() << std::endl;
        return false;
    }

    request.SetDryRun(false);
    auto outcome = ec2Client.RebootInstances(request);
    if (!outcome.IsSuccess()) {
        std::cout << "Failed to reboot instance " << instanceId << ": " <<
            outcome.GetError().GetMessage() << std::endl;
    }
    else {
        std::cout << "Successfully rebooted instance " << instanceId <<
            std::endl;
    }
}
```

- Untuk detail API, lihat [RebootInstances](#) di Referensi AWS SDK for C++ API.

CLI

AWS CLI

Untuk melakukan boot ulang instans Amazon EC2

Contoh ini melakukan boot ulang instans tertentu. Jika perintah berhasil, tidak ada output yang akan ditampilkan.

Perintah:

```
aws ec2 reboot-instances --instance-ids i-1234567890abcdef5
```


Untuk informasi selengkapnya, lihat Melakukan Boot Ulang Instans Anda di Panduan Pengguna Amazon Elastic Compute Cloud.

- Untuk detail API, lihat [RebootInstances](#) di Referensi AWS CLI Perintah.

JavaScript

SDK untuk JavaScript (v3)

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
import { RebootInstancesCommand } from "@aws-sdk/client-ec2";

import { client } from "../libs/client.js";

export const main = async () => {
  const command = new RebootInstancesCommand({
    InstanceIds: ["INSTANCE_ID"],
  });

  try {
    await client.send(command);
    console.log("Instance rebooted successfully.");
  } catch (err) {
    console.error(err);
  }
};
```

- Untuk detail API, lihat [RebootInstances](#) di Referensi AWS SDK for JavaScript API.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini me-reboot instance yang ditentukan.

```
Restart-EC2Instance -InstanceId i-12345678
```

- Untuk detail API, lihat [RebootInstances](#) di Referensi AWS Tools for PowerShell Cmdlet.

Python

SDK untuk Python (Boto3)

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
class AutoScaler:
    """
    Encapsulates Amazon EC2 Auto Scaling and EC2 management actions.
    """

    def __init__(
        self,
        resource_prefix,
        inst_type,
        ami_param,
        autoscaling_client,
        ec2_client,
        ssm_client,
        iam_client,
    ):
        """
        :param resource_prefix: The prefix for naming AWS resources that are
        created by this class.
        :param inst_type: The type of EC2 instance to create, such as t3.micro.
        :param ami_param: The Systems Manager parameter used to look up the AMI
        that is
                created.
        :param autoscaling_client: A Boto3 EC2 Auto Scaling client.
        :param ec2_client: A Boto3 EC2 client.
        :param ssm_client: A Boto3 Systems Manager client.
        :param iam_client: A Boto3 IAM client.
        """
```

```

self.inst_type = inst_type
self.ami_param = ami_param
self.autoscaling_client = autoscaling_client
self.ec2_client = ec2_client
self.ssm_client = ssm_client
self.iam_client = iam_client
self.launch_template_name = f"{resource_prefix}-template"
self.group_name = f"{resource_prefix}-group"
self.instance_policy_name = f"{resource_prefix}-pol"
self.instance_role_name = f"{resource_prefix}-role"
self.instance_profile_name = f"{resource_prefix}-prof"
self.bad_creds_policy_name = f"{resource_prefix}-bc-pol"
self.bad_creds_role_name = f"{resource_prefix}-bc-role"
self.bad_creds_profile_name = f"{resource_prefix}-bc-prof"
self.key_pair_name = f"{resource_prefix}-key-pair"

def replace_instance_profile(
    self, instance_id, new_instance_profile_name, profile_association_id
):
    """
    Replaces the profile associated with a running instance. After the
    profile is
    replaced, the instance is rebooted to ensure that it uses the new
    profile. When
    the instance is ready, Systems Manager is used to restart the Python web
    server.

    :param instance_id: The ID of the instance to update.
    :param new_instance_profile_name: The name of the new profile to
    associate with
                                the specified instance.
    :param profile_association_id: The ID of the existing profile association
    for the
                                instance.
    """
    try:
        self.ec2_client.replace_iam_instance_profile_association(
            IamInstanceProfile={"Name": new_instance_profile_name},
            AssociationId=profile_association_id,
        )
        log.info(
            "Replaced instance profile for association %s with profile %s.",
            profile_association_id,

```

```

        new_instance_profile_name,
    )
    time.sleep(5)
    inst_ready = False
    tries = 0
    while not inst_ready:
        if tries % 6 == 0:
            self.ec2_client.reboot_instances(InstanceIds=[instance_id])
            log.info(
                "Rebooting instance %s and waiting for it to be
ready.",
                instance_id,
            )
            tries += 1
            time.sleep(10)
            response = self.ssm_client.describe_instance_information()
            for info in response["InstanceInformationList"]:
                if info["InstanceId"] == instance_id:
                    inst_ready = True
            self.ssm_client.send_command(
                InstanceIds=[instance_id],
                DocumentName="AWS-RunShellScript",
                Parameters={"commands": ["cd / && sudo python3 server.py 80"]},
            )
            log.info("Restarted the Python web server on instance %s.",
instance_id)
        except ClientError as err:
            raise AutoScalerError(
                f"Couldn't replace instance profile for association
{profile_association_id}: {err}"
            )

```

- Untuk detail API, lihat [RebootInstances](#) di AWS SDK for Python (Boto3) Referensi API.

Rust

SDK untuk Rust

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
async fn reboot_instance(client: &Client, id: &str) -> Result<(), Error> {
    println!("Rebooting instance.");

    client.reboot_instances().instance_ids(id).send().await?;

    client
        .wait_until_instance_stopped()
        .instance_ids(id)
        .wait(Duration::from_secs(60))
        .await?;
    let wait_status_ok = client
        .wait_until_instance_status_ok()
        .instance_ids(id)
        .wait(Duration::from_secs(60))
        .await;


    match wait_status_ok {
        Ok(_) => println!("Rebooted instance {id}, it is started with status
OK."),
        Err(err) => return Err(err.into()),
    }

    Ok(())
}
```

- Untuk detail API, lihat [RebootInstances](#) referensi AWS SDK for Rust API.

SAP ABAP

SDK untuk SAP ABAP

 Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```

DATA lt_instance_ids TYPE /aws1/
cl_ec2instidstringlist_w=>tt_instanceidstringlist.
  APPEND NEW /aws1/cl_ec2instidstringlist_w( iv_value = iv_instance_id ) TO
  lt_instance_ids.

  "Perform dry run"
  TRY.
    " DryRun is set to true. This checks for the required permissions to
    reboot the instance without actually making the request. "
    lo_ec2->rebootinstances(
      it_instanceids = lt_instance_ids
      iv_dryrun = abap_true
    ).
  CATCH /aws1/cx_rt_service_generic INTO DATA(lo_exception).
    " If the error code returned is `DryRunOperation`, then you have the
    required permissions to reboot this instance. "
    IF lo_exception->av_err_code = 'DryRunOperation'.
      MESSAGE 'Dry run to reboot instance completed.' TYPE 'I'.
      " DryRun is set to false to make a reboot request. "
      lo_ec2->rebootinstances(
        it_instanceids = lt_instance_ids
        iv_dryrun = abap_false
      ).
      MESSAGE 'Instance rebooted.' TYPE 'I'.
    " If the error code returned is `UnauthorizedOperation`, then you don't
    have the required permissions to reboot this instance. "
    ELSEIF lo_exception->av_err_code = 'UnauthorizedOperation'.
      MESSAGE 'Dry run to reboot instance failed. User does not have
      permissions to reboot the instance.' TYPE 'E'.
    ELSE.
      DATA(lv_error) = |"{ lo_exception->av_err_code }" - { lo_exception-
      >av_err_msg }|.

```

```
    MESSAGE lv_error TYPE 'E'.  
  ENDIF.  
ENDTRY.
```

- Untuk detail API, lihat [RebootInstances](#) di AWS SDK untuk referensi SAP ABAP API.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **RegisterImage** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `RegisterImage`.

CLI

AWS CLI

Contoh 1: Untuk mendaftarkan AMI menggunakan file manifes

`register-image` Contoh berikut mendaftarkan AMI menggunakan file manifes yang ditentukan di Amazon S3.

```
aws ec2 register-image \  
  --name my-image \  
  --image-location my-s3-bucket/myimage/image.manifest.xml
```

Output:

```
{  
  "ImageId": "ami-1234567890EXAMPLE"  
}
```

Untuk informasi selengkapnya, lihat [Amazon Machine Image \(AMI\)](#) di Panduan Pengguna Amazon EC2.

Contoh 2: Untuk mendaftarkan AMI menggunakan snapshot perangkat root

`register-image` Contoh berikut mendaftarkan AMI menggunakan snapshot yang ditentukan dari volume root EBS sebagai perangkat. `/dev/xvda` Pemetaan perangkat blok juga menyertakan volume 100 GiB EBS kosong sebagai perangkat. `/dev/xvdf`

```
aws ec2 register-image \  
  --name my-image \  
  --root-device-name /dev/xvda \  
  --block-device-mappings DeviceName=/dev/  
xvda,Ebs={SnapshotId=snap-0db2cf683925d191f} DeviceName=/dev/  
xvdf,Ebs={VolumeSize=100}
```

Output:

```
{  
  "ImageId": "ami-1a2b3c4d5eEXAMPLE"  
}
```

Untuk informasi selengkapnya, lihat [Amazon Machine Image \(AMI\)](#) di Panduan Pengguna Amazon EC2.

- Untuk detail API, lihat [RegisterImage](#) di Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini mendaftarkan AMI menggunakan file manifes yang ditentukan di Amazon S3.

```
Register-EC2Image -ImageLocation my-s3-bucket/my-web-server-ami/  
image.manifest.xml -Name my-web-server-ami
```

- Untuk detail API, lihat [RegisterImage](#) di Referensi AWS Tools for PowerShell Cmdlet.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan `RejectVpcPeeringConnection` dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `RejectVpcPeeringConnection`.

CLI

AWS CLI

Untuk menolak koneksi peering VPC

Contoh ini menolak permintaan koneksi peering VPC yang ditentukan.

Perintah:

```
aws ec2 reject-vpc-peering-connection --vpc-peering-connection-id pcx-1a2b3c4d
```

Output:

```
{
  "Return": true
}
```

- Untuk detail API, lihat [RejectVpcPeeringConnection](#) di Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh di atas menolak permintaan `VpcPeering` permintaan id `pcx-01a2b3ce45fe67eb8`

```
Deny-EC2VpcPeeringConnection -VpcPeeringConnectionId pcx-01a2b3ce45fe67eb8
```

- Untuk detail API, lihat [RejectVpcPeeringConnection](#) di Referensi AWS Tools for PowerShell Cmdlet.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **ReleaseAddress** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `ReleaseAddress`.

Contoh tindakan adalah kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Anda dapat melihat tindakan ini dalam konteks dalam contoh kode berikut:

- [Memulai instans](#)

.NET

AWS SDK for .NET

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
/// <summary>
/// Release an Elastic IP address.
/// </summary>
/// <param name="allocationId">The allocation Id of the Elastic IP address.</
param>
/// <returns>A Boolean value indicating the success of the action.</returns>
public async Task<bool> ReleaseAddress(string allocationId)
{
    var request = new ReleaseAddressRequest
    {
        AllocationId = allocationId
    };

    var response = await _amazonEC2.ReleaseAddressAsync(request);
    return response.HttpStatusCode == HttpStatusCode.OK;
}
```

- Untuk detail API, lihat [ReleaseAddress](#) di Referensi AWS SDK for .NET API.

Bash

AWS CLI dengan skrip Bash

 Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
#####
# function ec2_release_address
#
# This function releases an Elastic IP address from an Amazon Elastic Compute
  Cloud (Amazon EC2) instance.
#
# Parameters:
#   -a allocation_id - The allocation ID of the Elastic IP address to
  release.
#
# Returns:
#   0 - If successful.
#   1 - If it fails.
#
#####
function ec2_release_address() {
    local allocation_id response

    # Function to display usage information
    function usage() {
        echo "function ec2_release_address"
        echo "Releases an Elastic IP address from an Amazon Elastic Compute Cloud
(Amazon EC2) instance."
        echo "  -a allocation_id - The allocation ID of the Elastic IP address to
release."
        echo ""
    }

    # Parse the command-line arguments
    while getopts "a:h" option; do
        case "${option}" in
            a) allocation_id="${OPTARG}" ;;

```

```

    h)
    usage
    return 0
    ;;
  \?)
    echo "Invalid parameter"
    usage
    return 1
    ;;
esac
done
export OPTIND=1

# Validate the input parameters
if [[ -z "$allocation_id" ]]; then
    errecho "ERROR: You must provide an allocation ID with the -a parameter."
    return 1
fi

response=$(aws ec2 release-address \
  --allocation-id "$allocation_id") || {
  aws_cli_error_log ${?}
  errecho "ERROR: AWS reports release-address operation failed."
  errecho "$response"
  return 1
}

return 0
}

```

Fungsi utilitas yang digunakan dalam contoh ini.

```

#####
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
#####
function errecho() {
  printf "%s\n" "$*" 1>&2
}

#####

```

```

# function aws_cli_error_log()
#
# This function is used to log the error messages from the AWS CLI.
#
# The function expects the following argument:
#     $1 - The error code returned by the AWS CLI.
#
# Returns:
#     0: - Success.
#
#####
function aws_cli_error_log() {
    local err_code=$1
    errecho "Error code : $err_code"
    if [ "$err_code" == 1 ]; then
        errecho " One or more S3 transfers failed."
    elif [ "$err_code" == 2 ]; then
        errecho " Command line failed to parse."
    elif [ "$err_code" == 130 ]; then
        errecho " Process received SIGINT."
    elif [ "$err_code" == 252 ]; then
        errecho " Command syntax invalid."
    elif [ "$err_code" == 253 ]; then
        errecho " The system environment or configuration was invalid."
    elif [ "$err_code" == 254 ]; then
        errecho " The service returned an error."
    elif [ "$err_code" == 255 ]; then
        errecho " 255 is a catch-all error."
    fi

    return 0
}

```

- Untuk detail API, lihat [ReleaseAddress](#) di Referensi AWS CLI Perintah.

C++

SDK untuk C++

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
Aws::EC2::EC2Client ec2(clientConfiguration);

Aws::EC2::Model::ReleaseAddressRequest request;
request.SetAllocationId(allocationID);

auto outcome = ec2.ReleaseAddress(request);
if (!outcome.IsSuccess()) {
    std::cerr << "Failed to release Elastic IP address " <<
        allocationID << ":" << outcome.GetError().GetMessage() <<
        std::endl;
}
else {
    std::cout << "Successfully released Elastic IP address " <<
        allocationID << std::endl;
}
```

- Untuk detail API, lihat [ReleaseAddress](#) di Referensi AWS SDK for C++ API.

CLI

AWS CLI

Guna menulis alamat IP Elastis untuk EC2-Classic

Contoh ini menulis alamat IP Elastis untuk digunakan dengan instans di EC2-Classic. Jika perintah berhasil, tidak ada output yang akan ditampilkan.

Perintah:

```
aws ec2 release-address --public-ip 198.51.100.0
```

Guna merilis alamat IP Elastis untuk EC2-VPC

Contoh ini merilis alamat IP Elastis untuk digunakan dengan instans di VPC. Jika perintah berhasil, tidak ada output yang akan ditampilkan.

Perintah:

```
aws ec2 release-address --allocation-id eipalloc-64d5890a
```

- Untuk detail API, lihat [ReleaseAddress](#) di Referensi AWS CLI Perintah.

Java

SDK untuk Java 2.x

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
public static void releaseEC2Address(Ec2Client ec2, String allocId) {
    try {
        ReleaseAddressRequest request = ReleaseAddressRequest.builder()
            .allocationId(allocId)
            .build();

        ec2.releaseAddress(request);
        System.out.println("Successfully released Elastic IP address " +
allocId);
    } catch (Ec2Exception e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
}
```

- Untuk detail API, lihat [ReleaseAddress](#) di Referensi AWS SDK for Java 2.x API.

JavaScript

SDK untuk JavaScript (v3)

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
import { ReleaseAddressCommand } from "@aws-sdk/client-ec2";

import { client } from "../libs/client.js";

export const main = async () => {
  const command = new ReleaseAddressCommand({
    // You can also use PublicIp, but that is for EC2 classic which is being
    // retired.
    AllocationId: "ALLOCATION_ID",
  });

  try {
    await client.send(command);
    console.log("Successfully released address.");
  } catch (err) {
    console.error(err);
  }
};
```

- Untuk detail API, lihat [ReleaseAddress](#) di Referensi AWS SDK for JavaScript API.

Kotlin

SDK untuk Kotlin

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).


```
suspend fun releaseEC2AddressSc(allocId: String?) {
    val request =
        ReleaseAddressRequest {
            allocationId = allocId
        }

    Ec2Client { region = "us-west-2" }.use { ec2 ->
        ec2.releaseAddress(request)
        println("Successfully released Elastic IP address $allocId")
    }
}
```

- Untuk detail API, lihat [ReleaseAddress](#) di AWS SDK untuk referensi API Kotlin.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini merilis alamat IP Elastis yang ditentukan untuk instance di VPC.

```
Remove-EC2Address -AllocationId eipalloc-12345678 -Force
```

Contoh 2: Contoh ini merilis alamat IP Elastis yang ditentukan untuk instance di EC2-Classic.

```
Remove-EC2Address -PublicIp 198.51.100.2 -Force
```

- Untuk detail API, lihat [ReleaseAddress](#) di Referensi AWS Tools for PowerShell Cmdlet.

Python

SDK untuk Python (Boto3)

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
class ElasticIpWrapper:
```

```
"""Encapsulates Amazon Elastic Compute Cloud (Amazon EC2) Elastic IP address
actions."""

def __init__(self, ec2_resource, elastic_ip=None):
    """
    :param ec2_resource: A Boto3 Amazon EC2 resource. This high-level
resource
                           is used to create additional high-level objects
                           that wrap low-level Amazon EC2 service actions.
    :param elastic_ip: A Boto3 VpcAddress object. This is a high-level object
that
                           wraps Elastic IP actions.
    """
    self.ec2_resource = ec2_resource
    self.elastic_ip = elastic_ip

    @classmethod
    def from_resource(cls):
        ec2_resource = boto3.resource("ec2")
        return cls(ec2_resource)

    def release(self):
        """
        Releases an Elastic IP address. After the Elastic IP address is released,
it can no longer be used.
        """
        if self.elastic_ip is None:
            logger.info("No Elastic IP to release.")
            return

        try:
            self.elastic_ip.release()
        except ClientError as err:
            logger.error(
                "Couldn't release Elastic IP address %s. Here's why: %s: %s",
                self.elastic_ip.allocation_id,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
```

- Untuk detail API, lihat [ReleaseAddress](#) di AWS SDK for Python (Boto3) Referensi API.

Ruby

SDK untuk Ruby

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
# Releases an Elastic IP address from an
# Amazon Elastic Compute Cloud (Amazon EC2) instance.
#
# Prerequisites:
#
# - An Amazon EC2 instance with an associated Elastic IP address.
#
# @param ec2_client [Aws::EC2::Client] An initialized EC2 client.
# @param allocation_id [String] The ID of the allocation corresponding to
#   the Elastic IP address.
# @return [Boolean] true if the Elastic IP address was released;
#   otherwise, false.
# @example
#   exit 1 unless elastic_ip_address_released?(
#     Aws::EC2::Client.new(region: 'us-west-2'),
#     'eipalloc-04452e528a66279EX'
#   )
def elastic_ip_address_released?(ec2_client, allocation_id)
  ec2_client.release_address(allocation_id: allocation_id)
  return true
rescue StandardError => e
  puts("Error releasing Elastic IP address: #{e.message}")
  return false
end
```

- Untuk detail API, lihat [ReleaseAddress](#) di Referensi AWS SDK for Ruby API.

SAP ABAP

SDK untuk SAP ABAP

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
TRY.  
    lo_ec2->releaseaddress( iv_allocationid = iv_allocation_id ).  
    MESSAGE 'Elastic IP address released.' TYPE 'I'.  
    CATCH /aws1/cx_rt_service_generic INTO DATA(lo_exception).  
    DATA(lv_error) = |"{ lo_exception->av_err_code }" - { lo_exception->  
>av_err_msg }|.  
    MESSAGE lv_error TYPE 'E'.  
ENDTRY.
```

- Untuk detail API, lihat [ReleaseAddress](#) di AWS SDK untuk referensi SAP ABAP API.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **ReleaseHosts** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `ReleaseHosts`.

CLI

AWS CLI

Untuk melepaskan host Khusus dari akun Anda

Untuk melepaskan host Khusus dari akun Anda. Instance yang ada di host harus dihentikan atau dihentikan sebelum host dapat dirilis.

Perintah:

```
aws ec2 release-hosts --host-id=h-0029d6e3cacf1b3da
```

Output:

```
{
  "Successful": [
    "h-0029d6e3cacf1b3da"
  ],
  "Unsuccessful": []
}
```

- Untuk detail API, lihat [ReleaseHosts](#) di Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini merilis ID host yang diberikan h-0badafd1dcb2f3456

```
Remove-EC2Host -HostId h-0badafd1dcb2f3456
```

Output:

```
Confirm
Are you sure you want to perform this action?
Performing the operation "Remove-EC2Host (ReleaseHosts)" on target
"h-0badafd1dcb2f3456".
[Y] Yes [A] Yes to All [N] No [L] No to All [S] Suspend [?] Help (default is
"Y"): Y

Successful                Unsuccessful
-----                -
{h-0badafd1dcb2f3456} {}
```

- Untuk detail API, lihat [ReleaseHosts](#) di Referensi AWS Tools for PowerShell Cmdlet.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan `ReplaceIamInstanceProfileAssociation` dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `ReplaceIamInstanceProfileAssociation`.

Contoh tindakan adalah kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Anda dapat melihat tindakan ini dalam konteks dalam contoh kode berikut:

- [Membangun dan mengelola layanan yang tangguh](#)

.NET

AWS SDK for .NET

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
/// <summary>
/// Replace the profile associated with a running instance. After the profile
is replaced, the instance
/// is rebooted to ensure that it uses the new profile. When the instance is
ready, Systems Manager is
/// used to restart the Python web server.
/// </summary>
/// <param name="instanceId">The Id of the instance to update.</param>
/// <param name="credsProfileName">The name of the new profile to associate
with the specified instance.</param>
/// <param name="associationId">The Id of the existing profile association
for the instance.</param>
/// <returns>Async task.</returns>
public async Task ReplaceInstanceProfile(string instanceId, string
credsProfileName, string associationId)
{
    await _amazonEc2.ReplaceIamInstanceProfileAssociationAsync(
        new ReplaceIamInstanceProfileAssociationRequest()
        {
            AssociationId = associationId,
```

```
        IamInstanceProfile = new IamInstanceProfileSpecification()
        {
            Name = credsProfileName
        }
    });
// Allow time before resetting.
Thread.Sleep(25000);
var instanceReady = false;
var retries = 5;
while (retries-- > 0 && !instanceReady)
{
    await _amazonEc2.RebootInstancesAsync(
        new RebootInstancesRequest(new List<string>() { instanceId }));
    Thread.Sleep(10000);

    var instancesPaginator =
        _amazonSsm.Paginators.DescribeInstanceInformation(
            new DescribeInstanceInformationRequest());
    // Get the entire list using the paginator.
    await foreach (var instance in
instancesPaginator.InstanceInformationList)
    {
        instanceReady = instance.InstanceId == instanceId;
        if (instanceReady)
        {
            break;
        }
    }
}
Console.WriteLine($"Sending restart command to instance {instanceId}");
await _amazonSsm.SendCommandAsync(
    new SendCommandRequest()
    {
        InstanceIds = new List<string>() { instanceId },
        DocumentName = "AWS-RunShellScript",
        Parameters = new Dictionary<string, List<string>>()
        {
            {"commands", new List<string>() { "cd / && sudo python3
server.py 80" }}
        }
    });
Console.WriteLine($"Restarted the web server on instance {instanceId}");
}
```

- Untuk detail API, lihat [ReplaceIamInstanceProfileAssociation](#) di Referensi AWS SDK for .NET API.

CLI

AWS CLI

Guna mengganti profil instans IAM untuk instans

Contoh ini menggantikan profil instans IAM yang direpresentasikan oleh kaitan `iip-assoc-060bae234aac2e7fa` dengan profil instans IAM bernama `AdminRole`.

```
aws ec2 replace-iam-instance-profile-association \  
  --iam-instance-profile Name=AdminRole \  
  --association-id iip-assoc-060bae234aac2e7fa
```

Output:

```
{  
  "IamInstanceProfileAssociation": {  
    "InstanceId": "i-087711ddaf98f9489",  
    "State": "associating",  
    "AssociationId": "iip-assoc-0b215292fab192820",  
    "IamInstanceProfile": {  
      "Id": "AIPAJLNLDX3AMYZNWYYAY",  
      "Arn": "arn:aws:iam::123456789012:instance-profile/AdminRole"  
    }  
  }  
}
```

- Untuk detail API, lihat [ReplaceIamInstanceProfileAssociation](#) di Referensi AWS CLI Perintah.

JavaScript

SDK untuk JavaScript (v3)

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
await retry({ intervalInMs: 1000, maxRetries: 30 }, () =>
  ec2Client.send(
    new ReplaceIamInstanceProfileAssociationCommand({
      AssociationId: state.instanceProfileAssociationId,
      IamInstanceProfile: { Name: NAMES.ssmOnlyInstanceProfileName },
    }),
  ),
);
```

- Untuk detail API, lihat [ReplacelamInstanceProfileAssociation](#) di Referensi AWS SDK for JavaScript API.

Python

SDK untuk Python (Boto3)

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturan dan menjalankannya di [Repositori Contoh Kode AWS](#).

Contoh ini mengganti profil instans dari instans yang sedang berjalan, menyalakan ulang instans, dan mengirimkan perintah ke instans tersebut setelah dimulai.

```
class AutoScaler:
    """
    Encapsulates Amazon EC2 Auto Scaling and EC2 management actions.
    """
```

```
def __init__(
    self,
    resource_prefix,
    inst_type,
    ami_param,
    autoscaling_client,
    ec2_client,
    ssm_client,
    iam_client,
):
    """
    :param resource_prefix: The prefix for naming AWS resources that are
    created by this class.
    :param inst_type: The type of EC2 instance to create, such as t3.micro.
    :param ami_param: The Systems Manager parameter used to look up the AMI
    that is
        created.
    :param autoscaling_client: A Boto3 EC2 Auto Scaling client.
    :param ec2_client: A Boto3 EC2 client.
    :param ssm_client: A Boto3 Systems Manager client.
    :param iam_client: A Boto3 IAM client.
    """
    self.inst_type = inst_type
    self.ami_param = ami_param
    self.autoscaling_client = autoscaling_client
    self.ec2_client = ec2_client
    self.ssm_client = ssm_client
    self.iam_client = iam_client
    self.launch_template_name = f"{resource_prefix}-template"
    self.group_name = f"{resource_prefix}-group"
    self.instance_policy_name = f"{resource_prefix}-pol"
    self.instance_role_name = f"{resource_prefix}-role"
    self.instance_profile_name = f"{resource_prefix}-prof"
    self.bad_creds_policy_name = f"{resource_prefix}-bc-pol"
    self.bad_creds_role_name = f"{resource_prefix}-bc-role"
    self.bad_creds_profile_name = f"{resource_prefix}-bc-prof"
    self.key_pair_name = f"{resource_prefix}-key-pair"

def replace_instance_profile(
    self, instance_id, new_instance_profile_name, profile_association_id
):
    """
```

Replaces the profile associated with a running instance. After the profile is replaced, the instance is rebooted to ensure that it uses the new profile. When the instance is ready, Systems Manager is used to restart the Python web server.

:param instance_id: The ID of the instance to update.
 :param new_instance_profile_name: The name of the new profile to associate with the specified instance.
 :param profile_association_id: The ID of the existing profile association for the instance.

```

"""
try:
    self.ec2_client.replace_iam_instance_profile_association(
        IamInstanceProfile={"Name": new_instance_profile_name},
        AssociationId=profile_association_id,
    )
    log.info(
        "Replaced instance profile for association %s with profile %s.",
        profile_association_id,
        new_instance_profile_name,
    )
    time.sleep(5)
    inst_ready = False
    tries = 0
    while not inst_ready:
        if tries % 6 == 0:
            self.ec2_client.reboot_instances(InstanceIds=[instance_id])
            log.info(
                "Rebooting instance %s and waiting for it to be
ready.",
                instance_id,
            )
            tries += 1
            time.sleep(10)
            response = self.ssm_client.describe_instance_information()
            for info in response["InstanceInformationList"]:
                if info["InstanceId"] == instance_id:
                    inst_ready = True
    self.ssm_client.send_command(
        InstanceIds=[instance_id],

```

```

        DocumentName="AWS-RunShellScript",
        Parameters={"commands": ["cd / && sudo python3 server.py 80"]},
    )
    log.info("Restarted the Python web server on instance %s.",
instance_id)
    except ClientError as err:
        raise AutoScalerError(
            f"Couldn't replace instance profile for association
{profile_association_id}: {err}"
        )

```

- Untuk detail API, lihat [ReplacelamInstanceProfileAssociation](#) di AWS SDK for Python (Boto3) Referensi API.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **ReplaceNetworkAclAssociation** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `ReplaceNetworkAclAssociation`.

CLI

AWS CLI

Untuk mengganti ACL jaringan yang terkait dengan subnet

Contoh ini mengaitkan ACL jaringan tertentu dengan subnet untuk asosiasi ACL jaringan tertentu.

Perintah:

```
aws ec2 replace-network-acl-association --association-id aclassoc-e5b95c8c --
network-acl-id acl-5fb85d36
```

Output:

```
{
```

```
"NewAssociationId": "aclassoc-3999875b"  
}
```

- Untuk detail API, lihat [ReplaceNetworkAclAssociation](#) di Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini mengaitkan ACL jaringan tertentu dengan subnet untuk asosiasi ACL jaringan tertentu.

```
Set-EC2NetworkAclAssociation -NetworkAclId acl-12345678 -AssociationId  
aclassoc-1a2b3c4d
```

Output:

```
aclassoc-87654321
```

- Untuk detail API, lihat [ReplaceNetworkAclAssociation](#) di Referensi AWS Tools for PowerShell Cmdlet.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **ReplaceNetworkAclEntry** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `ReplaceNetworkAclEntry`.

CLI

AWS CLI

Untuk mengganti entri ACL jaringan

Contoh ini menggantikan entri untuk ACL jaringan tertentu. Aturan baru 100 memungkinkan masuknya lalu lintas dari 203.0.113.12/24 pada port UDP 53 (DNS) ke subnet terkait.

Perintah:

```
aws ec2 replace-network-acl-entry --network-acl-id acl-5fb85d36 --ingress --rule-number 100 --protocol udp --port-range From=53,To=53 --cidr-block 203.0.113.12/24 --rule-action allow
```

- Untuk detail API, lihat [ReplaceNetworkAclEntry](#) di Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini menggantikan entri yang ditentukan untuk ACL jaringan tertentu. Aturan baru memungkinkan lalu lintas masuk dari alamat yang ditentukan ke subnet terkait.

```
Set-EC2NetworkAclEntry -NetworkAclId acl-12345678 -Egress $false -RuleNumber 100 -Protocol 17 -PortRange_From 53 -PortRange_To 53 -CidrBlock 203.0.113.12/24 -RuleAction allow
```

- Untuk detail API, lihat [ReplaceNetworkAclEntry](#) di Referensi AWS Tools for PowerShell Cmdlet.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **ReplaceRoute** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `ReplaceRoute`.

CLI

AWS CLI

Untuk mengganti rute

Contoh ini menggantikan rute yang ditentukan dalam tabel rute yang ditentukan. Rute baru cocok dengan CIDR yang ditentukan dan mengirimkan lalu lintas ke gateway pribadi virtual yang ditentukan. Jika perintah berhasil, tidak ada output yang akan ditampilkan.

Perintah:

```
aws ec2 replace-route --route-table-id rtb-22574640 --destination-cidr-block 10.0.0.0/16 --gateway-id vgw-9a4cacf3
```

- Untuk detail API, lihat [ReplaceRoute](#) di Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini menggantikan rute yang ditentukan untuk tabel rute yang ditentukan. Rute baru mengirimkan lalu lintas yang ditentukan ke gateway pribadi virtual yang ditentukan.

```
Set-EC2Route -RouteTableId rtb-1a2b3c4d -DestinationCidrBlock 10.0.0.0/24 - GatewayId vgw-1a2b3c4d
```

- Untuk detail API, lihat [ReplaceRoute](#) di Referensi AWS Tools for PowerShell Cmdlet.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **ReplaceRouteTableAssociation** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `ReplaceRouteTableAssociation`.

CLI

AWS CLI

Untuk mengganti tabel rute yang terkait dengan subnet

Contoh ini mengaitkan tabel rute yang ditentukan dengan subnet untuk asosiasi tabel rute yang ditentukan.

Perintah:

```
aws ec2 replace-route-table-association --association-id rtbassoc-781d0d1a -- route-table-id rtb-22574640
```

Output:

```
{
  "NewAssociationId": "rtbassoc-3a1f0f58"
}
```

- Untuk detail API, lihat [ReplaceRouteTableAssociation](#) di Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini mengaitkan tabel rute yang ditentukan dengan subnet untuk asosiasi tabel rute yang ditentukan.

```
Set-EC2RouteTableAssociation -RouteTableId rtb-1a2b3c4d -AssociationId
rtbassoc-12345678
```

Output:

```
rtbassoc-87654321
```

- Untuk detail API, lihat [ReplaceRouteTableAssociation](#) di Referensi AWS Tools for PowerShell Cmdlet.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **ReportInstanceStatus** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `ReportInstanceStatus`.

CLI

AWS CLI

Untuk melaporkan umpan balik status untuk sebuah instance

Perintah contoh ini melaporkan umpan balik status untuk instance yang ditentukan.

Perintah:


```
aws ec2 report-instance-status --instances i-1234567890abcdef0 --status impaired
--reason-codes unresponsive
```

- Untuk detail API, lihat [ReportInstanceStatus](#) di Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini melaporkan umpan balik status untuk contoh yang ditentukan.

```
Send-EC2InstanceStatus -Instance i-12345678 -Status impaired -ReasonCode
unresponsive
```

- Untuk detail API, lihat [ReportInstanceStatus](#) di Referensi AWS Tools for PowerShell Cmdlet.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **RequestSpotFleet** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `RequestSpotFleet`.

CLI

AWS CLI

Untuk meminta armada Spot di subnet dengan harga terendah

Perintah contoh ini membuat permintaan armada Spot dengan dua spesifikasi peluncuran yang hanya berbeda dengan subnet. Armada Spot meluncurkan instans di subnet yang ditentukan dengan harga terendah. Jika instance diluncurkan dalam VPC default, mereka menerima alamat IP publik secara default. Jika instans diluncurkan dalam VPC nondefault, mereka tidak menerima alamat IP publik secara default.

Perhatikan bahwa Anda tidak dapat menentukan subnet yang berbeda dari Availability Zone yang sama dalam permintaan armada Spot.

Perintah:

```
aws ec2 request-spot-fleet --spot-fleet-request-config file://config.json
```

config.json:

```
{
  "SpotPrice": "0.04",
  "TargetCapacity": 2,
  "IamFleetRole": "arn:aws:iam::123456789012:role/my-spot-fleet-role",
  "LaunchSpecifications": [
    {
      "ImageId": "ami-1a2b3c4d",
      "KeyName": "my-key-pair",
      "SecurityGroups": [
        {
          "GroupId": "sg-1a2b3c4d"
        }
      ],
      "InstanceType": "m3.medium",
      "SubnetId": "subnet-1a2b3c4d, subnet-3c4d5e6f",
      "IamInstanceProfile": {
        "Arn": "arn:aws:iam::123456789012:instance-profile/my-iam-role"
      }
    }
  ]
}
```

Output:

```
{
  "SpotFleetRequestId": "sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE"
}
```

Untuk meminta armada Spot di Availability Zone dengan harga terendah

Perintah contoh ini membuat permintaan armada Spot dengan dua spesifikasi peluncuran yang hanya berbeda dengan Availability Zone. Armada Spot meluncurkan instans di Availability Zone yang ditentukan dengan harga terendah. Jika akun Anda hanya mendukung EC2-VPC, Amazon EC2 meluncurkan instans Spot di subnet default Availability Zone. Jika akun Anda mendukung EC2-Classic, Amazon EC2 meluncurkan instans di EC2-Classic di Availability Zone.

Perintah:

```
aws ec2 request-spot-fleet --spot-fleet-request-config file://config.json
```

config.json:

```
{
  "SpotPrice": "0.04",
  "TargetCapacity": 2,
  "IamFleetRole": "arn:aws:iam::123456789012:role/my-spot-fleet-role",
  "LaunchSpecifications": [
    {
      "ImageId": "ami-1a2b3c4d",
      "KeyName": "my-key-pair",
      "SecurityGroups": [
        {
          "GroupId": "sg-1a2b3c4d"
        }
      ],
      "InstanceType": "m3.medium",
      "Placement": {
        "AvailabilityZone": "us-west-2a, us-west-2b"
      },
      "IamInstanceProfile": {
        "Arn": "arn:aws:iam::123456789012:instance-profile/my-iam-role"
      }
    }
  ]
}
```

Untuk meluncurkan instans Spot di subnet dan menentukannya alamat IP publik

Perintah contoh ini menetapkan alamat publik ke instance yang diluncurkan di VPC nondefault. Perhatikan bahwa ketika Anda menentukan antarmuka jaringan, Anda harus menyertakan ID subnet dan ID grup keamanan menggunakan antarmuka jaringan.

Perintah:

```
aws ec2 request-spot-fleet --spot-fleet-request-config file://config.json
```

config.json:

```
{
  "SpotPrice": "0.04",
  "TargetCapacity": 2,
  "IamFleetRole": "arn:aws:iam::123456789012:role/my-spot-fleet-role",
  "LaunchSpecifications": [
    {
      "ImageId": "ami-1a2b3c4d",
      "KeyName": "my-key-pair",
      "InstanceType": "m3.medium",
      "NetworkInterfaces": [
        {
          "DeviceIndex": 0,
          "SubnetId": "subnet-1a2b3c4d",
          "Groups": [ "sg-1a2b3c4d" ],
          "AssociatePublicIpAddress": true
        }
      ],
      "IamInstanceProfile": {
        "Arn": "arn:aws:iam::880185128111:instance-profile/my-iam-role"
      }
    }
  ]
}
```

Untuk meminta armada Spot menggunakan strategi alokasi yang beragam

Perintah contoh ini membuat permintaan armada Spot yang meluncurkan 30 instance menggunakan strategi alokasi yang beragam. Spesifikasi peluncuran berbeda menurut jenis instans. Armada Spot mendistribusikan instans di seluruh spesifikasi peluncuran sehingga ada 10 instance dari setiap jenis.

Perintah:

```
aws ec2 request-spot-fleet --spot-fleet-request-config file://config.json
```

config.json:

```
{
  "SpotPrice": "0.70",
  "TargetCapacity": 30,
  "AllocationStrategy": "diversified",
  "IamFleetRole": "arn:aws:iam::123456789012:role/my-spot-fleet-role",
```

```
"LaunchSpecifications": [  
  {  
    "ImageId": "ami-1a2b3c4d",  
    "InstanceType": "c4.2xlarge",  
    "SubnetId": "subnet-1a2b3c4d"  
  },  
  {  
    "ImageId": "ami-1a2b3c4d",  
    "InstanceType": "m3.2xlarge",  
    "SubnetId": "subnet-1a2b3c4d"  
  },  
  {  
    "ImageId": "ami-1a2b3c4d",  
    "InstanceType": "r3.2xlarge",  
    "SubnetId": "subnet-1a2b3c4d"  
  }  
]  
}
```

Untuk informasi selengkapnya, lihat [Permintaan Armada Spot](#) di Panduan Pengguna Amazon Elastic Compute Cloud.

- Untuk detail API, lihat [RequestSpotFleet](#) di Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini membuat permintaan armada Spot di Availability Zone dengan harga terendah untuk jenis instans yang ditentukan. Jika akun Anda hanya mendukung EC2-VPC, armada Spot akan meluncurkan instans di Availability Zone dengan harga terendah yang memiliki subnet default. Jika akun Anda mendukung EC2-Classic, armada Spot meluncurkan instans dalam EC2-Classic di Availability Zone dengan harga terendah. Perhatikan bahwa harga yang Anda bayar tidak akan melebihi harga Spot yang ditentukan untuk permintaan tersebut.

```
$sg = New-Object Amazon.EC2.Model.GroupIdentifier  
$sg.GroupId = "sg-12345678"  
$lc = New-Object Amazon.EC2.Model.SpotFleetLaunchSpecification  
$lc.ImageId = "ami-12345678"  
$lc.InstanceType = "m3.medium"  
$lc.SecurityGroups.Add($sg)
```

```
Request-EC2SpotFleet -SpotFleetRequestConfig_SpotPrice 0.04 `
-SpotFleetRequestConfig_TargetCapacity 2 `
-SpotFleetRequestConfig_IamFleetRole arn:aws:iam::123456789012:role/my-spot-
fleet-role `
-SpotFleetRequestConfig_LaunchSpecification $1c
```

- Untuk detail API, lihat [RequestSpotFleet](#) di Referensi AWS Tools for PowerShell Cmdlet.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **RequestSpotInstances** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `RequestSpotInstances`.

CLI

AWS CLI

Untuk meminta Instans Spot

Perintah contoh ini membuat permintaan Instans Spot satu kali untuk lima instance di Availability Zone yang ditentukan. Jika akun Anda hanya mendukung EC2-VPC, Amazon EC2 meluncurkan instans di subnet default dari Availability Zone yang ditentukan. Jika akun Anda mendukung EC2-Classic, Amazon EC2 meluncurkan instans di EC2-Classic di Availability Zone yang ditentukan.

Perintah:

```
aws ec2 request-spot-instances --spot-price "0.03" --instance-count 5 --type
"one-time" --launch-specification file://specification.json
```

Spesifikasi.json:

```
{
  "ImageId": "ami-1a2b3c4d",
  "KeyName": "my-key-pair",
  "SecurityGroupIds": [ "sg-1a2b3c4d" ],
  "InstanceType": "m3.medium",
  "Placement": {
```

```

    "AvailabilityZone": "us-west-2a"
  },
  "IamInstanceProfile": {
    "Arn": "arn:aws:iam::123456789012:instance-profile/my-iam-role"
  }
}

```

Output:

```

{
  "SpotInstanceRequests": [
    {
      "Status": {
        "UpdateTime": "2014-03-25T20:54:21.000Z",
        "Code": "pending-evaluation",
        "Message": "Your Spot request has been submitted for review, and is
pending evaluation."
      },
      "ProductDescription": "Linux/UNIX",
      "SpotInstanceRequestId": "sir-df6f405d",
      "State": "open",
      "LaunchSpecification": {
        "Placement": {
          "AvailabilityZone": "us-west-2a"
        },
        "ImageId": "ami-1a2b3c4d",
        "KeyName": "my-key-pair",
        "SecurityGroups": [
          {
            "GroupName": "my-security-group",
            "GroupId": "sg-1a2b3c4d"
          }
        ],
        "Monitoring": {
          "Enabled": false
        },
        "IamInstanceProfile": {
          "Arn": "arn:aws:iam::123456789012:instance-profile/my-iam-role"
        },
        "InstanceType": "m3.medium"
      },
      "Type": "one-time",
      "CreateTime": "2014-03-25T20:54:20.000Z",

```

```
        "SpotPrice": "0.050000"  
    },  
    ...  
]  
}
```

Perintah contoh ini membuat permintaan Instans Spot satu kali untuk lima instance di subnet yang ditentukan. Amazon EC2 meluncurkan instans di subnet yang ditentukan. Jika VPC adalah VPC nondefault, instance tidak menerima alamat IP publik secara default.

Perintah:

```
aws ec2 request-spot-instances --spot-price "0.050" --instance-count 5 --type  
"one-time" --launch-specification file://specification.json
```

Spesifikasi.json:

```
{  
  "ImageId": "ami-1a2b3c4d",  
  "SecurityGroupIds": [ "sg-1a2b3c4d" ],  
  "InstanceType": "m3.medium",  
  "SubnetId": "subnet-1a2b3c4d",  
  "IamInstanceProfile": {  
    "Arn": "arn:aws:iam::123456789012:instance-profile/my-iam-role"  
  }  
}
```

Output:

```
{  
  "SpotInstanceRequests": [  
    {  
      "Status": {  
        "UpdateTime": "2014-03-25T22:21:58.000Z",  
        "Code": "pending-evaluation",  
        "Message": "Your Spot request has been submitted for review, and is  
pending evaluation."  
      },  
      "ProductDescription": "Linux/UNIX",  
      "SpotInstanceRequestId": "sir-df6f405d",  
      "State": "open",  
    }  
  ]  
}
```



```

    "LaunchSpecification": {
      "Placement": {
        "AvailabilityZone": "us-west-2a"
      }
      "ImageId": "ami-1a2b3c4d"
      "SecurityGroups": [
        {
          "GroupName": "my-security-group",
          "GroupID": "sg-1a2b3c4d"
        }
      ]
      "SubnetId": "subnet-1a2b3c4d",
      "Monitoring": {
        "Enabled": false
      },
      "IamInstanceProfile": {
        "Arn": "arn:aws:iam::123456789012:instance-profile/my-iam-role"
      },
      "InstanceType": "m3.medium",
    },
    "Type": "one-time",
    "CreateTime": "2014-03-25T22:21:58.000Z",
    "SpotPrice": "0.050000"
  },
  ...
]
}

```

Contoh ini menetapkan alamat IP publik ke Instans Spot yang Anda luncurkan di VPC nondefault. Perhatikan bahwa ketika Anda menentukan antarmuka jaringan, Anda harus menyertakan ID subnet dan ID grup keamanan menggunakan antarmuka jaringan.

Perintah:

```
aws ec2 request-spot-instances --spot-price "0.050" --instance-count 1 --type
"one-time" --launch-specification file://specification.json
```

Spesifikasi.json:

```
{
  "ImageId": "ami-1a2b3c4d",
  "KeyName": "my-key-pair",

```

```

"InstanceType": "m3.medium",
"NetworkInterfaces": [
  {
    "DeviceIndex": 0,
    "SubnetId": "subnet-1a2b3c4d",
    "Groups": [ "sg-1a2b3c4d" ],
    "AssociatePublicIpAddress": true
  }
],
"IamInstanceProfile": {
  "Arn": "arn:aws:iam::123456789012:instance-profile/my-iam-role"
}
}

```

- Untuk detail API, lihat [RequestSpotInstances](#) di Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini meminta instance Spot satu kali di subnet yang ditentukan. Perhatikan bahwa grup keamanan harus dibuat untuk VPC yang berisi subnet yang ditentukan, dan harus ditentukan oleh ID menggunakan antarmuka jaringan. Saat Anda menentukan antarmuka jaringan, Anda harus menyertakan ID subnet menggunakan antarmuka jaringan.

```

$n = New-Object Amazon.EC2.Model.InstanceNetworkInterfaceSpecification
$n.DeviceIndex = 0
$n.SubnetId = "subnet-12345678"
$n.Groups.Add("sg-12345678")
Request-EC2SpotInstance -InstanceCount 1 -SpotPrice 0.050 -Type one-time `
-IamInstanceProfile_Arn arn:aws:iam::123456789012:instance-profile/my-iam-role `
-LaunchSpecification_ImageId ami-12345678 `
-LaunchSpecification_InstanceType m3.medium `
-LaunchSpecification_NetworkInterface $n

```

Output:

```

ActualBlockHourlyPrice      :
AvailabilityZoneGroup       :
BlockDurationMinutes        : 0
CreateTime                  : 12/26/2015 7:44:10 AM

```

```
Fault :  
InstanceId :  
LaunchedAvailabilityZone :  
LaunchGroup :  
LaunchSpecification : Amazon.EC2.Model.LaunchSpecification  
ProductDescription : Linux/UNIX  
SpotInstanceRequestId : sir-12345678  
SpotPrice : 0.050000  
State : open  
Status : Amazon.EC2.Model.SpotInstanceStatus  
Tags : {}  
Type : one-time
```

- Untuk detail API, lihat [RequestSpotInstances](#) di Referensi AWS Tools for PowerShell Cmdlet.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **ResetImageAttribute** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `ResetImageAttribute`.

CLI

AWS CLI

Untuk mengatur ulang atribut `launchPermission`

Contoh ini me-reset `launchPermission` atribut untuk AMI yang ditentukan ke nilai defaultnya. Secara default, AMI bersifat pribadi. Jika perintah berhasil, tidak ada output yang akan ditampilkan.

Perintah:

```
aws ec2 reset-image-attribute --image-id ami-5731123e --attribute  
launchPermission
```

- Untuk detail API, lihat [ResetImageAttribute](#) di Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini me-reset atribut 'launchPermission' ke nilai defaultnya. Secara default, AMI bersifat pribadi.

```
Reset-EC2ImageAttribute -ImageId ami-12345678 -Attribute launchPermission
```

- Untuk detail API, lihat [ResetImageAttribute](#) di Referensi AWS Tools for PowerShell Cmdlet.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **ResetInstanceAttribute** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `ResetInstanceAttribute`.

CLI

AWS CLI

Untuk mengatur ulang `sourceDestCheck` atribut

Contoh ini me-reset `sourceDestCheck` atribut dari instance tertentu. Instans harus dalam VPC. Jika perintah berhasil, tidak ada output yang akan ditampilkan.

Perintah:

```
aws ec2 reset-instance-attribute --instance-id i-1234567890abcdef0 --attribute sourceDestCheck
```

Untuk mengatur ulang atribut kernel

Contoh ini me-reset `kernel` atribut dari instance tertentu. Instans harus berada dalam status `stopped`. Jika perintah berhasil, tidak ada output yang akan ditampilkan.

Perintah:

```
aws ec2 reset-instance-attribute --instance-id i-1234567890abcdef0 --attribute
kernel
```

Untuk mengatur ulang atribut ramdisk

Contoh ini me-reset ramdisk atribut dari instance tertentu. Instans harus berada dalam status stopped. Jika perintah berhasil, tidak ada output yang akan ditampilkan.

Perintah:

```
aws ec2 reset-instance-attribute --instance-id i-1234567890abcdef0 --attribute
ramdisk
```

- Untuk detail API, lihat [ResetInstanceAttribute](#) di Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini me-reset atribut sriovNetSupport " untuk instance tertentu.

```
Reset-EC2InstanceAttribute -InstanceId i-12345678 -Attribute sriovNetSupport
```

Contoh 2: Contoh ini me-reset atribut 'EBSOptimized' untuk instance tertentu.

```
Reset-EC2InstanceAttribute -InstanceId i-12345678 -Attribute ebsOptimized
```

Contoh 3: Contoh ini me-reset atribut sourceDestCheck " untuk instance tertentu.

```
Reset-EC2InstanceAttribute -InstanceId i-12345678 -Attribute sourceDestCheck
```

Contoh 4: Contoh ini me-reset atribut disableApiTermination " untuk instance tertentu.

```
Reset-EC2InstanceAttribute -InstanceId i-12345678 -Attribute
disableApiTermination
```

Contoh 5: Contoh ini me-reset atribut 'instanceInitiatedShutdownBehavior' untuk instance tertentu.

```
Reset-EC2InstanceAttribute -InstanceId i-12345678 -Attribute
instanceInitiatedShutdownBehavior
```

- Untuk detail API, lihat [ResetInstanceAttribute](#) di Referensi AWS Tools for PowerShell Cmdlet.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **ResetNetworkInterfaceAttribute** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `ResetNetworkInterfaceAttribute`.

CLI

AWS CLI

Untuk mengatur ulang atribut antarmuka jaringan

`reset-network-interface-attribute` Contoh berikut me-reset nilai atribut pemeriksaan sumber/tujuan ke. `true`

```
aws ec2 reset-network-interface-attribute \
  --network-interface-id eni-686ea200 \
  --source-dest-check
```

Perintah ini tidak menghasilkan output.

- Untuk detail API, lihat [ResetNetworkInterfaceAttribute](#) di Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini mengatur ulang pemeriksaan sumber/tujuan untuk antarmuka jaringan yang ditentukan.

```
Reset-EC2NetworkInterfaceAttribute -NetworkInterfaceId eni-1a2b3c4d -  
SourceDestCheck
```

- Untuk detail API, lihat [ResetNetworkInterfaceAttribute](#) di Referensi AWS Tools for PowerShell Cmdlet.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **ResetSnapshotAttribute** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `ResetSnapshotAttribute`.

CLI

AWS CLI

Untuk mengatur ulang atribut snapshot

Contoh ini mengatur ulang izin volume buat untuk snapshot. `snap-1234567890abcdef0`
Jika perintah berhasil, tidak ada output yang akan ditampilkan.

Perintah:

```
aws ec2 reset-snapshot-attribute --snapshot-id snap-1234567890abcdef0 --attribute  
createVolumePermission
```

- Untuk detail API, lihat [ResetSnapshotAttribute](#) di Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini mengatur ulang atribut tertentu dari snapshot yang ditentukan.

```
Reset-EC2SnapshotAttribute -SnapshotId snap-12345678 -Attribute  
CreateVolumePermission
```

- Untuk detail API, lihat [ResetSnapshotAttribute](#) di Referensi AWS Tools for PowerShell Cmdlet.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **RevokeSecurityGroupEgress** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `RevokeSecurityGroupEgress`.

CLI

AWS CLI

Contoh 1: Untuk menghapus aturan yang memungkinkan lalu lintas keluar ke rentang alamat tertentu

`revoke-security-group-egress` Contoh perintah berikut menghapus aturan yang memberikan akses ke rentang alamat yang ditentukan pada port TCP 80.

```
aws ec2 revoke-security-group-egress \
  --group-id sg-026c12253ce15eff7 \
  --ip-permissions
  [{IpProtocol=tcp,FromPort=80,ToPort=80,IpRanges=[{CidrIp=10.0.0.0/16}]}
```

Perintah ini tidak menghasilkan output.

Untuk informasi selengkapnya, lihat [Grup keamanan](#) di Panduan Pengguna Amazon EC2.

Contoh 2: Untuk menghapus aturan yang memungkinkan lalu lintas keluar ke grup keamanan tertentu

`revoke-security-group-egress` Contoh perintah berikut menghapus aturan yang memberikan akses ke grup keamanan yang ditentukan pada port TCP 80.

```
aws ec2 revoke-security-group-egress \
  --group-id sg-026c12253ce15eff7 \
  --ip-permissions '[{"IpProtocol": "tcp", "FromPort": 443, "ToPort":
  443, "UserIdGroupPairs": [{"GroupId": "sg-06df23a01ff2df86d"}]}'
```


Perintah ini tidak menghasilkan output.

Untuk informasi selengkapnya, lihat [Grup keamanan](#) di Panduan Pengguna Amazon EC2.

- Untuk detail API, lihat [RevokeSecurityGroupEgress](#) di Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini menghapus aturan untuk grup keamanan yang ditentukan untuk EC2-VPC. Ini mencabut akses ke rentang alamat IP yang ditentukan pada port TCP 80. Sintaks yang digunakan oleh contoh ini membutuhkan PowerShell versi 3 atau lebih tinggi.

```
$ip = @{ IpProtocol="tcp"; FromPort="80"; ToPort="80";  
  IpRanges="203.0.113.0/24" }  
Revoke-EC2SecurityGroupEgress -GroupId sg-12345678 -IpPermission $ip
```

Contoh 2: Dengan PowerShell versi 2, Anda harus menggunakan New-Object untuk membuat objek. IpPermission

```
$ip = New-Object Amazon.EC2.Model.IpPermission  
$ip.IpProtocol = "tcp"  
$ip.FromPort = 80  
$ip.ToPort = 80  
$ip.IpRanges.Add("203.0.113.0/24")  
Revoke-EC2SecurityGroupEgress -GroupId sg-12345678 -IpPermission $ip
```

Contoh 3: Contoh ini mencabut akses ke grup keamanan sumber yang ditentukan pada port TCP 80.

```
$ug = New-Object Amazon.EC2.Model.UserIdGroupPair  
$ug.GroupId = "sg-1a2b3c4d"  
$ug.UserId = "123456789012"  
Revoke-EC2SecurityGroupEgress -GroupId sg-12345678 -IpPermission  
@( @{ IpProtocol="tcp"; FromPort="80"; ToPort="80"; UserIdGroupPairs=$ug } )
```

- Untuk detail API, lihat [RevokeSecurityGroupEgress](#) di Referensi AWS Tools for PowerShell Cmdlet.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan `RevokeSecurityGroupIngress` dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `RevokeSecurityGroupIngress`.

CLI

AWS CLI

Contoh 1: Untuk menghapus aturan dari grup keamanan

`revoke-security-group-ingress` Contoh berikut menghapus akses port TCP 22 untuk rentang `203.0.113.0/24` alamat dari grup keamanan yang ditentukan untuk VPC default.

```
aws ec2 revoke-security-group-ingress \
  --group-name mySecurityGroup \
  --protocol tcp \
  --port 22 \
  --cidr 203.0.113.0/24
```

Perintah ini tidak menghasilkan output jika berhasil.

Untuk informasi selengkapnya, lihat [Grup keamanan](#) di Panduan Pengguna Amazon EC2.

Contoh 2: Untuk menghapus aturan menggunakan set izin IP

`revoke-security-group-ingress` Contoh berikut menggunakan `ip-permissions` parameter untuk menghapus aturan masuk yang memungkinkan pesan ICMP Destination Unreachable: Fragmentation Needed and Don't Fragment was Set (Tipe 3, Kode 4).

```
aws ec2 revoke-security-group-ingress \
  --group-id sg-026c12253ce15eff7 \
  --ip-permissions \
  IpProtocol=icmp,FromPort=3,ToPort=4,IpRanges=[{CidrIp=0.0.0.0/0}]
```

Perintah ini tidak menghasilkan output jika berhasil.

Untuk informasi selengkapnya, lihat [Grup keamanan](#) di Panduan Pengguna Amazon EC2.

- Untuk detail API, lihat [RevokeSecurityGroupIngress](#) di Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini mencabut akses ke port TCP 22 dari rentang alamat yang ditentukan untuk grup keamanan yang ditentukan untuk EC2-VPC. Perhatikan bahwa Anda harus mengidentifikasi grup keamanan untuk EC2-VPC menggunakan ID grup keamanan bukan nama grup keamanan. Sintaks yang digunakan oleh contoh ini membutuhkan PowerShell versi 3 atau lebih tinggi.

```
$ip = @{ IpProtocol="tcp"; FromPort="22"; ToPort="22";  
  IpRanges="203.0.113.0/24" }  
Revoke-EC2SecurityGroupIngress -GroupId sg-12345678 -IpPermission $ip
```

Contoh 2: Dengan PowerShell versi 2, Anda harus menggunakan New-Object untuk membuat objek. IpPermission

```
$ip = New-Object Amazon.EC2.Model.IpPermission  
$ip.IpProtocol = "tcp"  
$ip.FromPort = 22  
$ip.ToPort = 22  
$ip.IpRanges.Add("203.0.113.0/24")  
  
Revoke-EC2SecurityGroupIngress -GroupId sg-12345678 -IpPermission $ip
```

Contoh 3: Contoh ini mencabut akses ke port TCP 22 dari rentang alamat yang ditentukan untuk grup keamanan yang ditentukan untuk EC2-Classic. Sintaks yang digunakan oleh contoh ini membutuhkan PowerShell versi 3 atau lebih tinggi.

```
$ip = @{ IpProtocol="tcp"; FromPort="22"; ToPort="22";  
  IpRanges="203.0.113.0/24" }  
  
Revoke-EC2SecurityGroupIngress -GroupName "my-security-group" -IpPermission $ip
```

Contoh 4: Dengan PowerShell versi 2, Anda harus menggunakan New-Object untuk membuat objek. IpPermission

```
$ip = New-Object Amazon.EC2.Model.IpPermission
```

```
$ip.IpProtocol = "tcp"
$ip.FromPort = 22
$ip.ToPort = 22
$ip.IpRanges.Add("203.0.113.0/24")

Revoke-EC2SecurityGroupIngress -GroupName "my-security-group" -IpPermission $ip
```

- Untuk detail API, lihat [RevokeSecurityGroupIngress](#) di Referensi AWS Tools for PowerShell Cmdlet.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **RunInstances** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `RunInstances`.

Contoh tindakan adalah kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Anda dapat melihat tindakan ini dalam konteks dalam contoh kode berikut:

- [Memulai instans](#)

.NET

AWS SDK for .NET

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
/// <summary>
/// Create and run an EC2 instance.
/// </summary>
/// <param name="ImageId">The image Id of the image used as a basis for the
/// EC2 instance.</param>
/// <param name="instanceType">The instance type of the EC2 instance to
create.</param>
```

```

    /// <param name="keyName">The name of the key pair to associate with the
    /// instance.</param>
    /// <param name="groupId">The Id of the Amazon EC2 security group that will
    be
    /// allowed to interact with the new EC2 instance.</param>
    /// <returns>The instance Id of the new EC2 instance.</returns>
    public async Task<string> RunInstances(string imageId, string instanceType,
    string keyName, string groupId)
    {
        var request = new RunInstancesRequest
        {
            ImageId = imageId,
            InstanceType = instanceType,
            KeyName = keyName,
            MinCount = 1,
            MaxCount = 1,
            SecurityGroupIds = new List<string> { groupId }
        };
        var response = await _amazonEC2.RunInstancesAsync(request);
        return response.Reservation.Instances[0].InstanceId;
    }

```

- Untuk detail API, lihat [RunInstances](#) di Referensi AWS SDK for .NET API.

Bash

AWS CLI dengan skrip Bash

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```

#####
# function ec2_run_instances
#
# This function launches one or more Amazon Elastic Compute Cloud (Amazon EC2)
# instances.
#

```

```

# Parameters:
#     -i image_id - The ID of the Amazon Machine Image (AMI) to use.
#     -t instance_type - The instance type to use (e.g., t2.micro).
#     -k key_pair_name - The name of the key pair to use.
#     -s security_group_id - The ID of the security group to use.
#     -c count - The number of instances to launch (default: 1).
#     -h - Display help.
#
# Returns:
#     0 - If successful.
#     1 - If it fails.
#####
function ec2_run_instances() {
    local image_id instance_type key_pair_name security_group_id count response
    local option OPTARG # Required to use getopt command in a function.

    # bashsupport disable=BP5008
    function usage() {
        echo "function ec2_run_instances"
        echo "Launches one or more Amazon Elastic Compute Cloud (Amazon EC2)
instances."
        echo "  -i image_id - The ID of the Amazon Machine Image (AMI) to use."
        echo "  -t instance_type - The instance type to use (e.g., t2.micro)."
        echo "  -k key_pair_name - The name of the key pair to use."
        echo "  -s security_group_id - The ID of the security group to use."
        echo "  -c count - The number of instances to launch (default: 1)."
        echo "  -h - Display help."
        echo ""
    }

    # Retrieve the calling parameters.
    while getopt "i:t:k:s:c:h" option; do
        case "${option}" in
            i) image_id="${OPTARG}" ;;
            t) instance_type="${OPTARG}" ;;
            k) key_pair_name="${OPTARG}" ;;
            s) security_group_id="${OPTARG}" ;;
            c) count="${OPTARG}" ;;
            h)
                usage
                return 0
                ;;
            \?)
                echo "Invalid parameter"

```

```
        usage
        return 1
    ;;
esac
done
export OPTIND=1

if [[ -z "$image_id" ]]; then
    errecho "ERROR: You must provide an Amazon Machine Image (AMI) ID with the -i
parameter."
    usage
    return 1
fi

if [[ -z "$instance_type" ]]; then
    errecho "ERROR: You must provide an instance type with the -t parameter."
    usage
    return 1
fi

if [[ -z "$key_pair_name" ]]; then
    errecho "ERROR: You must provide a key pair name with the -k parameter."
    usage
    return 1
fi

if [[ -z "$security_group_id" ]]; then
    errecho "ERROR: You must provide a security group ID with the -s parameter."
    usage
    return 1
fi

if [[ -z "$count" ]]; then
    count=1
fi

response=$(aws ec2 run-instances \
    --image-id "$image_id" \
    --instance-type "$instance_type" \
    --key-name "$key_pair_name" \
    --security-group-ids "$security_group_id" \
    --count "$count" \
    --query 'Instances[*].[InstanceId]' \
    --output text) || {
```

```

aws_cli_error_log ${?}
errecho "ERROR: AWS reports run-instances operation failed.$response"
return 1
}

echo "$response"

return 0
}

```

Fungsi utilitas yang digunakan dalam contoh ini.

```

#####
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
#####
function errecho() {
    printf "%s\n" "$*" 1>&2
}

#####
# function aws_cli_error_log()
#
# This function is used to log the error messages from the AWS CLI.
#
# The function expects the following argument:
#     $1 - The error code returned by the AWS CLI.
#
# Returns:
#     0: - Success.
#
#####
function aws_cli_error_log() {
    local err_code=$1
    errecho "Error code : $err_code"
    if [ "$err_code" == 1 ]; then
        errecho " One or more S3 transfers failed."
    elif [ "$err_code" == 2 ]; then
        errecho " Command line failed to parse."
    elif [ "$err_code" == 130 ]; then
        errecho " Process received SIGINT."
    fi
}

```



```
elif [ "$err_code" == 252 ]; then
    errecho " Command syntax invalid."
elif [ "$err_code" == 253 ]; then
    errecho " The system environment or configuration was invalid."
elif [ "$err_code" == 254 ]; then
    errecho " The service returned an error."
elif [ "$err_code" == 255 ]; then
    errecho " 255 is a catch-all error."
fi

return 0
}
```

- Untuk detail API, lihat [RunInstances](#) di Referensi AWS CLI Perintah.

C++

SDK untuk C++

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
Aws::EC2::EC2Client ec2Client(clientConfiguration);

Aws::EC2::Model::RunInstancesRequest runRequest;
runRequest.SetImageId(amiId);
runRequest.SetInstanceType(Aws::EC2::Model::InstanceType::t1_micro);
runRequest.SetMinCount(1);
runRequest.SetMaxCount(1);

Aws::EC2::Model::RunInstancesOutcome runOutcome = ec2Client.RunInstances(
    runRequest);
if (!runOutcome.IsSuccess()) {
    std::cerr << "Failed to launch EC2 instance " << instanceName <<
        " based on ami " << amiId << ":" <<
        runOutcome.GetError().GetMessage() << std::endl;
    return false;
}
```

```
const Aws::Vector<Aws::EC2::Model::Instance> &instances =
runOutcome.GetResult().GetInstances();
if (instances.empty()) {
    std::cerr << "Failed to launch EC2 instance " << instanceName <<
        " based on ami " << amiId << ":" <<
        runOutcome.GetError().GetMessage() << std::endl;
    return false;
}

instanceID = instances[0].GetInstanceId();
```

- Untuk detail API, lihat [RunInstances](#) di Referensi AWS SDK for C++ API.

CLI

AWS CLI

Contoh 1: Untuk meluncurkan instans ke subnet default

Contoh `run-instances` berikut meluncurkan instans tunggal tipe `t2.micro` ke subnet default untuk Wilayah saat ini dan mengaitkannya dengan subnet default untuk VPC default pada Wilayah tersebut. Pasangan Kunci bersifat opsional jika Anda tidak berencana untuk terhubung ke instans menggunakan SSH (Linux) atau RDP (Windows).

```
aws ec2 run-instances \
  --image-id ami-0abcdef1234567890 \
  --instance-type t2.micro \
  --key-name MyKeyPair
```

Output:

```
{
  "Instances": [
    {
      "AmiLaunchIndex": 0,
      "ImageId": "ami-0abcdef1234567890",
      "InstanceId": "i-1231231230abcdef0",
      "InstanceType": "t2.micro",
      "KeyName": "MyKeyPair",
      "LaunchTime": "2018-05-10T08:05:20.000Z",
```

```
"Monitoring": {
  "State": "disabled"
},
"Placement": {
  "AvailabilityZone": "us-east-2a",
  "GroupName": "",
  "Tenancy": "default"
},
"PrivateDnsName": "ip-10-0-0-157.us-east-2.compute.internal",
"PrivateIpAddress": "10.0.0.157",
"ProductCodes": [],
"PublicDnsName": "",
"State": {
  "Code": 0,
  "Name": "pending"
},
"StateTransitionReason": "",
"SubnetId": "subnet-04a636d18e83cfacb",
"VpcId": "vpc-1234567890abcdef0",
"Architecture": "x86_64",
"BlockDeviceMappings": [],
"ClientToken": "",
"EbsOptimized": false,
"Hypervisor": "xen",
"NetworkInterfaces": [
  {
    "Attachment": {
      "AttachTime": "2018-05-10T08:05:20.000Z",
      "AttachmentId": "eni-attach-0e325c07e928a0405",
      "DeleteOnTermination": true,
      "DeviceIndex": 0,
      "Status": "attaching"
    },
    "Description": "",
    "Groups": [
      {
        "GroupName": "MySecurityGroup",
        "GroupId": "sg-0598c7d356eba48d7"
      }
    ],
    "Ipv6Addresses": [],
    "MacAddress": "0a:ab:58:e0:67:e2",
    "NetworkInterfaceId": "eni-0c0a29997760baee7",
    "OwnerId": "123456789012",
```

```
        "PrivateDnsName": "ip-10-0-0-157.us-east-2.compute.internal",
        "PrivateIpAddress": "10.0.0.157",
        "PrivateIpAddresses": [
            {
                "Primary": true,
                "PrivateDnsName": "ip-10-0-0-157.us-
east-2.compute.internal",
                "PrivateIpAddress": "10.0.0.157"
            }
        ],
        "SourceDestCheck": true,
        "Status": "in-use",
        "SubnetId": "subnet-04a636d18e83cfacb",
        "VpcId": "vpc-1234567890abcdef0",
        "InterfaceType": "interface"
    }
],
"RootDeviceName": "/dev/xvda",
"RootDeviceType": "ebs",
"SecurityGroups": [
    {
        "GroupName": "MySecurityGroup",
        "GroupId": "sg-0598c7d356eba48d7"
    }
],
"SourceDestCheck": true,
"StateReason": {
    "Code": "pending",
    "Message": "pending"
},
"Tags": [],
"VirtualizationType": "hvm",
"CpuOptions": {
    "CoreCount": 1,
    "ThreadsPerCore": 1
},
"CapacityReservationSpecification": {
    "CapacityReservationPreference": "open"
},
"MetadataOptions": {
    "State": "pending",
    "HttpTokens": "optional",
    "HttpPutResponseHopLimit": 1,
    "HttpEndpoint": "enabled"
```

```

    }
  }
],
"OwnerId": "123456789012",
"ReservationId": "r-02a3f596d91211712"
}

```

Contoh 2: Untuk meluncurkan instans ke subnet non-default dan menambahkan alamat IP publik

Contoh `run-instances` berikut meminta alamat IP publik untuk instans yang Anda luncurkan ke subnet non-default. Instans tersebut dikaitkan dengan grup keamanan tertentu.

```

aws ec2 run-instances \
  --image-id ami-0abcdef1234567890 \
  --instance-type t2.micro \
  --subnet-id subnet-08fc749671b2d077c \
  --security-group-ids sg-0b0384b66d7d692f9 \
  --associate-public-ip-address \
  --key-name MyKeyPair

```

Untuk contoh output `run-instances`, lihat Contoh 1.

Contoh 3: Untuk meluncurkan instans dengan volume tambahan

Contoh `run-instances` berikut menggunakan pemetaan perangkat blok, yang ditentukan dalam `mapping.json`, untuk melampirkan volume tambahan saat peluncuran. Pemetaan perangkat blok dapat menentukan volume EBS, volume penyimpanan instans, atau volume EBS dan volume penyimpanan instans.

```

aws ec2 run-instances \
  --image-id ami-0abcdef1234567890 \
  --instance-type t2.micro \
  --subnet-id subnet-08fc749671b2d077c \
  --security-group-ids sg-0b0384b66d7d692f9 \
  --key-name MyKeyPair \
  --block-device-mappings file://mapping.json

```

Isi dari `mapping.json`. Contoh ini menambahkan `/dev/sdh` sebagai volume EBS kosong dengan ukuran 100 GiB.

```
[
```

```
{
  "DeviceName": "/dev/sdh",
  "Ebs": {
    "VolumeSize": 100
  }
}
```

Isi dari `mapping.json`. Contoh ini menambahkan `ephemeral1` sebagai volume penyimpanan instans.

```
[
  {
    "DeviceName": "/dev/sdc",
    "VirtualName": "ephemeral1"
  }
]
```

Untuk contoh output `run-instances`, lihat Contoh 1.

Untuk informasi selengkapnya tentang pemetaan perangkat blok, lihat [Pemetaan perangkat blok](#) di Panduan Pengguna Amazon EC2.

Contoh 4: Untuk meluncurkan instans dan menambahkan tanda pada pembuatan

Contoh `run-instances` berikut menambahkan tanda dengan kunci `webserver` dan nilai `production` pada instans. Perintah tersebut juga menerapkan tanda dengan kunci `cost-center` dan nilai `cc123` pada volume EBS yang dibuat (dalam hal ini, volume root).

```
aws ec2 run-instances \
  --image-id ami-0abcdef1234567890 \
  --instance-type t2.micro \
  --count 1 \
  --subnet-id subnet-08fc749671b2d077c \
  --key-name MyKeyPair \
  --security-group-ids sg-0b0384b66d7d692f9 \
  --tag-specifications
'ResourceType=instance,Tags=[{Key=webserver,Value=production}]'
'ResourceType=volume,Tags=[{Key=cost-center,Value=cc123}]'
```

Untuk contoh output `run-instances`, lihat Contoh 1.

Contoh 5: Untuk meluncurkan instans dengan data pengguna

Contoh `run-instances` berikut meneruskan data pengguna dalam file bernama `my_script.txt` yang berisi skrip konfigurasi untuk instans Anda. Skrip berjalan saat peluncuran.

```
aws ec2 run-instances \  
  --image-id ami-0abcdef1234567890 \  
  --instance-type t2.micro \  
  --count 1 \  
  --subnet-id subnet-08fc749671b2d077c \  
  --key-name MyKeyPair \  
  --security-group-ids sg-0b0384b66d7d692f9 \  
  --user-data file://my_script.txt
```

Untuk contoh output `run-instances`, lihat Contoh 1.

Untuk informasi selengkapnya tentang data pengguna instans, lihat [Bekerja dengan data pengguna instans](#) di Panduan Pengguna Amazon EC2.

Contoh 6: Untuk meluncurkan instans performa yang dapat melonjak

Contoh `run-instances` berikut meluncurkan instans `t2.micro` dengan opsi kredit `unlimited`. Ketika meluncurkan instans T2, jika Anda tidak menentukan `--credit-specification`, default-nya adalah opsi kredit `standard`. Ketika meluncurkan instans T3, default-nya adalah opsi kredit `unlimited`.

```
aws ec2 run-instances \  
  --image-id ami-0abcdef1234567890 \  
  --instance-type t2.micro \  
  --count 1 \  
  --subnet-id subnet-08fc749671b2d077c \  
  --key-name MyKeyPair \  
  --security-group-ids sg-0b0384b66d7d692f9 \  
  --credit-specification CpuCredits=unlimited
```

Untuk contoh output `run-instances`, lihat Contoh 1.

Untuk informasi selengkapnya tentang instans performa yang dapat melonjak, lihat [Instans performa yang dapat melonjak](#) di Panduan Pengguna Amazon EC2.

- Untuk detail API, lihat [RunInstances](#) di Referensi AWS CLI Perintah.

Java

SDK untuk Java 2.x

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.ec2.Ec2Client;
import software.amazon.awssdk.services.ec2.model.InstanceType;
import software.amazon.awssdk.services.ec2.model.RunInstancesRequest;
import software.amazon.awssdk.services.ec2.model.RunInstancesResponse;
import software.amazon.awssdk.services.ec2.model.Tag;
import software.amazon.awssdk.services.ec2.model.CreateTagsRequest;
import software.amazon.awssdk.services.ec2.model.Ec2Exception;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
 * started.html
 *
 * This code example requires an AMI value. You can learn more about this value
 * by reading this documentation topic:
 *
 * https://docs.aws.amazon.com/AWSEC2/latest/WindowsGuide/AMIs.html
 */
public class CreateInstance {
    public static void main(String[] args) {
        final String usage = ""

            Usage:
                <name> <amiId>

            Where:
```



```
        name - An instance name value that you can obtain from the AWS
Console (for example, ami-xxxxxx5c8b987b1a0).\s
        amiId - An Amazon Machine Image (AMI) value that you can
obtain from the AWS Console (for example, i-xxxxxx2734106d0ab).\s
        """;

    if (args.length != 2) {
        System.out.println(usage);
        System.exit(1);
    }

    String name = args[0];
    String amiId = args[1];
    Region region = Region.US_EAST_1;
    Ec2Client ec2 = Ec2Client.builder()
        .region(region)
        .build();

    String instanceId = createEC2Instance(ec2, name, amiId);
    System.out.println("The Amazon EC2 Instance ID is " + instanceId);
    ec2.close();
}

public static String createEC2Instance(Ec2Client ec2, String name, String
amiId) {
    RunInstancesRequest runRequest = RunInstancesRequest.builder()
        .imageId(amiId)
        .instanceType(InstanceType.T1_MICRO)
        .maxCount(1)
        .minCount(1)
        .build();

    // Use a waiter to wait until the instance is running.
    System.out.println("Going to start an EC2 instance using a waiter");
    RunInstancesResponse response = ec2.runInstances(runRequest);
    String instanceIdVal = response.instances().get(0).instanceId();
    ec2.waiter().waitUntilInstanceRunning(r -> r.instanceIds(instanceIdVal));
    Tag tag = Tag.builder()
        .key("Name")
        .value(name)
        .build();

    CreateTagsRequest tagRequest = CreateTagsRequest.builder()
        .resources(instanceIdVal)
```

```
        .tags(tag)
        .build();

    try {
        ec2.createTags(tagRequest);
        System.out.printf("Successfully started EC2 Instance %s based on AMI
%s", instanceIdVal, amiId);
        return instanceIdVal;

    } catch (Ec2Exception e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }

    return "";
}
}
```

- Untuk detail API, lihat [RunInstances](#) di Referensi AWS SDK for Java 2.x API.

JavaScript

SDK untuk JavaScript (v3)

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
import { RunInstancesCommand } from "@aws-sdk/client-ec2";

import { client } from "../libs/client.js";

// Create a new EC2 instance.
export const main = async () => {
    const command = new RunInstancesCommand({
        // Your key pair name.
        KeyName: "KEY_PAIR_NAME",
        // Your security group.
        SecurityGroupIds: ["SECURITY_GROUP_ID"],
```

```
// An x86_64 compatible image.
ImageId: "ami-0001a0d1a04bfcc30",
// An x86_64 compatible free-tier instance type.
InstanceType: "t1.micro",
// Ensure only 1 instance launches.
MinCount: 1,
MaxCount: 1,
});

try {
  const response = await client.send(command);
  console.log(response);
} catch (err) {
  console.error(err);
}
};
```

- Untuk detail API, lihat [RunInstances](#) di Referensi AWS SDK for JavaScript API.

Kotlin

SDK untuk Kotlin

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
suspend fun createEC2Instance(
  name: String,
  amiId: String,
): String? {
  val request =
    RunInstancesRequest {
      imageId = amiId
      instanceType = InstanceType.T1Micro
      maxCount = 1
      minCount = 1
    }
}
```

```
Ec2Client { region = "us-west-2" }.use { ec2 ->
    val response = ec2.runInstances(request)
    val instanceId = response.instances?.get(0)?.instanceId
    val tag =
        Tag {
            key = "Name"
            value = name
        }

    val requestTags =
        CreateTagsRequest {
            resources = listOf(instanceId.toString())
            tags = listOf(tag)
        }
    ec2.createTags(requestTags)
    println("Successfully started EC2 Instance $instanceId based on AMI
$amiId")
    return instanceId
}
}
```

- Untuk detail API, lihat [RunInstances](#) di AWS SDK untuk referensi API Kotlin.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini meluncurkan satu instance dari AMI yang ditentukan di EC2-Classic atau VPC default.

```
New-EC2Instance -ImageId ami-12345678 -MinCount 1 -MaxCount 1 -InstanceType
m3.medium -KeyName my-key-pair -SecurityGroup my-security-group
```

Contoh 2: Contoh ini meluncurkan satu instance dari AMI yang ditentukan dalam VPC.

```
New-EC2Instance -ImageId ami-12345678 -MinCount 1 -MaxCount 1 -SubnetId
subnet-12345678 -InstanceType t2.micro -KeyName my-key-pair -SecurityGroupId
sg-12345678
```

Contoh 3: Untuk menambahkan volume EBS atau volume penyimpanan instance, tentukan pemetaan perangkat blok dan tambahkan ke perintah. Contoh ini menambahkan volume penyimpanan instance.

```
$bdm = New-Object Amazon.EC2.Model.BlockDeviceMapping
$bdm.VirtualName = "ephemeral0"
$bdm.DeviceName = "/dev/sdf"

New-EC2Instance -ImageId ami-12345678 -BlockDeviceMapping $bdm ...
```

Contoh 4: Untuk menentukan salah satu AMI Windows saat ini, dapatkan ID AMI menggunakan `Get-EC2ImageByName`. Contoh ini meluncurkan instance dari basis AMI saat ini untuk Windows Server 2016.

```
$ami = Get-EC2ImageByName WINDOWS_2016_BASE

New-EC2Instance -ImageId $ami.ImageId ...
```

Contoh 5: Meluncurkan instance ke lingkungan host khusus yang ditentukan.

```
New-EC2Instance -ImageId ami-1a2b3c4d -InstanceType m4.large -KeyName my-key-pair
-SecurityGroupId sg-1a2b3c4d -AvailabilityZone us-west-1a -Tenancy host -HostID
h-1a2b3c4d5e6f1a2b3
```

Contoh 6: Permintaan ini meluncurkan dua instance dan menerapkan tag dengan kunci server web dan nilai produksi ke instance. Permintaan juga menerapkan tag dengan kunci cost-center dan nilai cc123 ke volume yang dibuat (dalam hal ini, volume root untuk setiap instance).

```
$tag1 = @{ Key="webserver"; Value="production" }
$tag2 = @{ Key="cost-center"; Value="cc123" }

$tagspec1 = new-object Amazon.EC2.Model.TagSpecification
$tagspec1.ResourceType = "instance"
$tagspec1.Tags.Add($tag1)

$tagspec2 = new-object Amazon.EC2.Model.TagSpecification
$tagspec2.ResourceType = "volume"
$tagspec2.Tags.Add($tag2)
```

```
New-EC2Instance -ImageId "ami-1a2b3c4d" -KeyName "my-key-pair" -MaxCount 2 -
InstanceType "t2.large" -SubnetId "subnet-1a2b3c4d" -TagSpecification $tagspec1,
$tagspec2
```

- Untuk detail API, lihat [RunInstances](#) di Referensi AWS Tools for PowerShell Cmdlet.

Python

SDK untuk Python (Boto3)

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
class InstanceWrapper:
    """Encapsulates Amazon Elastic Compute Cloud (Amazon EC2) instance
    actions."""

    def __init__(self, ec2_resource, instance=None):
        """
        :param ec2_resource: A Boto3 Amazon EC2 resource. This high-level
        resource
                               is used to create additional high-level objects
                               that wrap low-level Amazon EC2 service actions.
        :param instance: A Boto3 Instance object. This is a high-level object
        that
                               wraps instance actions.
        """
        self.ec2_resource = ec2_resource
        self.instance = instance

    @classmethod
    def from_resource(cls):
        ec2_resource = boto3.resource("ec2")
        return cls(ec2_resource)

    def create(self, image, instance_type, key_pair, security_groups=None):
        """
        Creates a new EC2 instance. The instance starts immediately after
```

```

        it is created.

        The instance is created in the default VPC of the current account.

        :param image: A Boto3 Image object that represents an Amazon Machine
Image (AMI)
                that defines attributes of the instance that is created.
The AMI
                defines things like the kind of operating system and the
type of
                storage used by the instance.
        :param instance_type: The type of instance to create, such as 't2.micro'.
                The instance type defines things like the number of
CPUs and
                the amount of memory.
        :param key_pair: A Boto3 KeyPair or KeyPairInfo object that represents
the key
                pair that is used to secure connections to the instance.
        :param security_groups: A list of Boto3 SecurityGroup objects that
represents the
                security groups that are used to grant access to
the
                instance. When no security groups are specified,
the
                default security group of the VPC is used.
        :return: A Boto3 Instance object that represents the newly created
instance.
        """
        try:
            instance_params = {
                "ImageId": image.id,
                "InstanceType": instance_type,
                "KeyName": key_pair.name,
            }
            if security_groups is not None:
                instance_params["SecurityGroupIds"] = [sg.id for sg in
security_groups]
            self.instance = self.ec2_resource.create_instances(
                **instance_params, MinCount=1, MaxCount=1
            )[0]
            self.instance.wait_until_running()
        except ClientError as err:
            logging.error(

```

```

        "Couldn't create instance with image %s, instance type %s, and
key %s. "
        "Here's why: %s: %s",
        image.id,
        instance_type,
        key_pair.name,
        err.response["Error"]["Code"],
        err.response["Error"]["Message"],
    )
    raise
else:
    return self.instance

```

- Untuk detail API, lihat [RunInstances](#) di AWS SDK for Python (Boto3) Referensi API.

SAP ABAP

SDK untuk SAP ABAP

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```

" Create tags for resource created during instance launch. "
DATA lt_tagsspecifications TYPE /aws1/
cl_ec2tagsspecification=>tt_tagsspecificationlist.
DATA ls_tagsspecifications LIKE LINE OF lt_tagsspecifications.
ls_tagsspecifications = NEW /aws1/cl_ec2tagsspecification(
    iv_resourcetype = 'instance'
    it_tags = VALUE /aws1/cl_ec2tag=>tt_taglist(
        ( NEW /aws1/cl_ec2tag( iv_key = 'Name' iv_value = iv_tag_value ) )
    )
).
APPEND ls_tagsspecifications TO lt_tagsspecifications.

TRY.
    " Create/launch Amazon Elastic Compute Cloud (Amazon EC2) instance. "

```



```

        oo_result = lo_ec2->runinstances(
            is returned for testing purposes. "
            iv_imageid = iv_ami_id
            iv_instancetype = 't2.micro'
            iv_maxcount = 1
            iv_mincount = 1
            it_tagspecifications = lt_tagspecifications
            iv_subnetid = iv_subnet_id
        ).
        MESSAGE 'EC2 instance created.' TYPE 'I'.
    CATCH /aws1/cx_rt_service_generic INTO DATA(lo_exception).
        DATA(lv_error) = |"{ lo_exception->av_err_code }" - { lo_exception-
>av_err_msg }|.
        MESSAGE lv_error TYPE 'E'.
    ENDTRY.

```

- Untuk detail API, lihat [RunInstances](#) di AWS SDK untuk referensi SAP ABAP API.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **RunScheduledInstances** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `RunScheduledInstances`.

CLI

AWS CLI

Untuk meluncurkan Instance Terjadwal

Contoh ini meluncurkan Instance Terjadwal yang ditentukan dalam VPC.

Perintah:

```

aws ec2 run-scheduled-instances --scheduled-instance-id
sci-1234-1234-1234-1234-123456789012 --instance-count 1 --launch-specification
file://launch-specification.json

```

Launch-specification.json:

```
{
  "ImageId": "ami-12345678",
  "KeyName": "my-key-pair",
  "InstanceType": "c4.large",
  "NetworkInterfaces": [
    {
      "DeviceIndex": 0,
      "SubnetId": "subnet-12345678",
      "AssociatePublicIpAddress": true,
      "Groups": ["sg-12345678"]
    }
  ],
  "IamInstanceProfile": {
    "Name": "my-iam-role"
  }
}
```

Output:

```
{
  "InstanceIdSet": [
    "i-1234567890abcdef0"
  ]
}
```

Contoh ini meluncurkan Instance Terjadwal yang ditentukan di EC2-Classic.

Perintah:

```
aws ec2 run-scheduled-instances --scheduled-instance-id
sci-1234-1234-1234-1234-123456789012 --instance-count 1 --launch-specification
file://launch-specification.json
```

Launch-specification.json:

```
{
  "ImageId": "ami-12345678",
  "KeyName": "my-key-pair",
  "SecurityGroupIds": ["sg-12345678"],
  "InstanceType": "c4.large",
  "Placement": {
    "AvailabilityZone": "us-west-2b"
  }
}
```

```
}
  "IamInstanceProfile": {
    "Name": "my-iam-role"
  }
}
```

Output:

```
{
  "InstanceIdSet": [
    "i-1234567890abcdef0"
  ]
}
```

- Untuk detail API, lihat [RunScheduledInstances](#) di Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini meluncurkan Instance Terjadwal yang ditentukan.

```
New-EC2ScheduledInstance -ScheduledInstanceId
sci-1234-1234-1234-1234-123456789012 -InstanceCount 1 `
-IamInstanceProfile_Name my-iam-role `
-LaunchSpecification_ImageId ami-12345678 `
-LaunchSpecification_InstanceType c4.large `
-LaunchSpecification_SubnetId subnet-12345678 `
-LaunchSpecification_SecurityGroupId sg-12345678
```

- Untuk detail API, lihat [RunScheduledInstances](#) di Referensi AWS Tools for PowerShell Cmdlet.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **StartInstances** dengan AWS SDK atau CLI


Contoh kode berikut menunjukkan cara menggunakan `StartInstances`.

Contoh tindakan adalah kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Anda dapat melihat tindakan ini dalam konteks dalam contoh kode berikut:

- [Memulai instans](#)

.NET

AWS SDK for .NET

 Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
/// <summary>
/// Start an EC2 instance.
/// </summary>
/// <param name="ec2InstanceId">The instance Id of the Amazon EC2 instance
/// to start.</param>
/// <returns>Async task.</returns>
public async Task StartInstances(string ec2InstanceId)
{
    var request = new StartInstancesRequest
    {
        InstanceIds = new List<string> { ec2InstanceId },
    };

    var response = await _amazonEC2.StartInstancesAsync(request);

    if (response.StartingInstances.Count > 0)
    {
        var instances = response.StartingInstances;
        instances.ForEach(i =>
        {
            Console.WriteLine($"Successfully started the EC2 instance with
instance ID: {i.InstanceId}.");
        });
    }
}
```

- Untuk detail API, lihat [StartInstances](#) di Referensi AWS SDK for .NET API.

Bash

AWS CLI dengan skrip Bash

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
#####
# function ec2_start_instances
#
# This function starts one or more Amazon Elastic Compute Cloud (Amazon EC2)
# instances.
#
# Parameters:
#     -i instance_id - The ID(s) of the instance(s) to start (comma-separated).
#     -h - Display help.
#
# Returns:
#     0 - If successful.
#     1 - If it fails.
#####
function ec2_start_instances() {
    local instance_ids
    local option OPTARG # Required to use getopt command in a function.

    # bashsupport disable=BP5008
    function usage() {
        echo "function ec2_start_instances"
        echo "Starts one or more Amazon Elastic Compute Cloud (Amazon EC2)
instances."
        echo "  -i instance_id - The ID(s) of the instance(s) to start (comma-
separated)."
        echo "  -h - Display help."
        echo ""
    }

    # Retrieve the calling parameters.
```

```

while getopts "i:h" option; do
  case "${option}" in
    i) instance_ids="${OPTARG}" ;;
    h)
      usage
      return 0
      ;;
    \?)
      echo "Invalid parameter"
      usage
      return 1
      ;;
  esac
done
export OPTIND=1

if [[ -z "$instance_ids" ]]; then
  errecho "ERROR: You must provide one or more instance IDs with the -i
parameter."
  usage
  return 1
fi

response=$(aws ec2 start-instances \
  --instance-ids "${instance_ids}") || {
  aws_cli_error_log ${?}
  errecho "ERROR: AWS reports start-instances operation failed with $response."
  return 1
}

return 0
}

```

Fungsi utilitas yang digunakan dalam contoh ini.

```

#####
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
#####
function errecho() {
  printf "%s\n" "$*" 1>&2
}

```

```

}

#####
# function aws_cli_error_log()
#
# This function is used to log the error messages from the AWS CLI.
#
# The function expects the following argument:
#     $1 - The error code returned by the AWS CLI.
#
# Returns:
#     0: - Success.
#
#####
function aws_cli_error_log() {
    local err_code=$1
    errecho "Error code : $err_code"
    if [ "$err_code" == 1 ]; then
        errecho " One or more S3 transfers failed."
    elif [ "$err_code" == 2 ]; then
        errecho " Command line failed to parse."
    elif [ "$err_code" == 130 ]; then
        errecho " Process received SIGINT."
    elif [ "$err_code" == 252 ]; then
        errecho " Command syntax invalid."
    elif [ "$err_code" == 253 ]; then
        errecho " The system environment or configuration was invalid."
    elif [ "$err_code" == 254 ]; then
        errecho " The service returned an error."
    elif [ "$err_code" == 255 ]; then
        errecho " 255 is a catch-all error."
    fi


    return 0
}

```

- Untuk detail API, lihat [StartInstances](#) di Referensi AWS CLI Perintah.

C++

SDK untuk C++

 Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
Aws::EC2::EC2Client ec2Client(clientConfiguration);

Aws::EC2::Model::StartInstancesRequest start_request;
start_request.AddInstanceIds(instanceId);
start_request.SetDryRun(true);

auto dry_run_outcome = ec2Client.StartInstances(start_request);
if (dry_run_outcome.IsSuccess()) {
    std::cerr
        << "Failed dry run to start instance. A dry run should trigger an
error."
        << std::endl;
    return false;
}
else if (dry_run_outcome.GetError().GetErrorType() !=
    Aws::EC2::EC2Errors::DRY_RUN_OPERATION) {
    std::cout << "Failed dry run to start instance " << instanceId << ": "
        << dry_run_outcome.GetError().GetMessage() << std::endl;
    return false;
}

start_request.SetDryRun(false);
auto start_instancesOutcome = ec2Client.StartInstances(start_request);

if (!start_instancesOutcome.IsSuccess()) {
    std::cout << "Failed to start instance " << instanceId << ": " <<
        start_instancesOutcome.GetError().GetMessage() << std::endl;
}
else {
    std::cout << "Successfully started instance " << instanceId <<
        std::endl;
}
}
```


- Untuk detail API, lihat [StartInstances](#) di Referensi AWS SDK for C++ API.

CLI

AWS CLI

Untuk memulai instans Amazon EC2

Contoh ini memulai instans yang didukung Amazon EBS tertentu.

Perintah:

```
aws ec2 start-instances --instance-ids i-1234567890abcdef0
```

Output:

```
{
  "StartingInstances": [
    {
      "InstanceId": "i-1234567890abcdef0",
      "CurrentState": {
        "Code": 0,
        "Name": "pending"
      },
      "PreviousState": {
        "Code": 80,
        "Name": "stopped"
      }
    }
  ]
}
```

Untuk informasi selengkapnya, lihat Hentikan dan Mulai Instans Anda di Panduan Pengguna Amazon Elastic Compute Cloud.

- Untuk detail API, lihat [StartInstances](#) di Referensi AWS CLI Perintah.

Java

SDK untuk Java 2.x

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
public static void startInstance(Ec2Client ec2, String instanceId) {
    Ec2Waiter ec2Waiter = Ec2Waiter.builder()
        .overrideConfiguration(b -> b.maxAttempts(100))
        .client(ec2)
        .build();

    StartInstancesRequest request = StartInstancesRequest.builder()
        .instanceIds(instanceId)
        .build();

    System.out.println("Use an Ec2Waiter to wait for the instance to run.
This will take a few minutes.");
    ec2.startInstances(request);
    DescribeInstancesRequest instanceRequest =
DescribeInstancesRequest.builder()
        .instanceIds(instanceId)
        .build();

    WaiterResponse<DescribeInstancesResponse> waiterResponse =
ec2Waiter.waitUntilInstanceRunning(instanceRequest);
    waiterResponse.matched().response().ifPresent(System.out::println);
    System.out.println("Successfully started instance " + instanceId);
}
```

- Untuk detail API, lihat [StartInstances](#) di Referensi AWS SDK for Java 2.x API.

JavaScript

SDK untuk JavaScript (v3)

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
import { StartInstancesCommand } from "@aws-sdk/client-ec2";

import { client } from "../libs/client.js";

export const main = async () => {
  const command = new StartInstancesCommand({
    // Use DescribeInstancesCommand to find InstanceIds
    InstanceIds: ["INSTANCE_ID"],
  });

  try {
    const { StartingInstances } = await client.send(command);
    const instanceIdList = StartingInstances.map(
      (instance) => ` • ${instance.InstanceId}`,
    );
    console.log("Starting instances:");
    console.log(instanceIdList.join("\n"));
  } catch (err) {
    console.error(err);
  }
};
```

- Untuk detail API, lihat [StartInstances](#) di Referensi AWS SDK for JavaScript API.

Kotlin

SDK untuk Kotlin

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
suspend fun startInstanceSc(instanceId: String) {
    val request =
        StartInstancesRequest {
            instanceIds = listOf(instanceId)
        }

    Ec2Client { region = "us-west-2" }.use { ec2 ->
        ec2.startInstances(request)
        println("Waiting until instance $instanceId starts. This will take a few
minutes.")
        ec2.waitForInstanceRunning {
            // suspend call
            instanceIds = listOf(instanceId)
        }
        println("Successfully started instance $instanceId")
    }
}
```

- Untuk detail API, lihat [StartInstances](#) di AWS SDK untuk referensi API Kotlin.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini memulai instance yang ditentukan.

```
Start-EC2Instance -InstanceId i-12345678
```

Output:

CurrentState	InstanceId	PreviousState
-----	-----	-----
Amazon.EC2.Model.InstanceState	i-12345678	Amazon.EC2.Model.InstanceState

Contoh 2: Contoh ini memulai instance yang ditentukan.

```
@("i-12345678", "i-76543210") | Start-EC2Instance
```

Contoh 3: Contoh ini memulai kumpulan instance yang saat ini dihentikan. Objek Instance yang dikembalikan oleh Get-EC2Instance disalurkan ke Start-EC2Instance. Sintaks yang digunakan oleh contoh ini membutuhkan PowerShell versi 3 atau lebih tinggi.

```
(Get-EC2Instance -Filter @{ Name="instance-state-name";
  Values="stopped"}).Instances | Start-EC2Instance
```

Contoh 4: Dengan PowerShell versi 2, Anda harus menggunakan New-Object untuk membuat filter untuk parameter Filter.

```
$filter = New-Object Amazon.EC2.Model.Filter
$filter.Name = "instance-state-name"
$filter.Values = "stopped"

(Get-EC2Instance -Filter $filter).Instances | Start-EC2Instance
```

- Untuk detail API, lihat [StartInstances](#) di Referensi AWS Tools for PowerShell Cmdlet.

Python

SDK untuk Python (Boto3)

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
class InstanceWrapper:
    """Encapsulates Amazon Elastic Compute Cloud (Amazon EC2) instance
    actions."""
```

```
def __init__(self, ec2_resource, instance=None):
    """
    :param ec2_resource: A Boto3 Amazon EC2 resource. This high-level
resource
                           is used to create additional high-level objects
                           that wrap low-level Amazon EC2 service actions.
    :param instance: A Boto3 Instance object. This is a high-level object
that
                           wraps instance actions.
    """
    self.ec2_resource = ec2_resource
    self.instance = instance

    @classmethod
    def from_resource(cls):
        ec2_resource = boto3.resource("ec2")
        return cls(ec2_resource)

    def start(self):
        """
        Starts an instance and waits for it to be in a running state.

        :return: The response to the start request.
        """
        if self.instance is None:
            logger.info("No instance to start.")
            return

        try:
            response = self.instance.start()
            self.instance.wait_until_running()
        except ClientError as err:
            logger.error(
                "Couldn't start instance %s. Here's why: %s: %s",
                self.instance.id,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
        else:
            return response
```

- Untuk detail API, lihat [StartInstances](#) di AWS SDK for Python (Boto3) Referensi API.

Ruby

SDK untuk Ruby

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
require "aws-sdk-ec2"

# Attempts to start an Amazon Elastic Compute Cloud (Amazon EC2) instance.
#
# Prerequisites:
#
# - The Amazon EC2 instance.
#
# @param ec2_client [Aws::EC2::Client] An initialized EC2 client.
# @param instance_id [String] The ID of the instance.
# @return [Boolean] true if the instance was started; otherwise, false.
# @example
#   exit 1 unless instance_started?(
#     Aws::EC2::Client.new(region: 'us-west-2'),
#     'i-123abc'
#   )
def instance_started?(ec2_client, instance_id)
  response = ec2_client.describe_instance_status(instance_ids: [instance_id])

  if response.instance_statuses.count.positive?
    state = response.instance_statuses[0].instance_state.name
    case state
    when "pending"
      puts "Error starting instance: the instance is pending. Try again later."
      return false
    when "running"
      puts "The instance is already running."
    end
  end
end
```

```
    return true
  when "terminated"
    puts "Error starting instance: " \
      "the instance is terminated, so you cannot start it."
    return false
  end
end

ec2_client.start_instances(instance_ids: [instance_id])
ec2_client.wait_until(:instance_running, instance_ids: [instance_id])
puts "Instance started."
return true
rescue StandardError => e
  puts "Error starting instance: #{e.message}"
  return false
end

# Example usage:
def run_me
  instance_id = ""
  region = ""
  # Print usage information and then stop.
  if ARGV[0] == "--help" || ARGV[0] == "-h"
    puts "Usage:  ruby ec2-ruby-example-start-instance-i-123abc.rb " \
      "INSTANCE_ID REGION "
    # Replace us-west-2 with the AWS Region you're using for Amazon EC2.
    puts "Example: ruby ec2-ruby-example-start-instance-i-123abc.rb " \
      "i-123abc us-west-2"
    exit 1
  # If no values are specified at the command prompt, use these default values.
  # Replace us-west-2 with the AWS Region you're using for Amazon EC2.
  elsif ARGV.count.zero?
    instance_id = "i-123abc"
    region = "us-west-2"
  # Otherwise, use the values as specified at the command prompt.
  else
    instance_id = ARGV[0]
    region = ARGV[1]
  end

  ec2_client = Aws::EC2::Client.new(region: region)

  puts "Attempting to start instance '#{instance_id}' " \
    "(this might take a few minutes)..."
end
```



```

unless instance_started?(ec2_client, instance_id)
  puts "Could not start instance."
end
end

run_me if $PROGRAM_NAME == __FILE__

```

- Untuk detail API, lihat [StartInstances](#) di Referensi AWS SDK for Ruby API.

Rust

SDK untuk Rust

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```

async fn start_instance(client: &Client, id: &str) -> Result<(), Error> {
  // start_instance has no unique errors to handle.
  client.start_instances().instance_ids(id).send().await?;

  println!("Waiting for instance to be running");

  let wait_for_running = client
    .wait_until_instance_running()
    .instance_ids(id)
    .wait(Duration::from_secs(60))
    .await;

  match wait_for_running {
    Ok(_) => println!("Instance is running"),
    Err(err) => match err {
      WaiterError::ExceededMaxWait(exceeded) => {
        println!(
          "Exceeded max time waiting for instance to start. Exceeded
          {}s by {}s.",
          exceeded.max_wait().as_secs(),
          (exceeded.elapsed() - exceeded.max_wait()).as_secs()
        );
      }
    }
  }
}

```

```

        return Ok(());
    }
    _ => return Err(err.into()),
},
}

println!("Started instance.");

Ok(())
}

```

- Untuk detail API, lihat [StartInstances](#) referensi AWS SDK for Rust API.

SAP ABAP

SDK untuk SAP ABAP

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```

DATA lt_instance_ids TYPE /aws1/
cl_ec2instidstringlist_w=>tt_instanceidstringlist.
APPEND NEW /aws1/cl_ec2instidstringlist_w( iv_value = iv_instance_id ) TO
lt_instance_ids.

"Perform dry run"
TRY.
    " DryRun is set to true. This checks for the required permissions to
start the instance without actually making the request. "
    lo_ec2->startinstances(
        it_instanceids = lt_instance_ids
        iv_dryrun = abap_true
    ).
CATCH /aws1/cx_rt_service_generic INTO DATA(lo_exception).
    " If the error code returned is `DryRunOperation`, then you have the
required permissions to start this instance. "
    IF lo_exception->av_err_code = 'DryRunOperation'.

```

```

        MESSAGE 'Dry run to start instance completed.' TYPE 'I'.
        " DryRun is set to false to start instance. "
        oo_result = lo_ec2->startinstances(          " oo_result is returned
for testing purposes. "
            it_instanceids = lt_instance_ids
            iv_dryrun = abap_false
        ).
        MESSAGE 'Successfully started the EC2 instance.' TYPE 'I'.
        " If the error code returned is `UnauthorizedOperation`, then you don't
have the required permissions to start this instance. "
        ELSEIF lo_exception->av_err_code = 'UnauthorizedOperation'.
            MESSAGE 'Dry run to start instance failed. User does not have
permissions to start the instance.' TYPE 'E'.
        ELSE.
            DATA(lv_error) = |"{ lo_exception->av_err_code }" - { lo_exception-
>av_err_msg }|.
            MESSAGE lv_error TYPE 'E'.
        ENDIF.
    ENDMETHOD.
ENDCLASS.

```

- Untuk detail API, lihat [StartInstances](#) di AWS SDK untuk referensi SAP ABAP API.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **StopInstances** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `StopInstances`.

Contoh tindakan adalah kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Anda dapat melihat tindakan ini dalam konteks dalam contoh kode berikut:

- [Memulai instans](#)

.NET

AWS SDK for .NET

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
/// <summary>
/// Stop an EC2 instance.
/// </summary>
/// <param name="ec2InstanceId">The instance Id of the EC2 instance to
/// stop.</param>
/// <returns>Async task.</returns>
public async Task StopInstances(string ec2InstanceId)
{
    // In addition to the list of instance Ids, the
    // request can also include the following properties:
    //     Force      When true, forces the instances to
    //                 stop but you must check the integrity
    //                 of the file system. Not recommended on
    //                 Windows instances.
    //     Hibernate  When true, hibernates the instance if the
    //                 instance was enabled for hibernation when
    //                 it was launched.
    var request = new StopInstancesRequest
    {
        InstanceIds = new List<string> { ec2InstanceId },
    };

    var response = await _amazonEC2.StopInstancesAsync(request);

    if (response.StoppingInstances.Count > 0)
    {
        var instances = response.StoppingInstances;
        instances.ForEach(i =>
        {
            Console.WriteLine($"Successfully stopped the EC2 Instance " +
                $"with InstanceID: {i.InstanceId}.");
        });
    }
}
```

```
    }
}
```

- Untuk detail API, lihat [StopInstances](#) di Referensi AWS SDK for .NET API.

Bash

AWS CLI dengan skrip Bash

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
#####
# function ec2_stop_instances
#
# This function stops one or more Amazon Elastic Compute Cloud (Amazon EC2)
# instances.
#
# Parameters:
#     -i instance_id - The ID(s) of the instance(s) to stop (comma-separated).
#     -h - Display help.
#
# Returns:
#     0 - If successful.
#     1 - If it fails.
#####
function ec2_stop_instances() {
    local instance_ids
    local option OPTARG # Required to use getopt command in a function.

    # bashsupport disable=BP5008
    function usage() {
        echo "function ec2_stop_instances"
        echo "Stops one or more Amazon Elastic Compute Cloud (Amazon EC2) instances."
        echo "  -i instance_id - The ID(s) of the instance(s) to stop (comma-
separated)."
        echo "  -h - Display help."
        echo ""
    }
}
```

```

}

# Retrieve the calling parameters.
while getopts "i:h" option; do
  case "${option}" in
    i) instance_ids="${OPTARG}" ;;
    h)
      usage
      return 0
      ;;
    \?)
      echo "Invalid parameter"
      usage
      return 1
      ;;
  esac
done
export OPTIND=1

if [[ -z "$instance_ids" ]]; then
  errecho "ERROR: You must provide one or more instance IDs with the -i
parameter."
  usage
  return 1
fi

response=$(aws ec2 stop-instances \
  --instance-ids "${instance_ids}") || {
  aws_cli_error_log ${?}
  errecho "ERROR: AWS reports stop-instances operation failed with $response."
  return 1
}

return 0
}

```

Fungsi utilitas yang digunakan dalam contoh ini.

```

#####
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).

```

```
#####
function errecho() {
    printf "%s\n" "$*" 1>&2
}

#####
# function aws_cli_error_log()
#
# This function is used to log the error messages from the AWS CLI.
#
# The function expects the following argument:
#     $1 - The error code returned by the AWS CLI.
#
# Returns:
#     0: - Success.
#
#####
function aws_cli_error_log() {
    local err_code=$1
    errecho "Error code : $err_code"
    if [ "$err_code" == 1 ]; then
        errecho " One or more S3 transfers failed."
    elif [ "$err_code" == 2 ]; then
        errecho " Command line failed to parse."
    elif [ "$err_code" == 130 ]; then
        errecho " Process received SIGINT."
    elif [ "$err_code" == 252 ]; then
        errecho " Command syntax invalid."
    elif [ "$err_code" == 253 ]; then
        errecho " The system environment or configuration was invalid."
    elif [ "$err_code" == 254 ]; then
        errecho " The service returned an error."
    elif [ "$err_code" == 255 ]; then
        errecho " 255 is a catch-all error."
    fi


    return 0
}

```

- Untuk detail API, lihat [StopInstances](#) di Referensi AWS CLI Perintah.

C++

SDK untuk C++

 Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
Aws::EC2::EC2Client ec2Client(clientConfiguration);
Aws::EC2::Model::StopInstancesRequest request;
request.AddInstanceIds(instanceId);
request.SetDryRun(true);

auto dry_run_outcome = ec2Client.StopInstances(request);
if (dry_run_outcome.IsSuccess()) {
    std::cerr
        << "Failed dry run to stop instance. A dry run should trigger an
error."
        << std::endl;
    return false;
}
else if (dry_run_outcome.GetError().GetErrorType() !=
    Aws::EC2::EC2Errors::DRY_RUN_OPERATION) {
    std::cout << "Failed dry run to stop instance " << instanceId << ": "
        << dry_run_outcome.GetError().GetMessage() << std::endl;
    return false;
}

request.SetDryRun(false);
auto outcome = ec2Client.StopInstances(request);
if (!outcome.IsSuccess()) {
    std::cout << "Failed to stop instance " << instanceId << ": " <<
        outcome.GetError().GetMessage() << std::endl;
}
else {
    std::cout << "Successfully stopped instance " << instanceId <<
        std::endl;
}
}
```


- Untuk detail API, lihat [StopInstances](#) di Referensi AWS SDK for C++ API.

CLI

AWS CLI

Contoh 1: Untuk menghentikan instans Amazon EC2

Contoh `stop-instances` berikut menghentikan instans yang didukung Amazon EBS tertentu.

```
aws ec2 stop-instances \  
  --instance-ids i-1234567890abcdef0
```

Output:

```
{  
  "StoppingInstances": [  
    {  
      "InstanceId": "i-1234567890abcdef0",  
      "CurrentState": {  
        "Code": 64,  
        "Name": "stopping"  
      },  
      "PreviousState": {  
        "Code": 16,  
        "Name": "running"  
      }  
    }  
  ]  
}
```

Untuk informasi selengkapnya, lihat [Hentikan dan Mulai Instans Anda](#) di Panduan Pengguna Amazon Elastic Compute Cloud.

Contoh 2: Untuk menghibernasi instans Amazon EC2

Contoh `stop-instances` berikut menghibernasi instans yang didukung Amazon EBS jika instans diaktifkan untuk hibernasi dan memenuhi prasyarat hibernasi. Setelah instans dimasukkan ke hibernasi, instans berhenti.

```
aws ec2 stop-instances \  
  --instance-ids i-1234567890abcdef0 \  
  --hibernate
```

Output:

```
{  
  "StoppingInstances": [  
    {  
      "CurrentState": {  
        "Code": 64,  
        "Name": "stopping"  
      },  
      "InstanceId": "i-1234567890abcdef0",  
      "PreviousState": {  
        "Code": 16,  
        "Name": "running"  
      }  
    }  
  ]  
}
```

Untuk informasi selengkapnya, lihat [Hibernasikan instans Linux Sesuai Permintaan Anda](#) di Panduan Pengguna Amazon Elastic Compute Cloud.

- Untuk detail API, lihat [StopInstances](#) di Referensi AWS CLI Perintah.

Java

SDK untuk Java 2.x

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
public static void stopInstance(Ec2Client ec2, String instanceId) {  
    Ec2Waiter ec2Waiter = Ec2Waiter.builder()  
        .overrideConfiguration(b -> b.maxAttempts(100))
```

```
        .client(ec2)
        .build();
    StopInstancesRequest request = StopInstancesRequest.builder()
        .instanceIds(instanceId)
        .build();

    System.out.println("Use an Ec2Waiter to wait for the instance to stop.
    This will take a few minutes.");
    ec2.stopInstances(request);
    DescribeInstancesRequest instanceRequest =
    DescribeInstancesRequest.builder()
        .instanceIds(instanceId)
        .build();

    WaiterResponse<DescribeInstancesResponse> waiterResponse =
    ec2Waiter.waitUntilInstanceStopped(instanceRequest);
    waiterResponse.matched().response().ifPresent(System.out::println);
    System.out.println("Successfully stopped instance " + instanceId);
    }
}
```

- Untuk detail API, lihat [StopInstances](#) di Referensi AWS SDK for Java 2.x API.

JavaScript

SDK untuk JavaScript (v3)

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
import { StopInstancesCommand } from "@aws-sdk/client-ec2";

import { client } from "../libs/client.js";

export const main = async () => {
    const command = new StopInstancesCommand({
        // Use DescribeInstancesCommand to find InstanceIds
        InstanceIds: ["INSTANCE_ID"],
    });
};
```

```
try {
  const { StoppingInstances } = await client.send(command);
  const instanceIdList = StoppingInstances.map(
    (instance) => ` • ${instance.InstanceId}`,
  );
  console.log("Stopping instances:");
  console.log(instanceIdList.join("\n"));
} catch (err) {
  console.error(err);
}
};
```

- Untuk detail API, lihat [StopInstances](#) di Referensi AWS SDK for JavaScript API.

Kotlin

SDK untuk Kotlin

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
suspend fun stopInstanceSc(instanceId: String) {
  val request =
    StopInstancesRequest {
      instanceIds = listOf(instanceId)
    }

  Ec2Client { region = "us-west-2" }.use { ec2 ->
    ec2.stopInstances(request)
    println("Waiting until instance $instanceId stops. This will take a few
minutes.")
    ec2.waitUntilInstanceStopped {
      // suspend call
      instanceIds = listOf(instanceId)
    }
    println("Successfully stopped instance $instanceId")
  }
}
```

```
}

```

- Untuk detail API, lihat [StopInstances](#) di AWS SDK untuk referensi API Kotlin.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini menghentikan instance yang ditentukan.

```
Stop-EC2Instance -InstanceId i-12345678
```

Output:

CurrentState	InstanceId	PreviousState
-----	-----	-----
Amazon.EC2.Model.InstanceState	i-12345678	Amazon.EC2.Model.InstanceState

- Untuk detail API, lihat [StopInstances](#) di Referensi AWS Tools for PowerShell Cmdlet.

Python

SDK untuk Python (Boto3)

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
class InstanceWrapper:
    """Encapsulates Amazon Elastic Compute Cloud (Amazon EC2) instance
    actions."""

    def __init__(self, ec2_resource, instance=None):
        """
        :param ec2_resource: A Boto3 Amazon EC2 resource. This high-level
        resource
```

```
        is used to create additional high-level objects
        that wrap low-level Amazon EC2 service actions.
:param instance: A Boto3 Instance object. This is a high-level object
that
        wraps instance actions.
"""
self.ec2_resource = ec2_resource
self.instance = instance

@classmethod
def from_resource(cls):
    ec2_resource = boto3.resource("ec2")
    return cls(ec2_resource)

def stop(self):
    """
    Stops an instance and waits for it to be in a stopped state.

    :return: The response to the stop request.
    """
    if self.instance is None:
        logger.info("No instance to stop.")
        return

    try:
        response = self.instance.stop()
        self.instance.wait_until_stopped()
    except ClientError as err:
        logger.error(
            "Couldn't stop instance %s. Here's why: %s: %s",
            self.instance.id,
            err.response["Error"]["Code"],
            err.response["Error"]["Message"],
        )
        raise
    else:
        return response
```

- Untuk detail API, lihat [StopInstances](#) di AWS SDK for Python (Boto3) Referensi API.

Ruby

SDK untuk Ruby

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
require "aws-sdk-ec2"

# Prerequisites:
#
# - The Amazon EC2 instance.
#
# @param ec2_client [Aws::EC2::Client] An initialized EC2 client.
# @param instance_id [String] The ID of the instance.
# @return [Boolean] true if the instance was stopped; otherwise, false.
# @example
#   exit 1 unless instance_stopped?(
#     Aws::EC2::Client.new(region: 'us-west-2'),
#     'i-123abc'
#   )
def instance_stopped?(ec2_client, instance_id)
  response = ec2_client.describe_instance_status(instance_ids: [instance_id])

  if response.instance_statuses.count.positive?
    state = response.instance_statuses[0].instance_state.name
    case state
    when "stopping"
      puts "The instance is already stopping."
      return true
    when "stopped"
      puts "The instance is already stopped."
      return true
    when "terminated"
      puts "Error stopping instance: " \
        "the instance is terminated, so you cannot stop it."
      return false
    end
  end
end
```

```
end

ec2_client.stop_instances(instance_ids: [instance_id])
ec2_client.wait_until(:instance_stopped, instance_ids: [instance_id])
puts "Instance stopped."
return true
rescue StandardError => e
  puts "Error stopping instance: #{e.message}"
  return false
end

# Example usage:
def run_me
  instance_id = ""
  region = ""
  # Print usage information and then stop.
  if ARGV[0] == "--help" || ARGV[0] == "-h"
    puts "Usage:  ruby ec2-ruby-example-stop-instance-i-123abc.rb " \
      "INSTANCE_ID REGION "
    # Replace us-west-2 with the AWS Region you're using for Amazon EC2.
    puts "Example: ruby ec2-ruby-example-start-instance-i-123abc.rb " \
      "i-123abc us-west-2"
    exit 1
  # If no values are specified at the command prompt, use these default values.
  # Replace us-west-2 with the AWS Region you're using for Amazon EC2.
  elsif ARGV.count.zero?
    instance_id = "i-123abc"
    region = "us-west-2"
  # Otherwise, use the values as specified at the command prompt.
  else
    instance_id = ARGV[0]
    region = ARGV[1]
  end

  ec2_client = Aws::EC2::Client.new(region: region)

  puts "Attempting to stop instance '#{instance_id}' " \
    "(this might take a few minutes)..."
  unless instance_stopped?(ec2_client, instance_id)
    puts "Could not stop instance."
  end
end

run_me if $PROGRAM_NAME == __FILE__
```


- Untuk detail API, lihat [StopInstances](#) di Referensi AWS SDK for Ruby API.

Rust

SDK untuk Rust

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
async fn stop_instance(client: &Client, id: &str) -> Result<(), Error> {
    client.stop_instances().instance_ids(id).send().await?;

    println!("Stopping instance...");

    let wait = client
        .wait_until_instance_stopped()
        .instance_ids(id)
        .wait(Duration::from_secs(60))
        .await;

    match wait {
        Ok(_) => {
            println!("Stopped instance.");
        }
        Err(err) => match err {
            WaiterError::ExceededMaxWait(exceeded) => {
                println!(
                    "Exceeded max time waiting for instance to stop. Exceeded {}s
by {}s",
                    exceeded.max_wait().as_secs(),
                    (exceeded.elapsed() - exceeded.max_wait()).as_secs()
                );
            }
            _ => return Err(err.into()),
        },
    };
    Ok(())
}
```

```
}

```

- Untuk detail API, lihat [StopInstances](#) referensi AWS SDK for Rust API.

SAP ABAP

SDK untuk SAP ABAP

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```

DATA lt_instance_ids TYPE /aws1/
cl_ec2instidstringlist_w=>tt_instanceidstringlist.
  APPEND NEW /aws1/cl_ec2instidstringlist_w( iv_value = iv_instance_id ) TO
  lt_instance_ids.

  "Perform dry run"
  TRY.
    " DryRun is set to true. This checks for the required permissions to stop
    the instance without actually making the request. "
    lo_ec2->stopinstances(
      it_instanceids = lt_instance_ids
      iv_dryrun = abap_true
    ).
  CATCH /aws1/cx_rt_service_generic INTO DATA(lo_exception).
    " If the error code returned is `DryRunOperation`, then you have the
    required permissions to stop this instance. "
    IF lo_exception->av_err_code = 'DryRunOperation'.
      MESSAGE 'Dry run to stop instance completed.' TYPE 'I'.
      " DryRun is set to false to stop instance. "
      oo_result = lo_ec2->stopinstances(           " oo_result is returned
for testing purposes. "
      it_instanceids = lt_instance_ids
      iv_dryrun = abap_false
    ).
      MESSAGE 'Successfully stopped the EC2 instance.' TYPE 'I'.

```

```

    " If the error code returned is `UnauthorizedOperation`, then you don't
    have the required permissions to stop this instance. "
    ELSEIF lo_exception->av_err_code = 'UnauthorizedOperation'.
        MESSAGE 'Dry run to stop instance failed. User does not have
        permissions to stop the instance.' TYPE 'E'.
    ELSE.
        DATA(lv_error) = |"{ lo_exception->av_err_code }" - { lo_exception-
        >av_err_msg }|.
        MESSAGE lv_error TYPE 'E'.
    ENDIF.
ENDTRY.

```

- Untuk detail API, lihat [StopInstances](#) di AWS SDK untuk referensi SAP ABAP API.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **TerminateInstances** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `TerminateInstances`.

Contoh tindakan adalah kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Anda dapat melihat tindakan ini dalam konteks dalam contoh kode berikut:

- [Memulai instans](#)

.NET

AWS SDK for .NET

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```

/// <summary>
/// Terminate an EC2 instance.

```

```

    /// </summary>
    /// <param name="ec2InstanceId">The instance Id of the EC2 instance
    /// to terminate.</param>
    /// <returns>Async task.</returns>
    public async Task<List<InstanceStateChange>> TerminateInstances(string
ec2InstanceId)
    {
        var request = new TerminateInstancesRequest
        {
            InstanceIds = new List<string> { ec2InstanceId }
        };

        var response = await _amazonEC2.TerminateInstancesAsync(request);
        return response.TerminatingInstances;
    }

```

- Untuk detail API, lihat [TerminateInstances](#) di Referensi AWS SDK for .NET API.

Bash

AWS CLI dengan skrip Bash

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```

#####
# function ec2_terminate_instances
#
# This function terminates one or more Amazon Elastic Compute Cloud (Amazon EC2)
# instances using the AWS CLI.
#
# Parameters:
#     -i instance_ids - A space-separated list of instance IDs.
#     -h - Display help.
#
# Returns:
#     0 - If successful.
#     1 - If it fails.

```

```
#####
function ec2_terminate_instances() {
    local instance_ids response
    local option OPTARG # Required to use getopt command in a function.

    # bashsupport disable=BP5008
    function usage() {
        echo "function ec2_terminate_instances"
        echo "Terminates one or more Amazon Elastic Compute Cloud (Amazon EC2)
instances."
        echo "  -i instance_ids - A space-separated list of instance IDs."
        echo "  -h - Display help."
        echo ""
    }

    # Retrieve the calling parameters.
    while getopt "i:h" option; do
        case "${option}" in
            i) instance_ids="${OPTARG}" ;;
            h)
                usage
                return 0
                ;;
            \?)
                echo "Invalid parameter"
                usage
                return 1
                ;;
        esac
    done
    export OPTIND=1

    # Check if instance ID is provided
    if [[ -z "${instance_ids}" ]]; then
        echo "Error: Missing required instance IDs parameter."
        usage
        return 1
    fi

    # shellcheck disable=SC2086
    response=$(aws ec2 terminate-instances \
        "--instance-ids" $instance_ids \
        "--query 'TerminatingInstances[*].[InstanceId,CurrentState.Name]' \
        "--output text) || {
```

```

aws_cli_error_log ${?}
errecho "ERROR: AWS reports terminate-instances operation failed.$response"
return 1
}

return 0
}

```

Fungsi utilitas yang digunakan dalam contoh ini.

```

#####
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
#####
function errecho() {
    printf "%s\n" "$*" 1>&2
}

#####
# function aws_cli_error_log()
#
# This function is used to log the error messages from the AWS CLI.
#
# The function expects the following argument:
#     $1 - The error code returned by the AWS CLI.
#
# Returns:
#     0: - Success.
#
#####
function aws_cli_error_log() {
    local err_code=$1
    errecho "Error code : $err_code"
    if [ "$err_code" == 1 ]; then
        errecho " One or more S3 transfers failed."
    elif [ "$err_code" == 2 ]; then
        errecho " Command line failed to parse."
    elif [ "$err_code" == 130 ]; then
        errecho " Process received SIGINT."
    elif [ "$err_code" == 252 ]; then
        errecho " Command syntax invalid."
    fi
}

```

```
elif [ "$err_code" == 253 ]; then
    errecho " The system environment or configuration was invalid."
elif [ "$err_code" == 254 ]; then
    errecho " The service returned an error."
elif [ "$err_code" == 255 ]; then
    errecho " 255 is a catch-all error."
fi

return 0
}
```

- Untuk detail API, lihat [TerminateInstances](#) di Referensi AWS CLI Perintah.

C++

SDK untuk C++

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
Aws::EC2::EC2Client ec2Client(clientConfiguration);

Aws::EC2::Model::TerminateInstancesRequest request;
request.SetInstanceIds({instanceID});

Aws::EC2::Model::TerminateInstancesOutcome outcome =
    ec2Client.TerminateInstances(request);
if (outcome.IsSuccess()) {
    std::cout << "Ec2 instance '" << instanceID <<
        "' was terminated." << std::endl;
}
else {
    std::cerr << "Failed to terminate ec2 instance " << instanceID <<
        ", " <<
        outcome.GetError().GetMessage() << std::endl;
    return false;
}
```

- Untuk detail API, lihat [TerminateInstances](#) di Referensi AWS SDK for C++ API.

CLI

AWS CLI

Untuk mengakhiri instans Amazon EC2

Contoh ini mengakhiri instans tertentu.

Perintah:

```
aws ec2 terminate-instances --instance-ids i-1234567890abcdef0
```

Output:


```
{
  "TerminatingInstances": [
    {
      "InstanceId": "i-1234567890abcdef0",
      "CurrentState": {
        "Code": 32,
        "Name": "shutting-down"
      },
      "PreviousState": {
        "Code": 16,
        "Name": "running"
      }
    }
  ]
}
```

Untuk informasi selengkapnya, lihat Menggunakan Instans Amazon EC2 di Panduan Pengguna Antarmuka Baris Perintah AWS .

- Untuk detail API, lihat [TerminateInstances](#) di Referensi AWS CLI Perintah.

Java

SDK untuk Java 2.x

 Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
public static void terminateEC2(Ec2Client ec2, String instanceId) {
    try {
        Ec2Waiter ec2Waiter = Ec2Waiter.builder()
            .overrideConfiguration(b -> b.maxAttempts(100))
            .client(ec2)
            .build();

        TerminateInstancesRequest ti = TerminateInstancesRequest.builder()
            .instanceIds(instanceId)
            .build();

        System.out.println("Use an Ec2Waiter to wait for the instance to
        terminate. This will take a few minutes.");
        ec2.terminateInstances(ti);
        DescribeInstancesRequest instanceRequest =
        DescribeInstancesRequest.builder()
            .instanceIds(instanceId)
            .build();

        WaiterResponse<DescribeInstancesResponse> waiterResponse = ec2Waiter
            .waitUntilInstanceTerminated(instanceRequest);
        waiterResponse.matched().response().ifPresent(System.out::println);
        System.out.println("Successfully started instance " + instanceId);
        System.out.println(instanceId + " is terminated!");

    } catch (Ec2Exception e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
}
```

- Untuk detail API, lihat [TerminateInstances](#) di Referensi AWS SDK for Java 2.x API.

JavaScript

SDK untuk JavaScript (v3)

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
import { TerminateInstancesCommand } from "@aws-sdk/client-ec2";

import { client } from "../libs/client.js";

export const main = async () => {
  const command = new TerminateInstancesCommand({
    InstanceIds: ["INSTANCE_ID"],
  });

  try {
    const { TerminatingInstances } = await client.send(command);
    const instanceList = TerminatingInstances.map(
      (instance) => ` • ${instance.InstanceId}`,
    );
    console.log("Terminating instances:");
    console.log(instanceList.join("\n"));
  } catch (err) {
    console.error(err);
  }
};
```

- Untuk detail API, lihat [TerminateInstances](#) di Referensi AWS SDK for JavaScript API.

Kotlin

SDK untuk Kotlin

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
suspend fun terminateEC2(instanceID: String) {
    val request =
        TerminateInstancesRequest {
            instanceIds = listOf(instanceID)
        }

    Ec2Client { region = "us-west-2" }.use { ec2 ->
        val response = ec2.terminateInstances(request)
        response.terminatingInstances?.forEach { instance ->
            println("The ID of the terminated instance is
                ${instance.instanceId}")
        }
    }
}
```

- Untuk detail API, lihat [TerminateInstances](#) di AWS SDK untuk referensi API Kotlin.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini mengakhiri instance yang ditentukan (instance mungkin berjalan atau dalam keadaan 'berhenti'). Cmdlet akan meminta konfirmasi sebelum melanjutkan; gunakan sakelar `-Force` untuk menekan prompt.

```
Remove-EC2Instance -InstanceId i-12345678
```

Output:

CurrentState	InstanceId	PreviousState
--------------	------------	---------------

```

-----
Amazon.EC2.Model.InstanceState    i-12345678    Amazon.EC2.Model.InstanceState

```

- Untuk detail API, lihat [TerminateInstances](#) di Referensi AWS Tools for PowerShell Cmdlet.

Python

SDK untuk Python (Boto3)

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```

class InstanceWrapper:
    """Encapsulates Amazon Elastic Compute Cloud (Amazon EC2) instance
    actions."""

    def __init__(self, ec2_resource, instance=None):
        """
        :param ec2_resource: A Boto3 Amazon EC2 resource. This high-level
        resource
                               is used to create additional high-level objects
                               that wrap low-level Amazon EC2 service actions.
        :param instance: A Boto3 Instance object. This is a high-level object
        that
                               wraps instance actions.
        """
        self.ec2_resource = ec2_resource
        self.instance = instance

    @classmethod
    def from_resource(cls):
        ec2_resource = boto3.resource("ec2")
        return cls(ec2_resource)

    def terminate(self):
        """
        Terminates an instance and waits for it to be in a terminated state.
        """

```

```
if self.instance is None:
    logger.info("No instance to terminate.")
    return

instance_id = self.instance.id
try:
    self.instance.terminate()
    self.instance.wait_until_terminated()
    self.instance = None
except ClientError as err:
    logging.error(
        "Couldn't terminate instance %s. Here's why: %s: %s",
        instance_id,
        err.response["Error"]["Code"],
        err.response["Error"]["Message"],
    )
    raise
```

- Untuk detail API, lihat [TerminateInstances](#) di AWS SDK for Python (Boto3) Referensi API.

Ruby

SDK untuk Ruby

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
require "aws-sdk-ec2"

# Prerequisites:
#
# - The Amazon EC2 instance.
#
# @param ec2_client [Aws::EC2::Client] An initialized EC2 client.
# @param instance_id [String] The ID of the instance.
# @return [Boolean] true if the instance was terminated; otherwise, false.
```

```
# @example
#   exit 1 unless instance_terminated?(
#     Aws::EC2::Client.new(region: 'us-west-2'),
#     'i-123abc'
#   )
def instance_terminated?(ec2_client, instance_id)
  response = ec2_client.describe_instance_status(instance_ids: [instance_id])

  if response.instance_statuses.count.positive? &&
    response.instance_statuses[0].instance_state.name == "terminated"

    puts "The instance is already terminated."
    return true
  end

  ec2_client.terminate_instances(instance_ids: [instance_id])
  ec2_client.wait_until(:instance_terminated, instance_ids: [instance_id])
  puts "Instance terminated."
  return true
rescue StandardError => e
  puts "Error terminating instance: #{e.message}"
  return false
end

# Example usage:
def run_me
  instance_id = ""
  region = ""
  # Print usage information and then stop.
  if ARGV[0] == "--help" || ARGV[0] == "-h"
    puts "Usage:  ruby ec2-ruby-example-terminate-instance-i-123abc.rb " \
      "INSTANCE_ID REGION "
    # Replace us-west-2 with the AWS Region you're using for Amazon EC2.
    puts "Example: ruby ec2-ruby-example-terminate-instance-i-123abc.rb " \
      "i-123abc us-west-2"
    exit 1
  # If no values are specified at the command prompt, use these default values.
  # Replace us-west-2 with the AWS Region you're using for Amazon EC2.
  elsif ARGV.count.zero?
    instance_id = "i-123abc"
    region = "us-west-2"
  # Otherwise, use the values as specified at the command prompt.
  else
    instance_id = ARGV[0]
```

```
    region = ARGV[1]
  end

  ec2_client = Aws::EC2::Client.new(region: region)

  puts "Attempting to terminate instance '#{instance_id}' " \
    "(this might take a few minutes)..."
  unless instance_terminated?(ec2_client, instance_id)
    puts "Could not terminate instance."
  end
end
end

run_me if $PROGRAM_NAME == __FILE__
```

- Untuk detail API, lihat [TerminateInstances](#) di Referensi AWS SDK for Ruby API.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **UnassignPrivateIpAddresses** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `UnassignPrivateIpAddresses`.

CLI

AWS CLI

Untuk membatalkan penetapan alamat IP pribadi sekunder dari antarmuka jaringan

Contoh ini membatalkan penetapan alamat IP pribadi yang ditentukan dari antarmuka jaringan yang ditentukan. Jika perintah berhasil, tidak ada output yang akan ditampilkan.

Perintah:

```
aws ec2 unassign-private-ip-addresses --network-interface-id eni-e5aa89a3 --
private-ip-addresses 10.0.0.82
```

- Untuk detail API, lihat [UnassignPrivateIpAddresses](#) di Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini membatalkan penetapan alamat IP pribadi yang ditentukan dari antarmuka jaringan yang ditentukan.

```
Unregister-EC2PrivateIpAddress -NetworkInterfaceId eni-1a2b3c4d -PrivateIpAddress 10.0.0.82
```

- Untuk detail API, lihat [UnassignPrivateIpAddresses](#) di Referensi AWS Tools for PowerShell Cmdlet.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **UnmonitorInstances** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `UnmonitorInstances`.

Contoh tindakan adalah kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Anda dapat melihat tindakan ini dalam konteks dalam contoh kode berikut:

- [Memulai instans](#)

C++

SDK untuk C++

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
Aws::EC2::EC2Client ec2Client(clientConfiguration);  
Aws::EC2::Model::UnmonitorInstancesRequest unrequest;  
unrequest.AddInstanceIds(instanceId);  
unrequest.SetDryRun(true);
```



```

auto undryRunOutcome = ec2Client.UnmonitorInstances(unrequest);
if (undryRunOutcome.IsSuccess()) {
    std::cerr
        << "Failed dry run to disable monitoring on instance. A dry run
should trigger an error."
        <<
        std::endl;
    return false;
}
else if (undryRunOutcome.GetError().GetErrorType() !=
    Aws::EC2::EC2Errors::DRY_RUN_OPERATION) {
    std::cout << "Failed dry run to disable monitoring on instance " <<
        instanceId << ": " << undryRunOutcome.GetError().GetMessage()
<<
        std::endl;
    return false;
}

unrequest.SetDryRun(false);
auto unmonitorInstancesOutcome = ec2Client.UnmonitorInstances(unrequest);
if (!unmonitorInstancesOutcome.IsSuccess()) {
    std::cout << "Failed to disable monitoring on instance " << instanceId
        << ": " << unmonitorInstancesOutcome.GetError().GetMessage() <<
        std::endl;
}
else {
    std::cout << "Successfully disable monitoring on instance " <<
        instanceId << std::endl;
}
}

```

- Untuk detail API, lihat [UnmonitorInstances](#) di Referensi AWS SDK for C++ API.

CLI

AWS CLI

Untuk menonaktifkan pemantauan yang mendetail untuk instans

Perintah contoh ini menonaktifkan pemantauan yang mendetail untuk instans tertentu.

Perintah:

```
aws ec2 unmonitor-instances --instance-ids i-1234567890abcdef0
```

Output:

```
{
  "InstanceMonitorings": [
    {
      "InstanceId": "i-1234567890abcdef0",
      "Monitoring": {
        "State": "disabling"
      }
    }
  ]
}
```

- Untuk detail API, lihat [UnmonitorInstances](#) di Referensi AWS CLI Perintah.

JavaScript

SDK untuk JavaScript (v3)

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
import { UnmonitorInstancesCommand } from "@aws-sdk/client-ec2";

import { client } from "../libs/client.js";

export const main = async () => {
  const command = new UnmonitorInstancesCommand({
    InstanceIds: ["i-09a3dfe7ae00e853f"],
  });

  try {
    const { InstanceMonitorings } = await client.send(command);
    const instanceMonitoringsList = InstanceMonitorings.map(
      (im) =>
```

```

    ` • Detailed monitoring state for ${im.InstanceId} is
    ${im.Monitoring.State}.`,
    );
    console.log("Monitoring status:");
    console.log(instanceMonitoringsList.join("\n"));
  } catch (err) {
    console.error(err);
  }
};

```

- Untuk detail API, lihat [UnmonitorInstances](#) di Referensi AWS SDK for JavaScript API.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini menonaktifkan pemantauan terperinci untuk contoh yang ditentukan.

```
Stop-EC2InstanceMonitoring -InstanceId i-12345678
```

Output:

InstanceId	Monitoring
-----	-----
i-12345678	Amazon.EC2.Model.Monitoring

- Untuk detail API, lihat [UnmonitorInstances](#) di Referensi AWS Tools for PowerShell Cmdlet.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Skenario untuk Amazon EC2 menggunakan SDK AWS

Contoh kode berikut menunjukkan cara menerapkan skenario umum di Amazon EC2 dengan AWS SDK. Skenario-skenario ini menunjukkan cara menyelesaikan tugas tertentu dengan memanggil beberapa fungsi di dalam Amazon EC2. Setiap skenario menyertakan tautan ke GitHub, di mana Anda dapat menemukan petunjuk tentang cara mengatur dan menjalankan kode.

Contoh

- [Membangun dan mengelola layanan tangguh menggunakan SDK AWS](#)
- [Memulai instans Amazon EC2 menggunakan SDK AWS](#)

Membangun dan mengelola layanan tangguh menggunakan SDK AWS

Contoh kode berikut menunjukkan cara membuat layanan web dengan beban seimbang yang mengembalikan rekomendasi buku, film, dan lagu. Contoh ini menunjukkan cara layanan tersebut merespons kegagalan, serta cara merestrukturisasi layanan agar lebih tangguh ketika terjadi kegagalan.

- Menggunakan grup Amazon EC2 Auto Scaling untuk membuat instans Amazon Elastic Compute Cloud (Amazon EC2) berdasarkan templat peluncuran dan menyimpan sejumlah instans dalam rentang yang ditentukan.
- Menangani dan mendistribusikan permintaan HTTP dengan Elastic Load Balancing.
- Memantau kondisi instans dalam grup Auto Scaling dan meneruskan permintaan hanya ke instans yang sehat.
- Menjalankan server web Python pada setiap instans EC2 untuk menangani permintaan HTTP. Server web merespons dengan memberikan rekomendasi dan melakukan pemeriksaan kondisi.
- Menyimulasikan layanan yang direkomendasikan dengan tabel Amazon DynamoDB.
- Kontrol respons server web terhadap permintaan dan pemeriksaan kesehatan dengan memperbarui AWS Systems Manager parameter.

.NET

AWS SDK for .NET

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturan dan menjalankannya di [Repositori Contoh Kode AWS](#).

Menjalankan skenario interaktif di prompt perintah.

```
static async Task Main(string[] args)
```

```
{
    _configuration = new ConfigurationBuilder()
        .SetBasePath(Directory.GetCurrentDirectory())
        .AddJsonFile("settings.json") // Load settings from .json file.
        .AddJsonFile("settings.local.json",
            true) // Optionally, load local settings.
        .Build();

    // Set up dependency injection for the AWS services.
    using var host = Host.CreateDefaultBuilder(args)
        .ConfigureLogging(logging =>
            logging.AddFilter("System", LogLevel.Debug)
                .AddFilter<DebugLoggerProvider>("Microsoft",
                    LogLevel.Information)
                .AddFilter<ConsoleLoggerProvider>("Microsoft",
                    LogLevel.Trace))
        .ConfigureServices((_, services) =>
            services.AddAWSService<IAmazonIdentityManagementService>()
                .AddAWSService<IAmazonDynamoDB>()
                .AddAWSService<IAmazonElasticLoadBalancingV2>()
                .AddAWSService<IAmazonSimpleSystemsManagement>()
                .AddAWSService<IAmazonAutoScaling>()
                .AddAWSService<IAmazonEC2>()
                .AddTransient<AutoScalerWrapper>()
                .AddTransient<ElasticLoadBalancerWrapper>()
                .AddTransient<SmParameterWrapper>()
                .AddTransient<Recommendations>()
                .AddSingleton<IConfiguration>(_configuration)
            )
        .Build();

    ServicesSetup(host);
    ResourcesSetup();

    try
    {
        Console.WriteLine(new string('-', 80));
        Console.WriteLine("Welcome to the Resilient Architecture Example
Scenario.");
        Console.WriteLine(new string('-', 80));
        await Deploy(true);

        Console.WriteLine("Now let's begin the scenario.");
    }
}
```

```
        Console.WriteLine(new string('-', 80));
        await Demo(true);

        Console.WriteLine(new string('-', 80));
        Console.WriteLine("Finally, let's clean up our resources.");
        Console.WriteLine(new string('-', 80));

        await DestroyResources(true);

        Console.WriteLine(new string('-', 80));
        Console.WriteLine("Resilient Architecture Example Scenario is
complete.");
        Console.WriteLine(new string('-', 80));
    }
    catch (Exception ex)
    {
        Console.WriteLine(new string('-', 80));
        Console.WriteLine($"There was a problem running the scenario:
{ex.Message}");
        await DestroyResources(true);
        Console.WriteLine(new string('-', 80));
    }
}

/// <summary>
/// Setup any common resources, also used for integration testing.
/// </summary>
public static void ResourcesSetup()
{
    _httpClient = new HttpClient();
}

/// <summary>
/// Populate the services for use within the console application.
/// </summary>
/// <param name="host">The services host.</param>
private static void ServicesSetup(IHost host)
{
    _elasticLoadBalancerWrapper =
host.Services.GetRequiredService<ElasticLoadBalancerWrapper>();
    _iamClient =
host.Services.GetRequiredService<IAmazonIdentityManagementService>();
    _recommendations = host.Services.GetRequiredService<Recommendations>();
```

```
        _autoScalerWrapper =
host.Services.GetRequiredService<AutoScalerWrapper>();
        _smParameterWrapper =
host.Services.GetRequiredService<SmParameterWrapper>();
    }

    /// <summary>
    /// Deploy necessary resources for the scenario.
    /// </summary>
    /// <param name="interactive">True to run as interactive.</param>
    /// <returns>True if successful.</returns>
    public static async Task<bool> Deploy(bool interactive)
    {
        var protocol = "HTTP";
        var port = 80;
        var sshPort = 22;

        Console.WriteLine(
            "\nFor this demo, we'll use the AWS SDK for .NET to create several
AWS resources\n" +
            "to set up a load-balanced web service endpoint and explore some ways
to make it resilient\n" +
            "against various kinds of failures.\n\n" +
            "Some of the resources create by this demo are:\n");

        Console.WriteLine(
            "\t* A DynamoDB table that the web service depends on to provide
book, movie, and song recommendations.");
        Console.WriteLine(
            "\t* An EC2 launch template that defines EC2 instances that each
contain a Python web server.");
        Console.WriteLine(
            "\t* An EC2 Auto Scaling group that manages EC2 instances across
several Availability Zones.");
        Console.WriteLine(
            "\t* An Elastic Load Balancing (ELB) load balancer that targets the
Auto Scaling group to distribute requests.");
        Console.WriteLine(new string('-', 80));
        Console.WriteLine("Press Enter when you're ready to start deploying
resources.");
        if (interactive)
            Console.ReadLine();

        // Create and populate the DynamoDB table.
```

```
var databaseTableName = _configuration["databaseName"];
var recommendationsPath = Path.Join(_configuration["resourcePath"],
    "recommendations_objects.json");
Console.WriteLine($"Creating and populating a DynamoDB table named
{databaseTableName}.");
await _recommendations.CreateDatabaseWithName(databaseTableName);
await _recommendations.PopulateDatabase(databaseTableName,
recommendationsPath);
Console.WriteLine(new string('-', 80));

// Create the EC2 Launch Template.

Console.WriteLine(
    $"Creating an EC2 launch template that runs
'server_startup_script.sh' when an instance starts.\n"
    + "\nThis script starts a Python web server defined in the
'server.py' script. The web server\n"
    + "listens to HTTP requests on port 80 and responds to requests to
 '/' and to '/healthcheck'.\n"
    + "For demo purposes, this server is run as the root user. In
production, the best practice is to\n"
    + "run a web server, such as Apache, with least-privileged
credentials.");
Console.WriteLine(
    "\nThe template also defines an IAM policy that each instance uses to
assume a role that grants\n"
    + "permissions to access the DynamoDB recommendation table and
Systems Manager parameters\n"
    + "that control the flow of the demo.");

var startupScriptPath = Path.Join(_configuration["resourcePath"],
    "server_startup_script.sh");
var instancePolicyPath = Path.Join(_configuration["resourcePath"],
    "instance_policy.json");
await _autoScalerWrapper.CreateTemplate(startupScriptPath,
instancePolicyPath);
Console.WriteLine(new string('-', 80));

Console.WriteLine(
    "Creating an EC2 Auto Scaling group that maintains three EC2
instances, each in a different\n"
    + "Availability Zone.\n");
var zones = await _autoScalerWrapper.DescribeAvailabilityZones();
```



```
        await _autoScalerWrapper.CreateGroupOfSize(3,
_autoScalerWrapper.GroupName, zones);
        Console.WriteLine(new string('-', 80));

        Console.WriteLine(
            "At this point, you have EC2 instances created. Once each instance
starts, it listens for\n"
            + "HTTP requests. You can see these instances in the console or
continue with the demo.\n");

        Console.WriteLine(new string('-', 80));
        Console.WriteLine("Press Enter when you're ready to continue.");
        if (interactive)
            Console.ReadLine();

        Console.WriteLine("Creating variables that control the flow of the
demo.");
        await _smParameterWrapper.Reset();

        Console.WriteLine(
            "\nCreating an Elastic Load Balancing target group and load balancer.
The target group\n"
            + "defines how the load balancer connects to instances. The load
balancer provides a\n"
            + "single endpoint where clients connect and dispatches requests to
instances in the group.");

        var defaultVpc = await _autoScalerWrapper.GetDefaultVpc();
        var subnets = await
_autoScalerWrapper.GetAllVpcSubnetsForZones(defaultVpc.VpcId, zones);
        var subnetIds = subnets.Select(s => s.SubnetId).ToList();
        var targetGroup = await
_elasticLoadBalancerWrapper.CreateTargetGroupOnVpc(_elasticLoadBalancerWrapper.TargetGroup
protocol, port, defaultVpc.VpcId);

        await
_elasticLoadBalancerWrapper.CreateLoadBalancerAndListener(_elasticLoadBalancerWrapper.Lo
subnetIds, targetGroup);
        await
_autoScalerWrapper.AttachLoadBalancerToGroup(_autoScalerWrapper.GroupName,
targetGroup.TargetGroupArn);
        Console.WriteLine("\nVerifying access to the load balancer endpoint...");
        var endPoint = await
_elasticLoadBalancerWrapper.GetEndpointForLoadBalancerByName(_elasticLoadBalancerWrapper
```

```
        var loadBalancerAccess = await
_elasticLoadBalancerWrapper.VerifyLoadBalancerEndpoint(endPoint);

        if (!loadBalancerAccess)
        {
            Console.WriteLine("\nCouldn't connect to the load balancer, verifying
that the port is open...");

            var ipString = await _httpClient.GetStringAsync("https://
checkip.amazonaws.com");
            ipString = ipString.Trim();

            var defaultSecurityGroup = await
_autoScalerWrapper.GetDefaultSecurityGroupForVpc(defaultVpc);
            var portIsOpen =
_autoScalerWrapper.VerifyInboundPortForGroup(defaultSecurityGroup, port,
ipString);
            var sshPortIsOpen =
_autoScalerWrapper.VerifyInboundPortForGroup(defaultSecurityGroup, sshPort,
ipString);

            if (!portIsOpen)
            {
                Console.WriteLine(
                    "\nFor this example to work, the default security group for
your default VPC must\n"
                    + "allows access from this computer. You can either add it
automatically from this\n"
                    + "example or add it yourself using the AWS Management
Console.\n");

                if (!interactive || GetYesNoResponse(
                    "Do you want to add a rule to the security group to allow
inbound traffic from your computer's IP address?"))
                {
                    await
_autoScalerWrapper.OpenInboundPort(defaultSecurityGroup.GroupId, port,
ipString);
                }
            }

            if (!sshPortIsOpen)
            {
                if (!interactive || GetYesNoResponse(
```

```
        "Do you want to add a rule to the security group to allow
inbound SSH traffic for debugging from your computer's IP address?"))
        {
            await
_autoScalerWrapper.OpenInboundPort(defaultSecurityGroup.GroupId, sshPort,
ipString);
        }
    }
    loadBalancerAccess = await
_elasticLoadBalancerWrapper.VerifyLoadBalancerEndpoint(endPoint);
}

if (loadBalancerAccess)
{
    Console.WriteLine("Your load balancer is ready. You can access it by
browsing to:");
    Console.WriteLine($"\\thttp://{endPoint}\\n");
}
else
{
    Console.WriteLine(
        "\\nCouldn't get a successful response from the load balancer
endpoint. Troubleshoot by\\n"
        + "manually verifying that your VPC and security group are
configured correctly and that\\n"
        + "you can successfully make a GET request to the load balancer
endpoint:\\n");
    Console.WriteLine($"\\thttp://{endPoint}\\n");
}
Console.WriteLine(new string('-', 80));
Console.WriteLine("Press Enter when you're ready to continue with the
demo.");
if (interactive)
    Console.ReadLine();
return true;
}

/// <summary>
/// Demonstrate the steps of the scenario.
/// </summary>
/// <param name="interactive">True to run as an interactive scenario.</param>
/// <returns>Async task.</returns>
public static async Task<bool> Demo(bool interactive)
{
```

```
var ssmOnlyPolicy = Path.Join(_configuration["resourcePath"],
    "ssm_only_policy.json");

Console.WriteLine(new string('-', 80));
Console.WriteLine("Resetting parameters to starting values for demo.");
await _smParameterWrapper.Reset();

Console.WriteLine("\nThis part of the demonstration shows how to toggle
different parts of the system\n" +
    "to create situations where the web service fails, and
shows how using a resilient\n" +
    "architecture can keep the web service running in spite
of these failures.");
Console.WriteLine(new string('-', 88));
Console.WriteLine("At the start, the load balancer endpoint returns
recommendations and reports that all targets are healthy.");
if (interactive)
    await DemoActionChoices();

Console.WriteLine($"The web service running on the EC2 instances gets
recommendations by querying a DynamoDB table.\n" +
    $"The table name is contained in a Systems Manager
parameter named '{_smParameterWrapper.TableParameter}'.\n" +
    $"To simulate a failure of the recommendation service,
let's set this parameter to name a non-existent table.\n");
await
_smParameterWrapper.PutParameterByName(_smParameterWrapper.TableParameter,
    "this-is-not-a-table");
Console.WriteLine("\nNow, sending a GET request to the load balancer
endpoint returns a failure code. But, the service reports as\n" +
    "healthy to the load balancer because shallow health
checks don't check for failure of the recommendation service.");
if (interactive)
    await DemoActionChoices();

Console.WriteLine("Instead of failing when the recommendation service
fails, the web service can return a static response.");
Console.WriteLine("While this is not a perfect solution, it presents the
customer with a somewhat better experience than failure.");

await
_smParameterWrapper.PutParameterByName(_smParameterWrapper.FailureResponseParameter,
    "static");
```

```
        Console.WriteLine("\nNow, sending a GET request to the load balancer
endpoint returns a static response.");
        Console.WriteLine("The service still reports as healthy because health
checks are still shallow.");
        if (interactive)
            await DemoActionChoices();

        Console.WriteLine("Let's reinstate the recommendation service.\n");
        await
_smParameterWrapper.PutParameterByName(_smParameterWrapper.TableParameter,
_smParameterWrapper.TableName);
        Console.WriteLine(
            "\nLet's also substitute bad credentials for one of the instances in
the target group so that it can't\n" +
            "access the DynamoDB recommendation table.\n"
        );
        await _autoScalerWrapper.CreateInstanceProfileWithName(
            _autoScalerWrapper.BadCredsPolicyName,
            _autoScalerWrapper.BadCredsRoleName,
            _autoScalerWrapper.BadCredsProfileName,
            ssmOnlyPolicy,
            new List<string> { "AmazonSSMManagedInstanceCore" }
        );
        var instances = await
_autoScalerWrapper.GetInstancesByGroupName(_autoScalerWrapper.GroupName);
        var badInstanceId = instances.First();
        var instanceProfile = await
_autoScalerWrapper.GetInstanceProfile(badInstanceId);
        Console.WriteLine(
            $"Replacing the profile for instance {badInstanceId} with a profile
that contains\n" +
            "bad credentials...\n"
        );
        await _autoScalerWrapper.ReplaceInstanceProfile(
            badInstanceId,
            _autoScalerWrapper.BadCredsProfileName,
            instanceProfile.AssociationId
        );
        Console.WriteLine(
            "Now, sending a GET request to the load balancer endpoint returns
either a recommendation or a static response,\n" +
            "depending on which instance is selected by the load balancer.\n"
        );
        if (interactive)
```

```
        await DemoActionChoices();

        Console.WriteLine("\nLet's implement a deep health check. For this demo,
a deep health check tests whether");
        Console.WriteLine("the web service can access the DynamoDB table that it
depends on for recommendations. Note that");
        Console.WriteLine("the deep health check is only for ELB routing and not
for Auto Scaling instance health.");
        Console.WriteLine("This kind of deep health check is not recommended for
Auto Scaling instance health, because it");
        Console.WriteLine("risks accidental termination of all instances in the
Auto Scaling group when a dependent service fails.");

        Console.WriteLine("\nBy implementing deep health checks, the load
balancer can detect when one of the instances is failing");
        Console.WriteLine("and take that instance out of rotation.");

        await
_smParameterWrapper.PutParameterByName(_smParameterWrapper.HealthCheckParameter,
"deep");

        Console.WriteLine($"Now, checking target health indicates that the
instance with bad credentials ({badInstanceId})");
        Console.WriteLine("is unhealthy. Note that it might take a minute or two
for the load balancer to detect the unhealthy");
        Console.WriteLine("instance. Sending a GET request to the load balancer
endpoint always returns a recommendation, because");
        Console.WriteLine("the load balancer takes unhealthy instances out of its
rotation.");

        if (interactive)
            await DemoActionChoices();

        Console.WriteLine("\nBecause the instances in this demo are controlled by
an auto scaler, the simplest way to fix an unhealthy");
        Console.WriteLine("instance is to terminate it and let the auto scaler
start a new instance to replace it.");

        await _autoScalerWrapper.TryTerminateInstanceById(badInstanceId);

        Console.WriteLine($"Even while the instance is terminating and the new
instance is starting, sending a GET");
        Console.WriteLine("request to the web service continues to get a
successful recommendation response because");
```

```
        Console.WriteLine("starts and reports as healthy, it is included in the
load balancing rotation.");
        Console.WriteLine("Note that terminating and replacing an instance
typically takes several minutes, during which time you");
        Console.WriteLine("can see the changing health check status until the new
instance is running and healthy.");

        if (interactive)
            await DemoActionChoices();

        Console.WriteLine("\nIf the recommendation service fails now, deep health
checks mean all instances report as unhealthy.");

        await
_smParameterWrapper.PutParameterByName(_smParameterWrapper.TableParameter,
"this-is-not-a-table");

        Console.WriteLine($"When all instances are unhealthy, the load balancer
continues to route requests even to");
        Console.WriteLine("unhealthy instances, allowing them to fail open and
return a static response rather than fail");
        Console.WriteLine("closed and report failure to the customer.");

        if (interactive)
            await DemoActionChoices();
        await _smParameterWrapper.Reset();

        Console.WriteLine(new string('-', 80));
        return true;
    }

    /// <summary>
    /// Clean up the resources from the scenario.
    /// </summary>
    /// <param name="interactive">True to ask the user for cleanup.</param>
    /// <returns>Async task.</returns>
    public static async Task<bool> DestroyResources(bool interactive)
    {
        Console.WriteLine(new string('-', 80));
        Console.WriteLine(
            "To keep things tidy and to avoid unwanted charges on your account,
we can clean up all AWS resources\n" +
            "that were created for this demo."
        );
    }
}
```

```

        if (!interactive || GetYesNoResponse("Do you want to clean up all demo
resources? (y/n) "))
        {
            await
            _elasticLoadBalancerWrapper.DeleteLoadBalancerByName(_elasticLoadBalancerWrapper.LoadBal
            await
            _elasticLoadBalancerWrapper.DeleteTargetGroupByName(_elasticLoadBalancerWrapper.TargetGr
            await
            _autoScalerWrapper.TerminateAndDeleteAutoScalingGroupWithName(_autoScalerWrapper.GroupNa
            await
            _autoScalerWrapper.DeleteKeyPairByName(_autoScalerWrapper.KeyPairName);
            await
            _autoScalerWrapper.DeleteTemplateByName(_autoScalerWrapper.LaunchTemplateName);
            await _autoScalerWrapper.DeleteInstanceProfile(
                _autoScalerWrapper.BadCredsProfileName,
                _autoScalerWrapper.BadCredsRoleName
            );
            await
            _recommendations.DestroyDatabaseByName(_recommendations.TableName);
        }
        else
        {
            Console.WriteLine(
                "Ok, we'll leave the resources intact.\n" +
                "Don't forget to delete them when you're done with them or you
might incur unexpected charges."
            );
        }

        Console.WriteLine(new string('-', 80));
        return true;
    }

```

Membuat kelas yang menggabungkan tindakan Penskalaan Otomatis dan Amazon EC2.

```

/// <summary>
/// Encapsulates Amazon EC2 Auto Scaling and EC2 management methods.
/// </summary>
public class AutoScalerWrapper
{
    private readonly IAmazonAutoScaling _amazonAutoScaling;

```



```
private readonly IAmazonEC2 _amazonEc2;
private readonly IAmazonSimpleSystemsManagement _amazonSsm;
private readonly IAmazonIdentityManagementService _amazonIam;

private readonly string _instanceType = "";
private readonly string _amiParam = "";
private readonly string _launchTemplateName = "";
private readonly string _groupName = "";
private readonly string _instancePolicyName = "";
private readonly string _instanceRoleName = "";
private readonly string _instanceProfileName = "";
private readonly string _badCredsProfileName = "";
private readonly string _badCredsRoleName = "";
private readonly string _badCredsPolicyName = "";
private readonly string _keyPairName = "";

public string GroupName => _groupName;
public string KeyPairName => _keyPairName;
public string LaunchTemplateName => _launchTemplateName;
public string InstancePolicyName => _instancePolicyName;
public string BadCredsProfileName => _badCredsProfileName;
public string BadCredsRoleName => _badCredsRoleName;
public string BadCredsPolicyName => _badCredsPolicyName;

/// <summary>
/// Constructor for the AutoScalerWrapper.
/// </summary>
/// <param name="amazonAutoScaling">The injected AutoScaling client.</param>
/// <param name="amazonEc2">The injected EC2 client.</param>
/// <param name="amazonIam">The injected IAM client.</param>
/// <param name="amazonSsm">The injected SSM client.</param>
public AutoScalerWrapper(
    IAmazonAutoScaling amazonAutoScaling,
    IAmazonEC2 amazonEc2,
    IAmazonSimpleSystemsManagement amazonSsm,
    IAmazonIdentityManagementService amazonIam,
    IConfiguration configuration)
{
    _amazonAutoScaling = amazonAutoScaling;
    _amazonEc2 = amazonEc2;
    _amazonSsm = amazonSsm;
    _amazonIam = amazonIam;

    var prefix = configuration["resourcePrefix"];
```

```

    _instanceType = configuration["instanceType"];
    _amiParam = configuration["amiParam"];

    _launchTemplateName = prefix + "-template";
    _groupName = prefix + "-group";
    _instancePolicyName = prefix + "-pol";
    _instanceRoleName = prefix + "-role";
    _instanceProfileName = prefix + "-prof";
    _badCredsPolicyName = prefix + "-bc-pol";
    _badCredsRoleName = prefix + "-bc-role";
    _badCredsProfileName = prefix + "-bc-prof";
    _keyPairName = prefix + "-key-pair";
}

/// <summary>
/// Create a policy, role, and profile that is associated with instances with
a specified name.
/// An instance's associated profile defines a role that is assumed by the
/// instance. The role has attached policies that specify the AWS permissions
granted to
/// clients that run on the instance.
/// </summary>
/// <param name="policyName">Name to use for the policy.</param>
/// <param name="roleName">Name to use for the role.</param>
/// <param name="profileName">Name to use for the profile.</param>
/// <param name="ssmOnlyPolicyFile">Path to a policy file for SSM.</param>
/// <param name="awsManagedPolicies">AWS Managed policies to be attached to
the role.</param>
/// <returns>The Arn of the profile.</returns>
public async Task<string> CreateInstanceProfileWithName(
    string policyName,
    string roleName,
    string profileName,
    string ssmOnlyPolicyFile,
    List<string>? awsManagedPolicies = null)
{
    var assumeRoleDoc = "{" +
        "\"Version\": \"2012-10-17\", " +
        "\"Statement\": [{" +
            "\"Effect\": \"Allow\", " +
            "\"Principal\": { " +
            "\"Service\": [ " +
                "\"ec2.amazonaws.com\" " +

```

```
        "]" +
        "}," +
        "\"Action\": \"sts:AssumeRole\"" +
        "}]\" +
        "}\";

var policyDocument = await File.ReadAllTextAsync(ssmOnlyPolicyFile);

var policyArn = "";

try
{
    var createPolicyResult = await _amazonIam.CreatePolicyAsync(
        new CreatePolicyRequest
        {
            PolicyName = policyName,
            PolicyDocument = policyDocument
        });
    policyArn = createPolicyResult.Policy.Arn;
}
catch (EntityAlreadyExistsException)
{
    // The policy already exists, so we look it up to get the Arn.
    var policiesPaginator = _amazonIam.Paginators.ListPolicies(
        new ListPoliciesRequest()
        {
            Scope = PolicyScopeType.Local
        });
    // Get the entire list using the paginator.
    await foreach (var policy in policiesPaginator.Policies)
    {
        if (policy.PolicyName.Equals(policyName))
        {
            policyArn = policy.Arn;
        }
    }

    if (policyArn == null)
    {
        throw new InvalidOperationException("Policy not found");
    }
}

try
```

```
{
    await _amazonIam.CreateRoleAsync(new CreateRoleRequest()
    {
        RoleName = roleName,
        AssumeRolePolicyDocument = assumeRoleDoc,
    });
    await _amazonIam.AttachRolePolicyAsync(new AttachRolePolicyRequest()
    {
        RoleName = roleName,
        PolicyArn = policyArn
    });
    if (awsManagedPolicies != null)
    {
        foreach (var awsPolicy in awsManagedPolicies)
        {
            await _amazonIam.AttachRolePolicyAsync(new
AttachRolePolicyRequest()
            {
                PolicyArn = $"arn:aws:iam::aws:policy/{awsPolicy}",
                RoleName = roleName
            });
        }
    }
}
catch (EntityAlreadyExistsException)
{
    Console.WriteLine("Role already exists.");
}

string profileArn = "";
try
{
    var profileCreateResponse = await
_amazonIam.CreateInstanceProfileAsync(
        new CreateInstanceProfileRequest()
        {
            InstanceProfileName = profileName
        });
    // Allow time for the profile to be ready.
    profileArn = profileCreateResponse.InstanceProfile.Arn;
    Thread.Sleep(10000);
    await _amazonIam.AddRoleToInstanceProfileAsync(
        new AddRoleToInstanceProfileRequest()
        {
```

```
        InstanceProfileName = profileName,
        RoleName = roleName
    });

}
catch (EntityAlreadyExistsException)
{
    Console.WriteLine("Policy already exists.");
    var profileGetResponse = await _amazonIam.GetInstanceProfileAsync(
        new GetInstanceProfileRequest()
        {
            InstanceProfileName = profileName
        });
    profileArn = profileGetResponse.InstanceProfile.Arn;
}
return profileArn;
}

/// <summary>
/// Create a new key pair and save the file.
/// </summary>
/// <param name="newKeyName">The name of the new key pair.</param>
/// <returns>Async task.</returns>
public async Task CreateKeyPair(string newKeyName)
{
    try
    {
        var keyResponse = await _amazonEc2.CreateKeyPairAsync(
            new CreateKeyPairRequest() { KeyName = newKeyName });
        await File.WriteAllTextAsync($"{newKeyName}.pem",
            keyResponse.KeyPair.KeyMaterial);
        Console.WriteLine($"Created key pair {newKeyName}.");
    }
    catch (AlreadyExistsException)
    {
        Console.WriteLine("Key pair already exists.");
    }
}

/// <summary>
/// Delete the key pair and file by name.
/// </summary>
/// <param name="deleteKeyName">The key pair to delete.</param>
/// <returns>Async task.</returns>
```

```
public async Task DeleteKeyPairByName(string deleteKeyName)
{
    try
    {
        await _amazonEc2.DeleteKeyPairAsync(
            new DeleteKeyPairRequest() { KeyName = deleteKeyName });
        File.Delete($"{deleteKeyName}.pem");
    }
    catch (FileNotFoundException)
    {
        Console.WriteLine($"Key pair {deleteKeyName} not found.");
    }
}

/// <summary>
/// Creates an Amazon EC2 launch template to use with Amazon EC2 Auto
Scaling.
/// The launch template specifies a Bash script in its user data field that
runs after
/// the instance is started. This script installs the Python packages and
starts a Python
/// web server on the instance.
/// </summary>
/// <param name="startupScriptPath">The path to a Bash script file that is
run.</param>
/// <param name="instancePolicyPath">The path to a permissions policy to
create and attach to the profile.</param>
/// <returns>The template object.</returns>
public async Task<Amazon.EC2.Model.LaunchTemplate> CreateTemplate(string
startupScriptPath, string instancePolicyPath)
{
    await CreateKeyPair(_keyPairName);
    await CreateInstanceProfileWithName(_instancePolicyName,
_instanceRoleName, _instanceProfileName, instancePolicyPath);

    var startServerText = await File.ReadAllTextAsync(startupScriptPath);
    var plainTextBytes = System.Text.Encoding.UTF8.GetBytes(startServerText);

    var amiLatest = await _amazonSsm.GetParameterAsync(
        new GetParameterRequest() { Name = _amiParam });
    var amiId = amiLatest.Parameter.Value;
    var launchTemplateResponse = await _amazonEc2.CreateLaunchTemplateAsync(
        new CreateLaunchTemplateRequest()
        {
```

```

        LaunchTemplateName = _launchTemplateName,
        LaunchTemplateData = new RequestLaunchTemplateData()
        {
            InstanceType = _instanceType,
            ImageId = amiId,
            IamInstanceProfile =
                new

LaunchTemplateIamInstanceProfileSpecificationRequest()
            {
                Name = _instanceProfileName
            },
            KeyName = _keyPairName,
            UserData = System.Convert.ToBase64String(plainTextBytes)
        }
    });
    return launchTemplateResponse.LaunchTemplate;
}

/// <summary>
/// Get a list of Availability Zones in the AWS Region of the Amazon EC2
Client.
/// </summary>
/// <returns>A list of availability zones.</returns>
public async Task<List<string>> DescribeAvailabilityZones()
{
    var zoneResponse = await _amazonEc2.DescribeAvailabilityZonesAsync(
        new DescribeAvailabilityZonesRequest());
    return zoneResponse.AvailabilityZones.Select(z => z.ZoneName).ToList();
}

/// <summary>
/// Create an EC2 Auto Scaling group of a specified size and name.
/// </summary>
/// <param name="groupSize">The size for the group.</param>
/// <param name="groupName">The name for the group.</param>
/// <param name="availabilityZones">The availability zones for the group.</
param>
/// <returns>Async task.</returns>
public async Task CreateGroupOfSize(int groupSize, string groupName,
List<string> availabilityZones)
{
    try

```

```

    {
        await _amazonAutoScaling.CreateAutoScalingGroupAsync(
            new CreateAutoScalingGroupRequest()
            {
                AutoScalingGroupName = groupName,
                AvailabilityZones = availabilityZones,
                LaunchTemplate =
                    new
Amazon.AutoScaling.Model.LaunchTemplateSpecification()
                    {
                        LaunchTemplateName = _launchTemplateName,
                        Version = "$Default"
                    },
                MaxSize = groupSize,
                MinSize = groupSize
            });
        Console.WriteLine($"Created EC2 Auto Scaling group {groupName} with
size {groupSize}.");
    }
    catch (EntityAlreadyExistsException)
    {
        Console.WriteLine($"EC2 Auto Scaling group {groupName} already
exists.");
    }
}

/// <summary>
/// Get the default VPC for the account.
/// </summary>
/// <returns>The default VPC object.</returns>
public async Task<Vpc> GetDefaultVpc()
{
    var vpcResponse = await _amazonEc2.DescribeVpcsAsync(
        new DescribeVpcsRequest()
        {
            Filters = new List<Amazon.EC2.Model.Filter>()
            {
                new ("is-default", new List<string>() { "true" })
            }
        });
    return vpcResponse.Vpcs[0];
}

/// <summary>

```



```
/// Get all the subnets for a Vpc in a set of availability zones.
/// </summary>
/// <param name="vpcId">The Id of the Vpc.</param>
/// <param name="availabilityZones">The list of availability zones.</param>
/// <returns>The collection of subnet objects.</returns>
public async Task<List<Subnet>> GetAllVpcSubnetsForZones(string vpcId,
List<string> availabilityZones)
{
    var subnets = new List<Subnet>();
    var subnetPaginator = _amazonEc2.Paginators.DescribeSubnets(
        new DescribeSubnetsRequest()
        {
            Filters = new List<Amazon.EC2.Model.Filter>()
            {
                new ("vpc-id", new List<string>() { vpcId}),
                new ("availability-zone", availabilityZones),
                new ("default-for-az", new List<string>() { "true" })
            }
        });

    // Get the entire list using the paginator.
    await foreach (var subnet in subnetPaginator.Subnets)
    {
        subnets.Add(subnet);
    }

    return subnets;
}

/// <summary>
/// Delete a launch template by name.
/// </summary>
/// <param name="templateName">The name of the template to delete.</param>
/// <returns>Async task.</returns>
public async Task DeleteTemplateByName(string templateName)
{
    try
    {
        await _amazonEc2.DeleteLaunchTemplateAsync(
            new DeleteLaunchTemplateRequest()
            {
                LaunchTemplateName = templateName
            });
    }
}
```

```
        catch (AmazonClientException)
        {
            Console.WriteLine($"Unable to delete template {templateName}.");
        }
    }

    /// <summary>
    /// Detaches a role from an instance profile, detaches policies from the
    role,
    /// and deletes all the resources.
    /// </summary>
    /// <param name="profileName">The name of the profile to delete.</param>
    /// <param name="roleName">The name of the role to delete.</param>
    /// <returns>Async task.</returns>
    public async Task DeleteInstanceProfile(string profileName, string roleName)
    {
        try
        {
            await _amazonIam.RemoveRoleFromInstanceProfileAsync(
                new RemoveRoleFromInstanceProfileRequest()
                {
                    InstanceProfileName = profileName,
                    RoleName = roleName
                });
            await _amazonIam.DeleteInstanceProfileAsync(
                new DeleteInstanceProfileRequest() { InstanceProfileName =
profileName });
            var attachedPolicies = await
_amazonIam.ListAttachedRolePoliciesAsync(
                new ListAttachedRolePoliciesRequest() { RoleName = roleName });
            foreach (var policy in attachedPolicies.AttachedPolicies)
            {
                await _amazonIam.DetachRolePolicyAsync(
                    new DetachRolePolicyRequest()
                    {
                        RoleName = roleName,
                        PolicyArn = policy.PolicyArn
                    });
                // Delete the custom policies only.
                if (!policy.PolicyArn.StartsWith("arn:aws:iam::aws"))
                {
                    await _amazonIam.DeletePolicyAsync(
                        new Amazon.IdentityManagement.Model.DeletePolicyRequest()
                        {
```

```

        PolicyArn = policy.PolicyArn
    });
    }
}

await _amazonIam.DeleteRoleAsync(
    new DeleteRoleRequest() { RoleName = roleName });
}
catch (NoSuchEntityException)
{
    Console.WriteLine($"Instance profile {profileName} does not exist.");
}
}

/// <summary>
/// Gets data about the instances in an EC2 Auto Scaling group by its group
name.
/// </summary>
/// <param name="group">The name of the auto scaling group.</param>
/// <returns>A collection of instance Ids.</returns>
public async Task<IEnumerable<string>> GetInstancesByGroupName(string group)
{
    var instanceResponse = await
    _amazonAutoScaling.DescribeAutoScalingGroupsAsync(
        new DescribeAutoScalingGroupsRequest()
        {
            AutoScalingGroupNames = new List<string>() { group }
        });
    var instanceIds = instanceResponse.AutoScalingGroups.SelectMany(
        g => g.Instances.Select(i => i.InstanceId));
    return instanceIds;
}

/// <summary>
/// Get the instance profile association data for an instance.
/// </summary>
/// <param name="instanceId">The Id of the instance.</param>
/// <returns>Instance profile associations data.</returns>
public async Task<IamInstanceProfileAssociation> GetInstanceProfile(string
instanceId)
{
    var response = await
    _amazonEc2.DescribeIamInstanceProfileAssociationsAsync(
        new DescribeIamInstanceProfileAssociationsRequest()

```

```
        {
            Filters = new List<Amazon.EC2.Model.Filter>()
            {
                new ("instance-id", new List<string>() { instanceId })
            },
        });
    return response.IamInstanceProfileAssociations[0];
}

/// <summary>
/// Replace the profile associated with a running instance. After the profile
is replaced, the instance
/// is rebooted to ensure that it uses the new profile. When the instance is
ready, Systems Manager is
/// used to restart the Python web server.
/// </summary>
/// <param name="instanceId">The Id of the instance to update.</param>
/// <param name="credsProfileName">The name of the new profile to associate
with the specified instance.</param>
/// <param name="associationId">The Id of the existing profile association
for the instance.</param>
/// <returns>Async task.</returns>
public async Task ReplaceInstanceProfile(string instanceId, string
credsProfileName, string associationId)
{
    await _amazonEc2.ReplaceIamInstanceProfileAssociationAsync(
        new ReplaceIamInstanceProfileAssociationRequest()
        {
            AssociationId = associationId,
            IamInstanceProfile = new IamInstanceProfileSpecification()
            {
                Name = credsProfileName
            }
        });
    // Allow time before resetting.
    Thread.Sleep(25000);
    var instanceReady = false;
    var retries = 5;
    while (retries-- > 0 && !instanceReady)
    {
        await _amazonEc2.RebootInstancesAsync(
            new RebootInstancesRequest(new List<string>() { instanceId }));
        Thread.Sleep(10000);
    }
}
```

```

        var instancesPaginator =
        _amazonSsm.Paginators.DescribeInstanceInformation(
            new DescribeInstanceInformationRequest());
        // Get the entire list using the paginator.
        await foreach (var instance in
instancesPaginator.InstanceInformationList)
        {
            instanceReady = instance.InstanceId == instanceId;
            if (instanceReady)
            {
                break;
            }
        }
    }
    Console.WriteLine($"Sending restart command to instance {instanceId}");
    await _amazonSsm.SendCommandAsync(
        new SendCommandRequest()
        {
            InstanceIds = new List<string>() { instanceId },
            DocumentName = "AWS-RunShellScript",
            Parameters = new Dictionary<string, List<string>>()
            {
                {"commands", new List<string>() { "cd / && sudo python3
server.py 80" }}
            }
        });
    Console.WriteLine($"Restarted the web server on instance {instanceId}");
}

/// <summary>
/// Try to terminate an instance by its Id.
/// </summary>
/// <param name="instanceId">The Id of the instance to terminate.</param>
/// <returns>Async task.</returns>
public async Task TryTerminateInstanceById(string instanceId)
{
    var stopping = false;
    Console.WriteLine($"Stopping {instanceId}...");
    while (!stopping)
    {
        try
        {
            await
            _amazonAutoScaling.TerminateInstanceInAutoScalingGroupAsync(

```

```
        new TerminateInstanceInAutoScalingGroupRequest()
        {
            InstanceId = instanceId,
            ShouldDecrementDesiredCapacity = false
        });
        stopping = true;
    }
    catch (ScalingActivityInProgressException)
    {
        Console.WriteLine($"Scaling activity in progress for
{instanceId}. Waiting...");
        Thread.Sleep(10000);
    }
}

/// <summary>
/// Tries to delete the EC2 Auto Scaling group. If the group is in use or in
progress,
/// waits and retries until the group is successfully deleted.
/// </summary>
/// <param name="groupName">The name of the group to try to delete.</param>
/// <returns>Async task.</returns>
public async Task TryDeleteGroupByName(string groupName)
{
    var stopped = false;
    while (!stopped)
    {
        try
        {
            await _amazonAutoScaling.DeleteAutoScalingGroupAsync(
                new DeleteAutoScalingGroupRequest()
                {
                    AutoScalingGroupName = groupName
                });
            stopped = true;
        }
        catch (Exception e)
            when ((e is ScalingActivityInProgressException)
                || (e is Amazon.AutoScaling.Model.ResourceInUseException))
        {
            Console.WriteLine($"Some instances are still running.
Waiting...");
            Thread.Sleep(10000);
        }
    }
}
```

```
    }
  }
}

/// <summary>
/// Terminate instances and delete the Auto Scaling group by name.
/// </summary>
/// <param name="groupName">The name of the group to delete.</param>
/// <returns>Async task.</returns>
public async Task TerminateAndDeleteAutoScalingGroupWithName(string
groupName)
{
    var describeGroupsResponse = await
_amazonAutoScaling.DescribeAutoScalingGroupsAsync(
    new DescribeAutoScalingGroupsRequest()
    {
        AutoScalingGroupNames = new List<string>() { groupName }
    });
    if (describeGroupsResponse.AutoScalingGroups.Any())
    {
        // Update the size to 0.
        await _amazonAutoScaling.UpdateAutoScalingGroupAsync(
            new UpdateAutoScalingGroupRequest()
            {
                AutoScalingGroupName = groupName,
                MinSize = 0
            });
        var group = describeGroupsResponse.AutoScalingGroups[0];
        foreach (var instance in group.Instances)
        {
            await TryTerminateInstanceById(instance.InstanceId);
        }

        await TryDeleteGroupByName(groupName);
    }
    else
    {
        Console.WriteLine($"No groups found with name {groupName}.");
    }
}

/// <summary>
/// Get the default security group for a specified Vpc.
```

```
/// </summary>
/// <param name="vpc">The Vpc to search.</param>
/// <returns>The default security group.</returns>
public async Task<SecurityGroup> GetDefaultSecurityGroupForVpc(Vpc vpc)
{
    var groupResponse = await _amazonEc2.DescribeSecurityGroupsAsync(
        new DescribeSecurityGroupsRequest()
        {
            Filters = new List<Amazon.EC2.Model.Filter>()
            {
                new ("group-name", new List<string>() { "default" }),
                new ("vpc-id", new List<string>() { vpc.VpcId })
            }
        });
    return groupResponse.SecurityGroups[0];
}

/// <summary>
/// Verify the default security group of a Vpc allows ingress from the
calling computer.
/// This can be done by allowing ingress from this computer's IP address.
/// In some situations, such as connecting from a corporate network, you must
instead specify
/// a prefix list Id. You can also temporarily open the port to any IP
address while running this example.
/// If you do, be sure to remove public access when you're done.
/// </summary>
/// <param name="vpc">The group to check.</param>
/// <param name="port">The port to verify.</param>
/// <param name="ipAddress">This computer's IP address.</param>
/// <returns>True if the ip address is allowed on the group.</returns>
public bool VerifyInboundPortForGroup(SecurityGroup group, int port, string
ipAddress)
{
    var portIsOpen = false;
    foreach (var ipPermission in group.IpPermissions)
    {
        if (ipPermission.FromPort == port)
        {
            foreach (var ipRange in ipPermission.Ipv4Ranges)
            {
                var cidr = ipRange.CidrIp;
                if (cidr.StartsWith(ipAddress) || cidr == "0.0.0.0/0")
                {
```



```
        portIsOpen = true;
    }
}

if (ipPermission.PrefixListIds.Any())
{
    portIsOpen = true;
}

if (!portIsOpen)
{
    Console.WriteLine("The inbound rule does not appear to be
open to either this computer's IP\n" +
                        "address, to all IP addresses (0.0.0.0/0),
or to a prefix list ID.");
}
else
{
    break;
}
}
}

return portIsOpen;
}

/// <summary>
/// Add an ingress rule to the specified security group that allows access on
the
/// specified port from the specified IP address.
/// </summary>
/// <param name="groupId">The Id of the security group to modify.</param>
/// <param name="port">The port to open.</param>
/// <param name="ipAddress">The IP address to allow access.</param>
/// <returns>Async task.</returns>
public async Task OpenInboundPort(string groupId, int port, string ipAddress)
{
    await _amazonEc2.AuthorizeSecurityGroupIngressAsync(
        new AuthorizeSecurityGroupIngressRequest()
        {
            GroupId = groupId,
            IpPermissions = new List<IpPermission>()
            {
                new IpPermission()
            }
        }
    );
}
```

```

        {
            FromPort = port,
            ToPort = port,
            IpProtocol = "tcp",
            Ipv4Ranges = new List<IpRange>()
            {
                new IpRange() { CidrIp = $"{ipAddress}/32" }
            }
        }
    });
}

/// <summary>
/// Attaches an Elastic Load Balancing (ELB) target group to this EC2 Auto
Scaling group.
/// The
/// </summary>
/// <param name="autoScalingGroupName">The name of the Auto Scaling group.</
param>
/// <param name="targetGroupArn">The Arn for the target group.</param>
/// <returns>Async task.</returns>
public async Task AttachLoadBalancerToGroup(string autoScalingGroupName,
string targetGroupArn)
{
    await _amazonAutoScaling.AttachLoadBalancerTargetGroupsAsync(
        new AttachLoadBalancerTargetGroupsRequest()
        {
            AutoScalingGroupName = autoScalingGroupName,
            TargetGroupARNs = new List<string>() { targetGroupArn }
        });
}
}

```

Membuat kelas yang menggabungkan tindakan Penyeimbangan Beban Elastis.

```

/// <summary>
/// Encapsulates Elastic Load Balancer actions.
/// </summary>
public class ElasticLoadBalancerWrapper
{

```

```
private readonly IAmazonElasticLoadBalancingV2 _amazonElasticLoadBalancingV2;
private string? _endpoint = null;
private readonly string _targetGroupName = "";
private readonly string _loadBalancerName = "";
HttpClient _httpClient = new();

public string TargetGroupName => _targetGroupName;
public string LoadBalancerName => _loadBalancerName;

/// <summary>
/// Constructor for the Elastic Load Balancer wrapper.
/// </summary>
/// <param name="amazonElasticLoadBalancingV2">The injected load balancing v2
client.</param>
/// <param name="configuration">The injected configuration.</param>
public ElasticLoadBalancerWrapper(
    IAmazonElasticLoadBalancingV2 amazonElasticLoadBalancingV2,
    IConfiguration configuration)
{
    _amazonElasticLoadBalancingV2 = amazonElasticLoadBalancingV2;
    var prefix = configuration["resourcePrefix"];
    _targetGroupName = prefix + "-tg";
    _loadBalancerName = prefix + "-lb";
}

/// <summary>
/// Get the HTTP Endpoint of a load balancer by its name.
/// </summary>
/// <param name="loadBalancerName">The name of the load balancer.</param>
/// <returns>The HTTP endpoint.</returns>
public async Task<string> GetEndpointForLoadBalancerByName(string
loadBalancerName)
{
    if (_endpoint == null)
    {
        var endpointResponse =
            await _amazonElasticLoadBalancingV2.DescribeLoadBalancersAsync(
                new DescribeLoadBalancersRequest()
                {
                    Names = new List<string>() { loadBalancerName }
                });
        _endpoint = endpointResponse.LoadBalancers[0].DNSName;
    }
}
```

```
        return _endpoint;
    }

    /// <summary>
    /// Return the GET response for an endpoint as text.
    /// </summary>
    /// <param name="endpoint">The endpoint for the request.</param>
    /// <returns>The request response.</returns>
    public async Task<string> GetEndPointResponse(string endpoint)
    {
        var endpointResponse = await _httpClient.GetAsync($"http://{endpoint}");
        var textResponse = await endpointResponse.Content.ReadAsStringAsync();
        return textResponse!;
    }

    /// <summary>
    /// Get the target health for a group by name.
    /// </summary>
    /// <param name="groupName">The name of the group.</param>
    /// <returns>The collection of health descriptions.</returns>
    public async Task<List<TargetHealthDescription>>
    CheckTargetHealthForGroup(string groupName)
    {
        List<TargetHealthDescription> result = null!;
        try
        {
            var groupResponse =
                await _amazonElasticLoadBalancingV2.DescribeTargetGroupsAsync(
                    new DescribeTargetGroupsRequest()
                    {
                        Names = new List<string>() { groupName }
                    });
            var healthResponse =
                await _amazonElasticLoadBalancingV2.DescribeTargetHealthAsync(
                    new DescribeTargetHealthRequest()
                    {
                        TargetGroupArn =
                            groupResponse.TargetGroups[0].TargetGroupArn
                    });
            ;
            result = healthResponse.TargetHealthDescriptions;
        }
        catch (TargetGroupNotFoundException)
        {
```

```
        Console.WriteLine($"Target group {groupName} not found.");
    }
    return result;
}

/// <summary>
/// Create an Elastic Load Balancing target group. The target group specifies
how the load balancer forwards
/// requests to instances in the group and how instance health is checked.
///
/// To speed up this demo, the health check is configured with shortened
times and lower thresholds. In production,
/// you might want to decrease the sensitivity of your health checks to avoid
unwanted failures.
/// </summary>
/// <param name="groupName">The name for the group.</param>
/// <param name="protocol">The protocol, such as HTTP.</param>
/// <param name="port">The port to use to forward requests, such as 80.</
param>
/// <param name="vpcId">The Id of the Vpc in which the load balancer
exists.</param>
/// <returns>The new TargetGroup object.</returns>
public async Task<TargetGroup> CreateTargetGroupOnVpc(string groupName,
ProtocolEnum protocol, int port, string vpcId)
{
    var createResponse = await
_amazonElasticLoadBalancingV2.CreateTargetGroupAsync(
    new CreateTargetGroupRequest()
    {
        Name = groupName,
        Protocol = protocol,
        Port = port,
        HealthCheckPath = "/healthcheck",
        HealthCheckIntervalSeconds = 10,
        HealthCheckTimeoutSeconds = 5,
        HealthyThresholdCount = 2,
        UnhealthyThresholdCount = 2,
        VpcId = vpcId
    });
    var targetGroup = createResponse.TargetGroups[0];
    return targetGroup;
}

/// <summary>
```

```
    /// Create an Elastic Load Balancing load balancer that uses the specified
subnets
    /// and forwards requests to the specified target group.
    /// </summary>
    /// <param name="name">The name for the new load balancer.</param>
    /// <param name="subnetIds">Subnets for the load balancer.</param>
    /// <param name="targetGroup">Target group for forwarded requests.</param>
    /// <returns>The new LoadBalancer object.</returns>
    public async Task<LoadBalancer> CreateLoadBalancerAndListener(string name,
List<string> subnetIds, TargetGroup targetGroup)
    {
        var createLbResponse = await
        _amazonElasticLoadBalancingV2.CreateLoadBalancerAsync(
            new CreateLoadBalancerRequest()
            {
                Name = name,
                Subnets = subnetIds
            });
        var loadBalancerArn = createLbResponse.LoadBalancers[0].LoadBalancerArn;

        // Wait for load balancer to be available.
        var loadBalancerReady = false;
        while (!loadBalancerReady)
        {
            try
            {
                var describeResponse =
                await
                _amazonElasticLoadBalancingV2.DescribeLoadBalancersAsync(
                    new DescribeLoadBalancersRequest()
                    {
                        Names = new List<string>() { name }
                    });

                var loadBalancerState =
                describeResponse.LoadBalancers[0].State.Code;

                loadBalancerReady = loadBalancerState ==
                LoadBalancerStateEnum.Active;
            }
            catch (LoadBalancerNotFoundException)
            {
                loadBalancerReady = false;
            }
        }
    }
}
```

```
        Thread.Sleep(10000);
    }
    // Create the listener.
    await _amazonElasticLoadBalancingV2.CreateListenerAsync(
        new CreateListenerRequest()
        {
            LoadBalancerArn = loadBalancerArn,
            Protocol = targetGroup.Protocol,
            Port = targetGroup.Port,
            DefaultActions = new List<Action>()
            {
                new Action()
                {
                    Type = ActionTypeEnum.Forward,
                    TargetGroupArn = targetGroup.TargetGroupArn
                }
            }
        });
    return createLbResponse.LoadBalancers[0];
}

/// <summary>
/// Verify this computer can successfully send a GET request to the
/// load balancer endpoint.
/// </summary>
/// <param name="endpoint">The endpoint to check.</param>
/// <returns>True if successful.</returns>
public async Task<bool> VerifyLoadBalancerEndpoint(string endpoint)
{
    var success = false;
    var retries = 3;
    while (!success && retries > 0)
    {
        try
        {
            var endpointResponse = await _httpClient.GetAsync($"http://{
{endpoint}");
            Console.WriteLine($"Response: {endpointResponse.StatusCode}.");

            if (endpointResponse.IsSuccessStatusCode)
            {
                success = true;
            }
            else

```

```
        {
            retries = 0;
        }
    }
    catch (HttpRequestException)
    {
        Console.WriteLine("Connection error, retrying...");
        retries--;
        Thread.Sleep(10000);
    }
}

return success;
}

/// <summary>
/// Delete a load balancer by its specified name.
/// </summary>
/// <param name="name">The name of the load balancer to delete.</param>
/// <returns>Async task.</returns>
public async Task DeleteLoadBalancerByName(string name)
{
    try
    {
        var describeLoadBalancerResponse =
            await _amazonElasticLoadBalancingV2.DescribeLoadBalancersAsync(
                new DescribeLoadBalancersRequest()
                {
                    Names = new List<string>() { name }
                });
        var lbArn =
describeLoadBalancerResponse.LoadBalancers[0].LoadBalancerArn;
            await _amazonElasticLoadBalancingV2.DeleteLoadBalancerAsync(
                new DeleteLoadBalancerRequest()
                {
                    LoadBalancerArn = lbArn
                }
            );
    }
    catch (LoadBalancerNotFoundException)
    {
        Console.WriteLine($"Load balancer {name} not found.");
    }
}
}
```



```
/// <summary>
/// Delete a TargetGroup by its specified name.
/// </summary>
/// <param name="groupName">Name of the group to delete.</param>
/// <returns>Async task.</returns>
public async Task DeleteTargetGroupByName(string groupName)
{
    var done = false;
    while (!done)
    {
        try
        {
            var groupResponse =
                await
                _amazonElasticLoadBalancingV2.DescribeTargetGroupsAsync(
                    new DescribeTargetGroupsRequest()
                    {
                        Names = new List<string>() { groupName }
                    });

            var targetArn = groupResponse.TargetGroups[0].TargetGroupArn;
            await _amazonElasticLoadBalancingV2.DeleteTargetGroupAsync(
                new DeleteTargetGroupRequest() { TargetGroupArn =
targetArn });
            Console.WriteLine($"Deleted load balancing target group
{groupName}.");
            done = true;
        }
        catch (TargetGroupNotFoundException)
        {
            Console.WriteLine(
                $"Target group {groupName} not found, could not delete.");
            done = true;
        }
        catch (ResourceInUseException)
        {
            Console.WriteLine("Target group not yet released, waiting...");
            Thread.Sleep(10000);
        }
    }
}
}
```

Membuat kelas yang menggunakan DynamoDB untuk menyimulasikan layanan yang direkomendasikan.

```
/// <summary>
/// Encapsulates a DynamoDB table to use as a service that recommends books,
/// movies, and songs.
/// </summary>
public class Recommendations
{
    private readonly IAmazonDynamoDB _amazonDynamoDb;
    private readonly DynamoDBContext _context;
    private readonly string _tableName;

    public string TableName => _tableName;

    /// <summary>
    /// Constructor for the Recommendations service.
    /// </summary>
    /// <param name="amazonDynamoDb">The injected DynamoDb client.</param>
    /// <param name="configuration">The injected configuration.</param>
    public Recommendations(IAmazonDynamoDB amazonDynamoDb, IConfiguration
configuration)
    {
        _amazonDynamoDb = amazonDynamoDb;
        _context = new DynamoDBContext(_amazonDynamoDb);
        _tableName = configuration["databaseName"]!;
    }

    /// <summary>
    /// Create the DynamoDb table with a specified name.
    /// </summary>
    /// <param name="tableName">The name for the table.</param>
    /// <returns>True when ready.</returns>
    public async Task<bool> CreateDatabaseWithName(string tableName)
    {
        try
        {
            Console.WriteLine($"Creating table {tableName}...");
            var createRequest = new CreateTableRequest()
            {
                TableName = tableName,
```

```
AttributeDefinitions = new List<AttributeDefinition>()
{
    new AttributeDefinition()
    {
        AttributeName = "MediaType",
        AttributeType = ScalarAttributeType.S
    },
    new AttributeDefinition()
    {
        AttributeName = "ItemId",
        AttributeType = ScalarAttributeType.N
    }
},
KeySchema = new List<KeySchemaElement>()
{
    new KeySchemaElement()
    {
        AttributeName = "MediaType",
        KeyType = KeyType.HASH
    },
    new KeySchemaElement()
    {
        AttributeName = "ItemId",
        KeyType = KeyType.RANGE
    }
},
ProvisionedThroughput = new ProvisionedThroughput()
{
    ReadCapacityUnits = 5,
    WriteCapacityUnits = 5
}
};
await _amazonDynamoDb.CreateTableAsync(createRequest);

// Wait until the table is ACTIVE and then report success.
Console.WriteLine("\nWaiting for table to become active...");

var request = new DescribeTableRequest
{
    TableName = tableName
};

TableStatus status;
do
```

```
        {
            Thread.Sleep(2000);

            var describeTableResponse = await
                _amazonDynamoDb.DescribeTableAsync(request);
            status = describeTableResponse.Table.TableStatus;

            Console.WriteLine(".");
        }
        while (status != "ACTIVE");

        return status == TableStatus.ACTIVE;
    }
    catch (ResourceInUseException)
    {
        Console.WriteLine($"Table {tableName} already exists.");
        return false;
    }
}

/// <summary>
/// Populate the database table with data from a specified path.
/// </summary>
/// <param name="databaseTableName">The name of the table.</param>
/// <param name="recommendationsPath">The path of the recommendations data.</
param>
/// <returns>Async task.</returns>
public async Task PopulateDatabase(string databaseTableName, string
recommendationsPath)
{
    var recommendationsText = await
        File.ReadAllTextAsync(recommendationsPath);
    var records =

        JsonSerializer.Deserialize<RecommendationModel[]>(recommendationsText);
    var batchWrite = _context.CreateBatchWrite<RecommendationModel>();

    foreach (var record in records!)
    {
        batchWrite.AddPutItem(record);
    }

    await batchWrite.ExecuteAsync();
}
```

```
/// <summary>
/// Delete the recommendation table by name.
/// </summary>
/// <param name="tableName">The name of the recommendation table.</param>
/// <returns>Async task.</returns>
public async Task DestroyDatabaseByName(string tableName)
{
    try
    {
        await _amazonDynamoDb.DeleteTableAsync(
            new DeleteTableRequest() { TableName = tableName });
        Console.WriteLine($"Table {tableName} was deleted.");
    }
    catch (ResourceNotFoundException)
    {
        Console.WriteLine($"Table {tableName} not found");
    }
}
}
```

Membuat kelas yang mengabungkan tindakan Systems Manager.

```
/// <summary>
/// Encapsulates Systems Manager parameter operations. This example uses these
/// parameters
/// to drive the demonstration of resilient architecture, such as failure of a
/// dependency or
/// how the service responds to a health check.
/// </summary>
public class SmParameterWrapper
{
    private readonly IAmazonSimpleSystemsManagement
        _amazonSimpleSystemsManagement;

    private readonly string _tableParameter = "doc-example-resilient-
architecture-table";
    private readonly string _failureResponseParameter = "doc-example-resilient-
architecture-failure-response";
    private readonly string _healthCheckParameter = "doc-example-resilient-
architecture-health-check";
    private readonly string _tableName = "";
}
```

```
public string TableParameter => _tableParameter;
public string TableName => _tableName;
public string HealthCheckParameter => _healthCheckParameter;
public string FailureResponseParameter => _failureResponseParameter;

/// <summary>
/// Constructor for the SmParameterWrapper.
/// </summary>
/// <param name="amazonSimpleSystemsManagement">The injected Simple Systems
Management client.</param>
/// <param name="configuration">The injected configuration.</param>
public SmParameterWrapper(IAmazonSimpleSystemsManagement
amazonSimpleSystemsManagement, IConfiguration configuration)
{
    _amazonSimpleSystemsManagement = amazonSimpleSystemsManagement;
    _tableName = configuration["databaseName"]!;
}

/// <summary>
/// Reset the Systems Manager parameters to starting values for the demo.
/// </summary>
/// <returns>Async task.</returns>
public async Task Reset()
{
    await this.PutParameterByName(_tableParameter, _tableName);
    await this.PutParameterByName(_failureResponseParameter, "none");
    await this.PutParameterByName(_healthCheckParameter, "shallow");
}

/// <summary>
/// Set the value of a named Systems Manager parameter.
/// </summary>
/// <param name="name">The name of the parameter.</param>
/// <param name="value">The value to set.</param>
/// <returns>Async task.</returns>
public async Task PutParameterByName(string name, string value)
{
    await _amazonSimpleSystemsManagement.PutParameterAsync(
        new PutParameterRequest() { Name = name, Value = value, Overwrite =
true });
}
}
```

- Untuk detail API, lihat topik berikut di Referensi API AWS SDK for .NET .
 - [AttachLoadBalancerTargetGroups](#)
 - [CreateAutoScalingGroup](#)
 - [CreateInstanceProfile](#)
 - [CreateLaunchTemplate](#)
 - [CreateListener](#)
 - [CreateLoadBalancer](#)
 - [CreateTargetGroup](#)
 - [DeleteAutoScalingGroup](#)
 - [DeleteInstanceProfile](#)
 - [DeleteLaunchTemplate](#)
 - [DeleteLoadBalancer](#)
 - [DeleteTargetGroup](#)
 - [DescribeAutoScalingGroups](#)
 - [DescribeAvailabilityZones](#)
 - [DescribeInstanceProfileAssociations](#)
 - [DescribeInstances](#)
 - [DescribeLoadBalancers](#)
 - [DescribeSubnets](#)
 - [DescribeTargetGroups](#)
 - [DescribeTargetHealth](#)
 - [DescribeVpcs](#)
 - [RebootInstances](#)
 - [ReplacelamInstanceProfileAssociation](#)
 - [TerminateInstanceInAutoScalingGroup](#)
 - [UpdateAutoScalingGroup](#)

Java

SDK untuk Java 2.x

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturannya dan menjalankannya di [Repositori Contoh Kode AWS](#).

Menjalankan skenario interaktif di prompt perintah.

```
public class Main {

    public static final String fileName = "C:\\AWS\\resworkflow\\
\\recommendations.json"; // Modify file location.
    public static final String tableName = "doc-example-recommendation-service";
    public static final String startScript = "C:\\AWS\\resworkflow\\
\\server_startup_script.sh"; // Modify file location.
    public static final String policyFile = "C:\\AWS\\resworkflow\\
\\instance_policy.json"; // Modify file location.
    public static final String ssmJSON = "C:\\AWS\\resworkflow\\
\\ssm_only_policy.json"; // Modify file location.
    public static final String failureResponse = "doc-example-resilient-
architecture-failure-response";
    public static final String healthCheck = "doc-example-resilient-architecture-
health-check";
    public static final String templateName = "doc-example-resilience-template";
    public static final String roleName = "doc-example-resilience-role";
    public static final String policyName = "doc-example-resilience-pol";
    public static final String profileName = "doc-example-resilience-prof";

    public static final String badCredsProfileName = "doc-example-resilience-
prof-bc";

    public static final String targetGroupName = "doc-example-resilience-tg";
    public static final String autoScalingGroupName = "doc-example-resilience-
group";
    public static final String lbName = "doc-example-resilience-lb";
    public static final String protocol = "HTTP";
    public static final int port = 80;
```



```
public static final String DASHES = new String(new char[80]).replace("\0",
"-");

public static void main(String[] args) throws IOException,
InterruptedException {
    Scanner in = new Scanner(System.in);
    Database database = new Database();
    AutoScaler autoScaler = new AutoScaler();
    LoadBalancer loadBalancer = new LoadBalancer();

    System.out.println(DASHES);
    System.out.println("Welcome to the demonstration of How to Build and
Manage a Resilient Service!");
    System.out.println(DASHES);

    System.out.println(DASHES);
    System.out.println("A - SETUP THE RESOURCES");
    System.out.println("Press Enter when you're ready to start deploying
resources.");
    in.nextLine();
    deploy(loadBalancer);
    System.out.println(DASHES);
    System.out.println(DASHES);
    System.out.println("B - DEMO THE RESILIENCE FUNCTIONALITY");
    System.out.println("Press Enter when you're ready.");
    in.nextLine();
    demo(loadBalancer);
    System.out.println(DASHES);

    System.out.println(DASHES);
    System.out.println("C - DELETE THE RESOURCES");
    System.out.println("""
        This concludes the demo of how to build and manage a resilient
service.

        To keep things tidy and to avoid unwanted charges on your
account, we can clean up all AWS resources
that were created for this demo.
        """);

    System.out.println("\n Do you want to delete the resources (y/n)? ");
    String userInput = in.nextLine().trim().toLowerCase(); // Capture user
input

    if (userInput.equals("y")) {
```

```

        // Delete resources here
        deleteResources(loadBalancer, autoScaler, database);
        System.out.println("Resources deleted.");
    } else {
        System.out.println("""
            Okay, we'll leave the resources intact.
            Don't forget to delete them when you're done with them or you
might incur unexpected charges.
            """);
    }
    System.out.println(DASHES);

    System.out.println(DASHES);
    System.out.println("The example has completed. ");
    System.out.println("\n Thanks for watching!");
    System.out.println(DASHES);
}

// Deletes the AWS resources used in this example.
private static void deleteResources(LoadBalancer loadBalancer, AutoScaler
autoScaler, Database database)
    throws IOException, InterruptedException {
    loadBalancer.deleteLoadBalancer(lbName);
    System.out.println("*** Wait 30 secs for resource to be deleted");
    TimeUnit.SECONDS.sleep(30);
    loadBalancer.deleteTargetGroup(targetGroupName);
    autoScaler.deleteAutoScaleGroup(autoScalingGroupName);
    autoScaler.deleteRolesPolicies(policyName, roleName, profileName);
    autoScaler.deleteTemplate(templateName);
    database.deleteTable(tableName);
}

private static void deploy(LoadBalancer loadBalancer) throws
InterruptedException, IOException {
    Scanner in = new Scanner(System.in);
    System.out.println(
        """

            For this demo, we'll use the AWS SDK for Java (v2) to
create several AWS resources
            to set up a load-balanced web service endpoint and
explore some ways to make it resilient
            against various kinds of failures.

            Some of the resources create by this demo are:

```

```

        \t* A DynamoDB table that the web service depends on to
provide book, movie, and song recommendations.
        \t* An EC2 launch template that defines EC2 instances
that each contain a Python web server.
        \t* An EC2 Auto Scaling group that manages EC2 instances
across several Availability Zones.
        \t* An Elastic Load Balancing (ELB) load balancer that
targets the Auto Scaling group to distribute requests.
        """);

    System.out.println("Press Enter when you're ready.");
    in.nextLine();
    System.out.println(DASHES);

    System.out.println(DASHES);
    System.out.println("Creating and populating a DynamoDB table named " +
tableName);
    Database database = new Database();
    database.createTable(tableName, fileName);
    System.out.println(DASHES);

    System.out.println(DASHES);
    System.out.println("""
        Creating an EC2 launch template that runs '{startup_script}' when
an instance starts.
        This script starts a Python web server defined in the `server.py`
script. The web server
        listens to HTTP requests on port 80 and responds to requests to
`/` and to `/healthcheck`.
        For demo purposes, this server is run as the root user. In
production, the best practice is to
        run a web server, such as Apache, with least-privileged
credentials.

        The template also defines an IAM policy that each instance uses
to assume a role that grants
        permissions to access the DynamoDB recommendation table and
Systems Manager parameters
        that control the flow of the demo.
        """);

    LaunchTemplateCreator templateCreator = new LaunchTemplateCreator();
    templateCreator.createTemplate(policyFile, policyName, profileName,
startScript, templateName, roleName);

```

```
System.out.println(DASHES);

System.out.println(DASHES);
System.out.println(
    "Creating an EC2 Auto Scaling group that maintains three EC2
instances, each in a different Availability Zone.");
System.out.println("*** Wait 30 secs for the VPC to be created");
TimeUnit.SECONDS.sleep(30);
AutoScaler autoScaler = new AutoScaler();
String[] zones = autoScaler.createGroup(3, templateName,
autoScalingGroupName);

System.out.println("""
    At this point, you have EC2 instances created. Once each instance
starts, it listens for
    HTTP requests. You can see these instances in the console or
continue with the demo.
    Press Enter when you're ready to continue.
    """);

in.nextLine();
System.out.println(DASHES);

System.out.println(DASHES);
System.out.println("Creating variables that control the flow of the
demo.");
ParameterHelper paramHelper = new ParameterHelper();
paramHelper.reset();
System.out.println(DASHES);

System.out.println(DASHES);
System.out.println("""
    Creating an Elastic Load Balancing target group and load
balancer. The target group
    defines how the load balancer connects to instances. The load
balancer provides a
    single endpoint where clients connect and dispatches requests to
instances in the group.
    """);

String vpcId = autoScaler.getDefaultVPC();
List<Subnet> subnets = autoScaler.getSubnets(vpcId, zones);
System.out.println("You have retrieved a list with " + subnets.size() + "
subnets");
```

```

        String targetGroupArn = loadBalancer.createTargetGroup(protocol, port,
vpcId, targetGroupName);
        String elbDnsName = loadBalancer.createLoadBalancer(subnets,
targetGroupArn, lbName, port, protocol);
        autoScaler.attachLoadBalancerTargetGroup(autoScalingGroupName,
targetGroupArn);
        System.out.println("Verifying access to the load balancer endpoint...");
        boolean wasSuccessful =
loadBalancer.verifyLoadBalancerEndpoint(elbDnsName);
        if (!wasSuccessful) {
            System.out.println("Couldn't connect to the load balancer, verifying
that the port is open...");
            CloseableHttpClient httpClient = HttpClients.createDefault();

            // Create an HTTP GET request to "http://checkip.amazonaws.com"
            HttpGet httpGet = new HttpGet("http://checkip.amazonaws.com");
            try {
                // Execute the request and get the response
                HttpResponse response = httpClient.execute(httpGet);

                // Read the response content.
                String ipAddress =
IOUtils.toString(response.getEntity().getContent(),
StandardCharsets.UTF_8).trim();

                // Print the public IP address.
                System.out.println("Public IP Address: " + ipAddress);
                GroupInfo groupInfo = autoScaler.verifyInboundPort(vpcId, port,
ipAddress);
                if (!groupInfo.isPortOpen()) {
                    System.out.println("""
                        For this example to work, the default security group
for your default VPC must
                        allow access from this computer. You can either add
it automatically from this
                        example or add it yourself using the AWS Management
Console.
                        """);

                    System.out.println(
                        "Do you want to add a rule to security group " +
groupInfo.getGroupName() + " to allow");
                    System.out.println("inbound traffic on port " + port + " from
your computer's IP address (y/n) ");

```

```

        String ans = in.nextLine();
        if ("y".equalsIgnoreCase(ans)) {
            autoScaler.openInboundPort(groupInfo.getGroupName(),
String.valueOf(port), ipAddress);
            System.out.println("Security group rule added.");
        } else {
            System.out.println("No security group rule added.");
        }
    }

    } catch (AutoScalingException e) {
        e.printStackTrace();
    }
} else if (wasSuccessul) {
    System.out.println("Your load balancer is ready. You can access it by
browsing to:");
    System.out.println("\t http://" + elbDnsName);
} else {
    System.out.println("Couldn't get a successful response from the load
balancer endpoint. Troubleshoot by");
    System.out.println("manually verifying that your VPC and security
group are configured correctly and that");
    System.out.println("you can successfully make a GET request to the
load balancer.");
}

    System.out.println("Press Enter when you're ready to continue with the
demo.");
    in.nextLine();
}

// A method that controls the demo part of the Java program.
public static void demo(LoadBalancer loadBalancer) throws IOException,
InterruptedException {
    ParameterHelper paramHelper = new ParameterHelper();
    System.out.println("Read the ssm_only_policy.json file");
    String ssmOnlyPolicy = readFileAsString(ssmJSON);

    System.out.println("Resetting parameters to starting values for demo.");
    paramHelper.reset();

    System.out.println(
        """"

```

This part of the demonstration shows how to toggle different parts of the system to create situations where the web service fails, and shows how using a resilient architecture can keep the web service running in spite of these failures.

At the start, the load balancer endpoint returns recommendations and reports that all targets are healthy.

```
        """);
demoChoices(loadBalancer);
```

```
System.out.println(
    ""
```

The web service running on the EC2 instances gets recommendations by querying a DynamoDB table.

The table name is contained in a Systems Manager parameter named `self.param_helper.table`.

To simulate a failure of the recommendation service, let's set this parameter to name a non-existent table.

```
        """);
paramHelper.put(paramHelper.tableName, "this-is-not-a-table");
```

```
System.out.println(
    ""
```

\nNow, sending a GET request to the load balancer endpoint returns a failure code. But, the service reports as healthy to the load balancer because shallow health checks don't check for failure of the recommendation service.

```
        """);
demoChoices(loadBalancer);
```

```
System.out.println(
    ""
```

Instead of failing when the recommendation service fails, the web service can return a static response.

While this is not a perfect solution, it presents the customer with a somewhat better experience than failure.

```
        """);
paramHelper.put(paramHelper.failureResponse, "static");
```

```
System.out.println("""
```

Now, sending a GET request to the load balancer endpoint returns a static response.

```

        The service still reports as healthy because health checks are
still shallow.
        """);
demoChoices(loadBalancer);

System.out.println("Let's reinstate the recommendation service.");
paramHelper.put(paramHelper.tableName, paramHelper.dyntable);

System.out.println("""
        Let's also substitute bad credentials for one of the instances in
the target group so that it can't
        access the DynamoDB recommendation table. We will get an instance
id value.
        """);

LaunchTemplateCreator templateCreator = new LaunchTemplateCreator();
AutoScaler autoScaler = new AutoScaler();

// Create a new instance profile based on badCredsProfileName.
templateCreator.createInstanceProfile(policyFile, policyName,
badCredsProfileName, roleName);
String badInstanceId = autoScaler.getBadInstance(autoScalingGroupName);
System.out.println("The bad instance id values used for this demo is " +
badInstanceId);

String profileAssociationId =
autoScaler.getInstanceProfile(badInstanceId);
System.out.println("The association Id value is " +
profileAssociationId);
System.out.println("Replacing the profile for instance " + badInstanceId
+ " with a profile that contains bad credentials");
autoScaler.replaceInstanceProfile(badInstanceId, badCredsProfileName,
profileAssociationId);

System.out.println(
        ""
        Now, sending a GET request to the load balancer endpoint
returns either a recommendation or a static response,
        depending on which instance is selected by the load
balancer.
        """);

demoChoices(loadBalancer);

```



```
System.out.println("""
    Let's implement a deep health check. For this demo, a deep health
check tests whether
    the web service can access the DynamoDB table that it depends on
for recommendations. Note that
    the deep health check is only for ELB routing and not for Auto
Scaling instance health.
    This kind of deep health check is not recommended for Auto
Scaling instance health, because it
    risks accidental termination of all instances in the Auto Scaling
group when a dependent service fails.
    """);

System.out.println("""
    By implementing deep health checks, the load balancer can detect
when one of the instances is failing
    and take that instance out of rotation.
    """);

paramHelper.put(paramHelper.healthCheck, "deep");

System.out.println("""
    Now, checking target health indicates that the instance with bad
credentials
    is unhealthy. Note that it might take a minute or two for the
load balancer to detect the unhealthy
    instance. Sending a GET request to the load balancer endpoint
always returns a recommendation, because
    the load balancer takes unhealthy instances out of its rotation.
    """);

demoChoices(loadBalancer);

System.out.println(
    """
        Because the instances in this demo are controlled by an
auto scaler, the simplest way to fix an unhealthy
        instance is to terminate it and let the auto scaler start
a new instance to replace it.
    """);
autoScaler.terminateInstance(badInstanceId);

System.out.println("""
```

Even while the instance is terminating and the new instance is starting, sending a GET request to the web service continues to get a successful recommendation response because the load balancer routes requests to the healthy instances. After the replacement instance starts and reports as healthy, it is included in the load balancing rotation.

Note that terminating and replacing an instance typically takes several minutes, during which time you can see the changing health check status until the new instance is running and healthy.

```

        """);

        demoChoices(loadBalancer);
        System.out.println(
            "If the recommendation service fails now, deep health checks mean
            all instances report as unhealthy.");
        paramHelper.put(paramHelper.tableName, "this-is-not-a-table");

        demoChoices(loadBalancer);
        paramHelper.reset();
    }

    public static void demoChoices(LoadBalancer loadBalancer) throws IOException,
    InterruptedException {
        String[] actions = {
            "Send a GET request to the load balancer endpoint.",
            "Check the health of load balancer targets.",
            "Go to the next part of the demo."
        };

        Scanner scanner = new Scanner(System.in);

        while (true) {
            System.out.println("-".repeat(88));
            System.out.println("See the current state of the service by selecting
            one of the following choices:");
            for (int i = 0; i < actions.length; i++) {
                System.out.println(i + ": " + actions[i]);
            }

            try {
                System.out.print("\nWhich action would you like to take? ");
                int choice = scanner.nextInt();

```

```
System.out.println("-".repeat(88));

switch (choice) {
    case 0 -> {
        System.out.println("Request:\n");
        System.out.println("GET http://" +
loadBalancer.getEndpoint(lbName));
        CloseableHttpClient httpClient =
HttpClientClients.createDefault();

        // Create an HTTP GET request to the ELB.
        HttpGet httpGet = new HttpGet("http://" +
loadBalancer.getEndpoint(lbName));

        // Execute the request and get the response.
        HttpResponse response = httpClient.execute(httpGet);
        int statusCode =
response.getStatusLine().getStatusCode();
        System.out.println("HTTP Status Code: " + statusCode);

        // Display the JSON response
        BufferedReader reader = new BufferedReader(
            new
InputStreamReader(response.getEntity().getContent()));
        StringBuilder jsonResponse = new StringBuilder();
        String line;
        while ((line = reader.readLine()) != null) {
            jsonResponse.append(line);
        }
        reader.close();

        // Print the formatted JSON response.
        System.out.println("Full Response:\n");
        System.out.println(jsonResponse.toString());

        // Close the HTTP client.
        httpClient.close();
    }
    case 1 -> {
        System.out.println("\nChecking the health of load
balancer targets:\n");
        List<TargetHealthDescription> health =
loadBalancer.checkTargetHealth(targetGroupName);
```

```

        for (TargetHealthDescription target : health) {
            System.out.printf("\tTarget %s on port %d is %s\n",
target.target().id(),
                                target.target().port(),
target.targetHealth().stateAsString());
        }
        System.out.println("""
health check to update
                                Note that it can take a minute or two for the
                                after changes are made.
                                """);
    }
    case 2 -> {
        System.out.println("\nOkay, let's move on.");
        System.out.println("-".repeat(88));
        return; // Exit the method when choice is 2
    }
    default -> System.out.println("You must choose a value
between 0-2. Please select again.");
}

    } catch (java.util.InputMismatchException e) {
        System.out.println("Invalid input. Please select again.");
        scanner.nextLine(); // Clear the input buffer.
    }
}

public static String readFileAsString(String filePath) throws IOException {
    byte[] bytes = Files.readAllBytes(Paths.get(filePath));
    return new String(bytes);
}
}

```

Membuat kelas yang menggabungkan tindakan Penskalaan Otomatis dan Amazon EC2.

```

public class AutoScaler {

    private static Ec2Client ec2Client;
    private static AutoScalingClient autoScalingClient;
    private static IamClient iamClient;
}

```

```
private static SsmClient ssmClient;

private IAMClient getIAMClient() {
    if (iamClient == null) {
        iamClient = IAMClient.builder()
            .region(Region.US_EAST_1)
            .build();
    }
    return iamClient;
}

private SsmClient getSSMClient() {
    if (ssmClient == null) {
        ssmClient = SsmClient.builder()
            .region(Region.US_EAST_1)
            .build();
    }
    return ssmClient;
}

private EC2Client getEc2Client() {
    if (ec2Client == null) {
        ec2Client = EC2Client.builder()
            .region(Region.US_EAST_1)
            .build();
    }
    return ec2Client;
}

private AutoScalingClient getAutoScalingClient() {
    if (autoScalingClient == null) {
        autoScalingClient = AutoScalingClient.builder()
            .region(Region.US_EAST_1)
            .build();
    }
    return autoScalingClient;
}

/**
 * Terminates and instances in an EC2 Auto Scaling group. After an instance
is
 * terminated, it can no longer be accessed.
 */
public void terminateInstance(String instanceId) {
```

```
        TerminateInstanceInAutoScalingGroupRequest terminateInstanceIRequest =
TerminateInstanceInAutoScalingGroupRequest
            .builder()
            .instanceId(instanceId)
            .shouldDecrementDesiredCapacity(false)
            .build();

getAutoScalingClient().terminateInstanceInAutoScalingGroup(terminateInstanceIRequest);
    System.out.format("Terminated instance %s.", instanceId);
}

/**
 * Replaces the profile associated with a running instance. After the profile
is
 * replaced, the instance is rebooted to ensure that it uses the new profile.
 * When
 * the instance is ready, Systems Manager is used to restart the Python web
 * server.
 */
public void replaceInstanceProfile(String instanceId, String
newInstanceProfileName, String profileAssociationId)
    throws InterruptedException {
    // Create an IAM instance profile specification.
    software.amazon.awssdk.services.ec2.model.IamInstanceProfileSpecification
iamInstanceProfile =
software.amazon.awssdk.services.ec2.model.IamInstanceProfileSpecification
    .builder()
    .name(newInstanceProfileName) // Make sure
'newInstanceProfileName' is a valid IAM Instance Profile
        // name.
    .build();

    // Replace the IAM instance profile association for the EC2 instance.
    ReplaceIamInstanceProfileAssociationRequest replaceRequest =
ReplaceIamInstanceProfileAssociationRequest
    .builder()
    .iamInstanceProfile(iamInstanceProfile)
    .associationId(profileAssociationId) // Make sure
'profileAssociationId' is a valid association ID.
    .build();

    try {
        getEc2Client().replaceIamInstanceProfileAssociation(replaceRequest);
```

```
        // Handle the response as needed.
    } catch (Ec2Exception e) {
        // Handle exceptions, log, or report the error.
        System.err.println("Error: " + e.getMessage());
    }
    System.out.format("Replaced instance profile for association %s with
profile %s.", profileAssociationId,
        newInstanceProfileName);
    TimeUnit.SECONDS.sleep(15);
    boolean instReady = false;
    int tries = 0;

    // Reboot after 60 seconds
    while (!instReady) {
        if (tries % 6 == 0) {
            getEc2Client().rebootInstances(RebootInstancesRequest.builder()
                .instanceIds(instanceId)
                .build());
            System.out.println("Rebooting instance " + instanceId + " and
waiting for it to be ready.");
        }
        tries++;
        try {
            TimeUnit.SECONDS.sleep(10);
        } catch (InterruptedException e) {
            e.printStackTrace();
        }

        DescribeInstanceInformationResponse informationResponse =
getSSMClient().describeInstanceInformation();
        List<InstanceInformation> instanceInformationList =
informationResponse.getInstanceInformationList();
        for (InstanceInformation info : instanceInformationList) {
            if (info.getInstanceId().equals(instanceId)) {
                instReady = true;
                break;
            }
        }
    }

    SendCommandRequest sendCommandRequest = SendCommandRequest.builder()
        .instanceIds(instanceId)
        .documentName("AWS-RunShellScript")
        .parameters(Collections.singletonMap("commands",
```

```

        Collections.singletonList("cd / && sudo python3 server.py
80"))))
        .build();

        getSSMClient().sendCommand(sendCommandRequest);
        System.out.println("Restarted the Python web server on instance " +
instanceId + ".");
    }

    public void openInboundPort(String secGroupId, String port, String ipAddress)
    {
        AuthorizeSecurityGroupIngressRequest ingressRequest =
AuthorizeSecurityGroupIngressRequest.builder()
            .groupName(secGroupId)
            .cidrIp(ipAddress)
            .fromPort(Integer.parseInt(port))
            .build();

        getEc2Client().authorizeSecurityGroupIngress(ingressRequest);
        System.out.format("Authorized ingress to %s on port %s from %s.",
secGroupId, port, ipAddress);
    }

    /**
     * Detaches a role from an instance profile, detaches policies from the role,
     * and deletes all the resources.
     */
    public void deleteInstanceProfile(String roleName, String profileName) {
        try {
            software.amazon.awssdk.services.iam.model.GetInstanceProfileRequest
getInstanceProfileRequest =
software.amazon.awssdk.services.iam.model.GetInstanceProfileRequest
            .builder()
            .instanceProfileName(profileName)
            .build();

            GetInstanceProfileResponse response =
getIAMClient().getInstanceProfile(getInstanceProfileRequest);
            String name = response.getInstanceProfile().getInstanceProfileName();
            System.out.println(name);

            RemoveRoleFromInstanceProfileRequest profileRequest =
RemoveRoleFromInstanceProfileRequest.builder()
                .instanceProfileName(profileName)

```



```
        .roleName(roleName)
        .build();

        getIAMClient().removeRoleFromInstanceProfile(profileRequest);
        DeleteInstanceProfileRequest deleteInstanceProfileRequest =
DeleteInstanceProfileRequest.builder()
        .instanceProfileName(profileName)
        .build();

        getIAMClient().deleteInstanceProfile(deleteInstanceProfileRequest);
        System.out.println("Deleted instance profile " + profileName);

        DeleteRoleRequest deleteRoleRequest = DeleteRoleRequest.builder()
        .roleName(roleName)
        .build();

        // List attached role policies.
        ListAttachedRolePoliciesResponse rolesResponse = getIAMClient()
        .listAttachedRolePolicies(role -> role.roleName(roleName));
        List<AttachedPolicy> attachedPolicies =
rolesResponse.attachedPolicies();
        for (AttachedPolicy attachedPolicy : attachedPolicies) {
            DetachRolePolicyRequest request =
DetachRolePolicyRequest.builder()
            .roleName(roleName)
            .policyArn(attachedPolicy.policyArn())
            .build();

            getIAMClient().detachRolePolicy(request);
            System.out.println("Detached and deleted policy " +
attachedPolicy.policyName());
        }

        getIAMClient().deleteRole(deleteRoleRequest);
        System.out.println("Instance profile and role deleted.");

    } catch (IamException e) {
        System.err.println(e.getMessage());
        System.exit(1);
    }
}

public void deleteTemplate(String templateName) {
```

```
        getEc2Client().deleteLaunchTemplate(name ->
name.launchTemplateName(templateName));
        System.out.format(templateName + " was deleted.");
    }

    public void deleteAutoScaleGroup(String groupName) {
        DeleteAutoScalingGroupRequest deleteAutoScalingGroupRequest =
DeleteAutoScalingGroupRequest.builder()
            .autoScalingGroupName(groupName)
            .forceDelete(true)
            .build();

getAutoScalingClient().deleteAutoScalingGroup(deleteAutoScalingGroupRequest);
        System.out.println(groupName + " was deleted.");
    }

    /**
     * Verify the default security group of the specified VPC allows ingress from
     * this
     * computer. This can be done by allowing ingress from this computer's IP
     * address. In some situations, such as connecting from a corporate network,
you
     * must instead specify a prefix list ID. You can also temporarily open the
port
     * to
     * any IP address while running this example. If you do, be sure to remove
     * public
     * access when you're done.
     */
    public GroupInfo verifyInboundPort(String VPC, int port, String ipAddress) {
        boolean portIsOpen = false;
        GroupInfo groupInfo = new GroupInfo();
        try {
            Filter filter = Filter.builder()
                .name("group-name")
                .values("default")
                .build();

            Filter filter1 = Filter.builder()
                .name("vpc-id")
                .values(VPC)
                .build();
```

```

        DescribeSecurityGroupsRequest securityGroupsRequest =
DescribeSecurityGroupsRequest.builder()
        .filters(filter, filter1)
        .build();

        DescribeSecurityGroupsResponse securityGroupsResponse =
getEc2Client()
        .describeSecurityGroups(securityGroupsRequest);
        String securityGroup =
securityGroupsResponse.securityGroups().get(0).groupName();
        groupInfo.setGroupName(securityGroup);

        for (SecurityGroup secGroup :
securityGroupsResponse.securityGroups()) {
            System.out.println("Found security group: " +
secGroup.groupId());

            for (IpPermission ipPermission : secGroup.ipPermissions()) {
                if (ipPermission.fromPort() == port) {
                    System.out.println("Found inbound rule: " +
ipPermission);

                    for (IpRange ipRange : ipPermission.ipRanges()) {
                        String cidrIp = ipRange.cidrIp();
                        if (cidrIp.startsWith(ipAddress) ||
cidrIp.equals("0.0.0.0/0")) {
                            System.out.println(cidrIp + " is applicable");
                            portIsOpen = true;
                        }
                    }

                    if (!ipPermission.prefixListIds().isEmpty()) {
                        System.out.println("Prefix lList is applicable");
                        portIsOpen = true;
                    }

                    if (!portIsOpen) {
                        System.out
                            .println("The inbound rule does not appear to
be open to either this computer's IP,"
                                + " all IP addresses (0.0.0.0/0), or
to a prefix list ID.");
                    } else {
                        break;
                    }
                }
            }
        }
    }
}

```

```
        }
    }
}

} catch (AutoScalingException e) {
    System.err.println(e.awsErrorDetails().errorMessage());
}

groupInfo.setPortOpen(portIsOpen);
return groupInfo;
}

/*
 * Attaches an Elastic Load Balancing (ELB) target group to this EC2 Auto
 * Scaling group.
 * The target group specifies how the load balancer forward requests to the
 * instances
 * in the group.
 */
public void attachLoadBalancerTargetGroup(String asGroupName, String
targetGroupARN) {
    try {
        AttachLoadBalancerTargetGroupsRequest targetGroupsRequest =
AttachLoadBalancerTargetGroupsRequest.builder()
            .autoScalingGroupName(asGroupName)
            .targetGroupARNs(targetGroupARN)
            .build();

getAutoScalingClient().attachLoadBalancerTargetGroups(targetGroupsRequest);
        System.out.println("Attached load balancer to " + asGroupName);

    } catch (AutoScalingException e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
}

// Creates an EC2 Auto Scaling group with the specified size.
public String[] createGroup(int groupSize, String templateName, String
autoScalingGroupName) {

    // Get availability zones.
```

```
software.amazon.awssdk.services.ec2.model.DescribeAvailabilityZonesRequest
zonesRequest =
software.amazon.awssdk.services.ec2.model.DescribeAvailabilityZonesRequest
    .builder()
    .build();

    DescribeAvailabilityZonesResponse zonesResponse =
getEc2Client().describeAvailabilityZones(zonesRequest);
    List<String> availabilityZoneNames =
zonesResponse.availabilityZones().stream()

.map(software.amazon.awssdk.services.ec2.model.AvailabilityZone::zoneName)
    .collect(Collectors.toList());

    String availabilityZones = String.join(",", availabilityZoneNames);
    LaunchTemplateSpecification specification =
LaunchTemplateSpecification.builder()
    .launchTemplateName(templateName)
    .version("$Default")
    .build();

    String[] zones = availabilityZones.split(",");
    CreateAutoScalingGroupRequest groupRequest =
CreateAutoScalingGroupRequest.builder()
    .launchTemplate(specification)
    .availabilityZones(zones)
    .maxSize(groupSize)
    .minSize(groupSize)
    .autoScalingGroupName(autoScalingGroupName)
    .build();

    try {
        getAutoScalingClient().createAutoScalingGroup(groupRequest);

    } catch (AutoScalingException e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
    System.out.println("Created an EC2 Auto Scaling group named " +
autoScalingGroupName);
    return zones;
}
```

```
public String getDefaultVPC() {
    // Define the filter.
    Filter defaultFilter = Filter.builder()
        .name("is-default")
        .values("true")
        .build();

    software.amazon.awssdk.services.ec2.model.DescribeVpcsRequest request =
software.amazon.awssdk.services.ec2.model.DescribeVpcsRequest
        .builder()
        .filters(defaultFilter)
        .build();

    DescribeVpcsResponse response = getEc2Client().describeVpcs(request);
    return response.vpcs().get(0).vpcId();
}

// Gets the default subnets in a VPC for a specified list of Availability
Zones.
public List<Subnet> getSubnets(String vpcId, String[] availabilityZones) {
    List<Subnet> subnets = null;
    Filter vpcFilter = Filter.builder()
        .name("vpc-id")
        .values(vpcId)
        .build();

    Filter azFilter = Filter.builder()
        .name("availability-zone")
        .values(availabilityZones)
        .build();

    Filter defaultForAZ = Filter.builder()
        .name("default-for-az")
        .values("true")
        .build();

    DescribeSubnetsRequest request = DescribeSubnetsRequest.builder()
        .filters(vpcFilter, azFilter, defaultForAZ)
        .build();

    DescribeSubnetsResponse response =
getEc2Client().describeSubnets(request);
    subnets = response.subnets();
    return subnets;
}
```

```
}

// Gets data about the instances in the EC2 Auto Scaling group.
public String getBadInstance(String groupName) {
    DescribeAutoScalingGroupsRequest request =
DescribeAutoScalingGroupsRequest.builder()
        .autoScalingGroupNames(groupName)
        .build();

    DescribeAutoScalingGroupsResponse response =
getAutoScalingClient().describeAutoScalingGroups(request);
    AutoScalingGroup autoScalingGroup = response.autoScalingGroups().get(0);
    List<String> instanceIds = autoScalingGroup.instances().stream()
        .map(instance -> instance.instanceId())
        .collect(Collectors.toList());

    String[] instanceIdArray = instanceIds.toArray(new String[0]);
    for (String instanceId : instanceIdArray) {
        System.out.println("Instance ID: " + instanceId);
        return instanceId;
    }
    return "";
}

// Gets data about the profile associated with an instance.
public String getInstanceProfile(String instanceId) {
    Filter filter = Filter.builder()
        .name("instance-id")
        .values(instanceId)
        .build();

    DescribeIamInstanceProfileAssociationsRequest associationsRequest =
DescribeIamInstanceProfileAssociationsRequest
        .builder()
        .filters(filter)
        .build();

    DescribeIamInstanceProfileAssociationsResponse response = getEc2Client()
        .describeIamInstanceProfileAssociations(associationsRequest);
    return response.iamInstanceProfileAssociations().get(0).associationId();
}

public void deleteRolesPolicies(String policyName, String roleName, String
InstanceProfile) {
```

```

        ListPoliciesRequest listPoliciesRequest =
ListPoliciesRequest.builder().build();
        ListPoliciesResponse listPoliciesResponse =
getIAMClient().listPolicies(listPoliciesRequest);
        for (Policy policy : listPoliciesResponse.policies()) {
            if (policy.policyName().equals(policyName)) {
                // List the entities (users, groups, roles) that are attached to
the policy.

software.amazon.awssdk.services.iam.model.ListEntitiesForPolicyRequest
listEntitiesRequest =
software.amazon.awssdk.services.iam.model.ListEntitiesForPolicyRequest
                .builder()
                .policyArn(policy.arn())
                .build();
                ListEntitiesForPolicyResponse listEntitiesResponse = iamClient
                .listEntitiesForPolicy(listEntitiesRequest);
                if (!listEntitiesResponse.policyGroups().isEmpty() || !
listEntitiesResponse.policyUsers().isEmpty()
                || !listEntitiesResponse.policyRoles().isEmpty()) {
                    // Detach the policy from any entities it is attached to.
                    DetachRolePolicyRequest detachPolicyRequest =
DetachRolePolicyRequest.builder()
                        .policyArn(policy.arn())
                        .roleName(roleName) // Specify the name of the IAM
role

                        .build();

                    getIAMClient().detachRolePolicy(detachPolicyRequest);
                    System.out.println("Policy detached from entities.");
                }

                // Now, you can delete the policy.
                DeletePolicyRequest deletePolicyRequest =
DeletePolicyRequest.builder()
                    .policyArn(policy.arn())
                    .build();

                getIAMClient().deletePolicy(deletePolicyRequest);
                System.out.println("Policy deleted successfully.");
                break;
            }
        }
    }
}

```



```

        // List the roles associated with the instance profile
        ListInstanceProfilesForRoleRequest listRolesRequest =
ListInstanceProfilesForRoleRequest.builder()
        .roleName(roleName)
        .build();

        // Detach the roles from the instance profile
        ListInstanceProfilesForRoleResponse listRolesResponse =
iamClient.listInstanceProfilesForRole(listRolesRequest);
        for (software.amazon.awssdk.services.iam.model.InstanceProfile profile :
listRolesResponse.instanceProfiles()) {
            RemoveRoleFromInstanceProfileRequest removeRoleRequest =
RemoveRoleFromInstanceProfileRequest.builder()
                .instanceProfileName(InstanceProfile)
                .roleName(roleName) // Remove the extra dot here
                .build();

            getIAMClient().removeRoleFromInstanceProfile(removeRoleRequest);
            System.out.println("Role " + roleName + " removed from instance
profile " + InstanceProfile);
        }

        // Delete the instance profile after removing all roles
        DeleteInstanceProfileRequest deleteInstanceProfileRequest =
DeleteInstanceProfileRequest.builder()
            .instanceProfileName(InstanceProfile)
            .build();

        getIAMClient().deleteInstanceProfile(r ->
r.instanceProfileName(InstanceProfile));
        System.out.println(InstanceProfile + " Deleted");
        System.out.println("All roles and policies are deleted.");
    }
}

```

Membuat kelas yang menggabungkan tindakan Penyeimbangan Beban Elastis.

```

public class LoadBalancer {
    public ElasticLoadBalancingV2Client elasticLoadBalancingV2Client;

    public ElasticLoadBalancingV2Client getLoadBalancerClient() {
        if (elasticLoadBalancingV2Client == null) {

```

```
        elasticLoadBalancingV2Client = ElasticLoadBalancingV2Client.builder()
            .region(Region.US_EAST_1)
            .build();
    }

    return elasticLoadBalancingV2Client;
}

// Checks the health of the instances in the target group.
public List<TargetHealthDescription> checkTargetHealth(String
targetGroupName) {
    DescribeTargetGroupsRequest targetGroupsRequest =
DescribeTargetGroupsRequest.builder()
        .names(targetGroupName)
        .build();

    DescribeTargetGroupsResponse tgResponse =
getLoadBalancerClient().describeTargetGroups(targetGroupsRequest);

    DescribeTargetHealthRequest healthRequest =
DescribeTargetHealthRequest.builder()

.targetGroupArn(tgResponse.targetGroups().get(0).targetGroupArn())
        .build();

    DescribeTargetHealthResponse healthResponse =
getLoadBalancerClient().describeTargetHealth(healthRequest);
    return healthResponse.targetHealthDescriptions();
}

// Gets the HTTP endpoint of the load balancer.
public String getEndpoint(String lbName) {
    DescribeLoadBalancersResponse res = getLoadBalancerClient()
        .describeLoadBalancers(describe -> describe.names(lbName));
    return res.loadBalancers().get(0).dnsName();
}

// Deletes a load balancer.
public void deleteLoadBalancer(String lbName) {
    try {
        // Use a waiter to delete the Load Balancer.
        DescribeLoadBalancersResponse res = getLoadBalancerClient()
            .describeLoadBalancers(describe -> describe.names(lbName));
```

```

        ElasticLoadBalancingV2Waiter loadBalancerWaiter =
getLoadBalancerClient().waiter();
        DescribeLoadBalancersRequest request =
DescribeLoadBalancersRequest.builder()

.loadBalancerArns(res.loadBalancers().get(0).loadBalancerArn())
        .build();

        getLoadBalancerClient().deleteLoadBalancer(
            builder ->
builder.loadBalancerArn(res.loadBalancers().get(0).loadBalancerArn()));
        WaiterResponse<DescribeLoadBalancersResponse> waiterResponse =
loadBalancerWaiter
            .waitUntilLoadBalancersDeleted(request);
        waiterResponse.matched().response().ifPresent(System.out::println);

    } catch (ElasticLoadBalancingV2Exception e) {
        System.err.println(e.awsErrorDetails().errorMessage());
    }
    System.out.println(lbName + " was deleted.");
}

// Deletes the target group.
public void deleteTargetGroup(String targetGroupName) {
    try {
        DescribeTargetGroupsResponse res = getLoadBalancerClient()
            .describeTargetGroups(describe ->
describe.names(targetGroupName));
        getLoadBalancerClient()
            .deleteTargetGroup(builder ->
builder.targetGroupArn(res.targetGroups().get(0).targetGroupArn()));
    } catch (ElasticLoadBalancingV2Exception e) {
        System.err.println(e.awsErrorDetails().errorMessage());
    }
    System.out.println(targetGroupName + " was deleted.");
}

// Verify this computer can successfully send a GET request to the load
balancer
// endpoint.
public boolean verifyLoadBalancerEndpoint(String elbDnsName) throws
IOException, InterruptedException {
    boolean success = false;
    int retries = 3;

```

```
CloseableHttpClient httpClient = HttpClients.createDefault();

// Create an HTTP GET request to the ELB.
HttpGet httpGet = new HttpGet("http://" + elbDnsName);
try {
    while ((!success) && (retries > 0)) {
        // Execute the request and get the response.
        HttpResponse response = httpClient.execute(httpGet);
        int statusCode = response.getStatusLine().getStatusCode();
        System.out.println("HTTP Status Code: " + statusCode);
        if (statusCode == 200) {
            success = true;
        } else {
            retries--;
            System.out.println("Got connection error from load balancer
endpoint, retrying...");
            TimeUnit.SECONDS.sleep(15);
        }
    }

    } catch (org.apache.http.conn.HttpHostConnectException e) {
        System.out.println(e.getMessage());
    }

    System.out.println("Status.." + success);
    return success;
}

/*
 * Creates an Elastic Load Balancing target group. The target group specifies
 * how
 * the load balancer forward requests to instances in the group and how
instance
 * health is checked.
 */
public String createTargetGroup(String protocol, int port, String vpcId,
String targetGroupName) {
    CreateTargetGroupRequest targetGroupRequest =
CreateTargetGroupRequest.builder()
        .healthCheckPath("/healthcheck")
        .healthCheckTimeoutSeconds(5)
        .port(port)
        .vpcId(vpcId)
        .name(targetGroupName)
```

```
        .protocol(protocol)
        .build();

        CreateTargetGroupResponse targetGroupResponse =
getLoadBalancerClient().createTargetGroup(targetGroupRequest);
        String targetGroupArn =
targetGroupResponse.targetGroups().get(0).targetGroupArn();
        String targetGroup =
targetGroupResponse.targetGroups().get(0).targetGroupName();
        System.out.println("The " + targetGroup + " was created with ARN" +
targetGroupArn);
        return targetGroupArn;
    }

    /**
     * Creates an Elastic Load Balancing load balancer that uses the specified
     * subnets
     * and forwards requests to the specified target group.
     */
    public String createLoadBalancer(List<Subnet> subnetIds, String
targetGroupARN, String lbName, int port,
        String protocol) {
        try {
            List<String> subnetIdStrings = subnetIds.stream()
                .map(Subnet::subnetId)
                .collect(Collectors.toList());

            CreateLoadBalancerRequest balancerRequest =
CreateLoadBalancerRequest.builder()
                .subnets(subnetIdStrings)
                .name(lbName)
                .scheme("internet-facing")
                .build();

            // Create and wait for the load balancer to become available.
            CreateLoadBalancerResponse lsResponse =
getLoadBalancerClient().createLoadBalancer(balancerRequest);
            String lbARN = lsResponse.loadBalancers().get(0).loadBalancerArn();

            ElasticLoadBalancingV2Waiter loadBalancerWaiter =
getLoadBalancerClient().waiter();
            DescribeLoadBalancersRequest request =
DescribeLoadBalancersRequest.builder()
                .loadBalancerArns(lbARN)
```

```
        .build();

        System.out.println("Waiting for Load Balancer " + lbName + " to
become available.");
        WaiterResponse<DescribeLoadBalancersResponse> waiterResponse =
loadBalancerWaiter
            .waitUntilLoadBalancerAvailable(request);
        waiterResponse.matched().response().ifPresent(System.out::println);
        System.out.println("Load Balancer " + lbName + " is available.");

        // Get the DNS name (endpoint) of the load balancer.
        String lbDNSName = lsResponse.loadBalancers().get(0).dnsName();
        System.out.println("*** Load Balancer DNS Name: " + lbDNSName);

        // Create a listener for the load balance.
        Action action = Action.builder()
            .targetGroupArn(targetGroupARN)
            .type("forward")
            .build();

        CreateListenerRequest listenerRequest =
CreateListenerRequest.builder()

            .loadBalancerArn(lsResponse.loadBalancers().get(0).loadBalancerArn())
                .defaultActions(action)
                .port(port)
                .protocol(protocol)
                .defaultActions(action)
                .build();

        getLoadBalancerClient().createListener(listenerRequest);
        System.out.println("Created listener to forward traffic from load
balancer " + lbName + " to target group "
            + targetGroupARN);

        // Return the load balancer DNS name.
        return lbDNSName;

    } catch (ElasticLoadBalancingV2Exception e) {
        e.printStackTrace();
    }
    return "";
}
}
```

Membuat kelas yang menggunakan DynamoDB untuk menyimulasikan layanan yang direkomendasikan.

```
public class Database {

    private static DynamoDbClient dynamoDbClient;

    public static DynamoDbClient getDynamoDbClient() {
        if (dynamoDbClient == null) {
            dynamoDbClient = DynamoDbClient.builder()
                .region(Region.US_EAST_1)
                .build();
        }
        return dynamoDbClient;
    }

    // Checks to see if the Amazon DynamoDB table exists.
    private boolean doesTableExist(String tableName) {
        try {
            // Describe the table and catch any exceptions.
            DescribeTableRequest describeTableRequest =
DescribeTableRequest.builder()
                .tableName(tableName)
                .build();

            getDynamoDbClient().describeTable(describeTableRequest);
            System.out.println("Table '" + tableName + "' exists.");
            return true;

        } catch (ResourceNotFoundException e) {
            System.out.println("Table '" + tableName + "' does not exist.");
        } catch (DynamoDbException e) {
            System.err.println("Error checking table existence: " +
e.getMessage());
        }
        return false;
    }

    /*
     * Creates a DynamoDB table to use a recommendation service. The table has a
```

```
* hash key named 'MediaType' that defines the type of media recommended,
such
* as
* Book or Movie, and a range key named 'ItemId' that, combined with the
* MediaType,
* forms a unique identifier for the recommended item.
*/
public void createTable(String tableName, String fileName) throws IOException
{
    // First check to see if the table exists.
    boolean doesExist = doesTableExist(tableName);
    if (!doesExist) {
        DynamoDbWaiter dbWaiter = getDynamoDbClient().waiter();
        CreateTableRequest createTableRequest = CreateTableRequest.builder()
            .tableName(tableName)
            .attributeDefinitions(
                AttributeDefinition.builder()
                    .attributeName("MediaType")
                    .attributeType(ScalarAttributeType.S)
                    .build(),
                AttributeDefinition.builder()
                    .attributeName("ItemId")
                    .attributeType(ScalarAttributeType.N)
                    .build())
            .keySchema(
                KeySchemaElement.builder()
                    .attributeName("MediaType")
                    .keyType(KeyType.HASH)
                    .build(),
                KeySchemaElement.builder()
                    .attributeName("ItemId")
                    .keyType(KeyType.RANGE)
                    .build())
            .provisionedThroughput(
                ProvisionedThroughput.builder()
                    .readCapacityUnits(5L)
                    .writeCapacityUnits(5L)
                    .build())
            .build();

        getDynamoDbClient().createTable(createTableRequest);
        System.out.println("Creating table " + tableName + "...");

        // Wait until the Amazon DynamoDB table is created.
    }
}
```



```
DescribeTableRequest tableRequest = DescribeTableRequest.builder()
    .tableName(tableName)
    .build();

WaiterResponse<DescribeTableResponse> waiterResponse =
dbWaiter.waitUntilTableExists(tableRequest);
waiterResponse.matched().response().ifPresent(System.out::println);
System.out.println("Table " + tableName + " created.");

// Add records to the table.
populateTable(fileName, tableName);
}
}

public void deleteTable(String tableName) {
    getDynamoDbClient().deleteTable(table -> table.tableName(tableName));
    System.out.println("Table " + tableName + " deleted.");
}

// Populates the table with data located in a JSON file using the DynamoDB
// enhanced client.
public void populateTable(String fileName, String tableName) throws
IOException {
    DynamoDbEnhancedClient enhancedClient = DynamoDbEnhancedClient.builder()
        .dynamoDbClient(getDynamoDbClient())
        .build();
    ObjectMapper objectMapper = new ObjectMapper();
    File jsonFile = new File(fileName);
    JsonNode rootNode = objectMapper.readTree(jsonFile);

    DynamoDbTable<Recommendation> mappedTable =
enhancedClient.table(tableName,
        TableSchema.fromBean(Recommendation.class));
    for (JsonNode currentNode : rootNode) {
        String mediaType = currentNode.path("MediaType").path("S").asText();
        int itemId = currentNode.path("ItemId").path("N").asInt();
        String title = currentNode.path("Title").path("S").asText();
        String creator = currentNode.path("Creator").path("S").asText();

        // Create a Recommendation object and set its properties.
        Recommendation rec = new Recommendation();
        rec.setMediaType(mediaType);
        rec.setItemId(itemId);
        rec.setTitle(title);
    }
}
```

```
        rec.setCreator(creator);

        // Put the item into the DynamoDB table.
        mappedTable.putItem(rec); // Add the Recommendation to the list.
    }
    System.out.println("Added all records to the " + tableName);
}
}
```

Membuat kelas yang mengabungkan tindakan Systems Manager.

```
public class ParameterHelper {

    String tableName = "doc-example-resilient-architecture-table";
    String dyntable = "doc-example-recommendation-service";
    String failureResponse = "doc-example-resilient-architecture-failure-
response";
    String healthCheck = "doc-example-resilient-architecture-health-check";

    public void reset() {
        put(dyntable, tableName);
        put(failureResponse, "none");
        put(healthCheck, "shallow");
    }

    public void put(String name, String value) {
        SsmClient ssmClient = SsmClient.builder()
            .region(Region.US_EAST_1)
            .build();

        PutParameterRequest parameterRequest = PutParameterRequest.builder()
            .name(name)
            .value(value)
            .overwrite(true)
            .type("String")
            .build();

        ssmClient.putParameter(parameterRequest);
        System.out.printf("Setting demo parameter %s to '%s'.", name, value);
    }
}
```

- Untuk detail API, lihat topik berikut di Referensi API AWS SDK for Java 2.x .
 - [AttachLoadBalancerTargetGroups](#)
 - [CreateAutoScalingGroup](#)
 - [CreateInstanceProfile](#)
 - [CreateLaunchTemplate](#)
 - [CreateListener](#)
 - [CreateLoadBalancer](#)
 - [CreateTargetGroup](#)
 - [DeleteAutoScalingGroup](#)
 - [DeleteInstanceProfile](#)
 - [DeleteLaunchTemplate](#)
 - [DeleteLoadBalancer](#)
 - [DeleteTargetGroup](#)
 - [DescribeAutoScalingGroups](#)
 - [DescribeAvailabilityZones](#)
 - [DescribeInstanceProfileAssociations](#)
 - [DescribeInstances](#)
 - [DescribeLoadBalancers](#)
 - [DescribeSubnets](#)
 - [DescribeTargetGroups](#)
 - [DescribeTargetHealth](#)
 - [DescribeVpcs](#)
 - [RebootInstances](#)
 - [ReplacelamInstanceProfileAssociation](#)
 - [TerminateInstanceInAutoScalingGroup](#)
 - [UpdateAutoScalingGroup](#)

JavaScript

SDK untuk JavaScript (v3)

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturannya dan menjalankannya di [Repositori Contoh Kode AWS](#).

Menjalankan skenario interaktif di prompt perintah.

```
#!/usr/bin/env node
// Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
// SPDX-License-Identifier: Apache-2.0

import {
  Scenario,
  parseScenarioArgs,
} from "@aws-doc-sdk-examples/lib/scenario/index.js";

/**
 * The workflow steps are split into three stages:
 * - deploy
 * - demo
 * - destroy
 *
 * Each of these stages has a corresponding file prefixed with steps-*.
 */
import { deploySteps } from "./steps-deploy.js";
import { demoSteps } from "./steps-demo.js";
import { destroySteps } from "./steps-destroy.js";

/**
 * The context is passed to every scenario. Scenario steps
 * will modify the context.
 */
const context = {};

/**
 * Three Scenarios are created for the workflow. A Scenario is an orchestration
 class
```

```
* that simplifies running a series of steps.
*/
export const scenarios = {
  // Deploys all resources necessary for the workflow.
  deploy: new Scenario("Resilient Workflow - Deploy", deploySteps, context),
  // Demonstrates how a fragile web service can be made more resilient.
  demo: new Scenario("Resilient Workflow - Demo", demoSteps, context),
  // Destroys the resources created for the workflow.
  destroy: new Scenario("Resilient Workflow - Destroy", destroySteps, context),
};

// Call function if run directly
import { fileURLToPath } from "url";

if (process.argv[1] === fileURLToPath(import.meta.url)) {
  parseScenarioArgs(scenarios);
}
```

Menyusun langkah-langkah untuk men-deploy semua sumber daya.

```
// Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
// SPDX-License-Identifier: Apache-2.0
import { join } from "node:path";
import { readFileSync, writeFileSync } from "node:fs";
import axios from "axios";

import {
  BatchWriteItemCommand,
  CreateTableCommand,
  DynamoDBClient,
  waitUntilTableExists,
} from "@aws-sdk/client-dynamodb";
import {
  EC2Client,
  CreateKeyPairCommand,
  CreateLaunchTemplateCommand,
  DescribeAvailabilityZonesCommand,
  DescribeVpcsCommand,
  DescribeSubnetsCommand,
  DescribeSecurityGroupsCommand,
  AuthorizeSecurityGroupIngressCommand,
} from "@aws-sdk/client-ec2";
```

```
import {
  IAMClient,
  CreatePolicyCommand,
  CreateRoleCommand,
  CreateInstanceProfileCommand,
  AddRoleToInstanceProfileCommand,
  AttachRolePolicyCommand,
  waitUntilInstanceProfileExists,
} from "@aws-sdk/client-iam";
import { SSMClient, GetParameterCommand } from "@aws-sdk/client-ssm";
import {
  CreateAutoScalingGroupCommand,
  AutoScalingClient,
  AttachLoadBalancerTargetGroupsCommand,
} from "@aws-sdk/client-auto-scaling";
import {
  CreateListenerCommand,
  CreateLoadBalancerCommand,
  CreateTargetGroupCommand,
  ElasticLoadBalancingV2Client,
  waitUntilLoadBalancerAvailable,
} from "@aws-sdk/client-elastic-load-balancing-v2";

import {
  ScenarioOutput,
  ScenarioInput,
  ScenarioAction,
} from "@aws-doc-sdk-examples/lib/scenario/index.js";
import { retry } from "@aws-doc-sdk-examples/lib/utils/util-timers.js";

import { MESSAGES, NAMES, RESOURCES_PATH, ROOT } from "./constants.js";
import { initParamsSteps } from "./steps-reset-params.js";

/**
 * @type {import('@aws-doc-sdk-examples/lib/scenario.js').Step[]}
 */
export const deploySteps = [
  new ScenarioOutput("introduction", MESSAGES.introduction, { header: true }),
  new ScenarioInput("confirmDeployment", MESSAGES.confirmDeployment, {
    type: "confirm",
  }),
  new ScenarioAction(
    "handleConfirmDeployment",
    (c) => c.confirmDeployment === false && process.exit(),
  ),
];
```

```
),
new ScenarioOutput(
  "creatingTable",
  MESSAGES.creatingTable.replace("${TABLE_NAME}", NAMES.tableName),
),
new ScenarioAction("createTable", async () => {
  const client = new DynamoDBClient({});
  await client.send(
    new CreateTableCommand({
      TableName: NAMES.tableName,
      ProvisionedThroughput: {
        ReadCapacityUnits: 5,
        WriteCapacityUnits: 5,
      },
      AttributeDefinitions: [
        {
          AttributeName: "MediaType",
          AttributeType: "S",
        },
        {
          AttributeName: "ItemId",
          AttributeType: "N",
        },
      ],
      KeySchema: [
        {
          AttributeName: "MediaType",
          KeyType: "HASH",
        },
        {
          AttributeName: "ItemId",
          KeyType: "RANGE",
        },
      ],
    }),
  );
  await waitUntilTableExists({ client }, { TableName: NAMES.tableName });
}),
new ScenarioOutput(
  "createdTable",
  MESSAGES.createdTable.replace("${TABLE_NAME}", NAMES.tableName),
),
new ScenarioOutput(
  "populatingTable",
```

```

    MESSAGES.populatingTable.replace("${TABLE_NAME}", NAMES.tableName),
  ),
  new ScenarioAction("populateTable", () => {
    const client = new DynamoDBClient({});
    /**
     * @type {{ default: import("@aws-sdk/client-dynamodb").PutRequest['Item']
[] }}
    */
    const recommendations = JSON.parse(
      readFileSync(join(RESOURCES_PATH, "recommendations.json")),
    );

    return client.send(
      new BatchWriteItemCommand({
        RequestItems: {
          [NAMES.tableName]: recommendations.map((item) => ({
            PutRequest: { Item: item },
          })),
        },
      }),
    );
  }),
  new ScenarioOutput(
    "populatedTable",
    MESSAGES.populatedTable.replace("${TABLE_NAME}", NAMES.tableName),
  ),
  new ScenarioOutput(
    "creatingKeyPair",
    MESSAGES.creatingKeyPair.replace("${KEY_PAIR_NAME}", NAMES.keyPairName),
  ),
  new ScenarioAction("createKeyPair", async () => {
    const client = new EC2Client({});
    const { KeyMaterial } = await client.send(
      new CreateKeyPairCommand({
        KeyName: NAMES.keyPairName,
      }),
    );

    writeFileSync(`${NAMES.keyPairName}.pem`, KeyMaterial, { mode: 0o600 });
  }),
  new ScenarioOutput(
    "createdKeyPair",
    MESSAGES.createdKeyPair.replace("${KEY_PAIR_NAME}", NAMES.keyPairName),
  ),

```



```
new ScenarioOutput(
  "creatingInstancePolicy",
  MESSAGES.creatingInstancePolicy.replace(
    "${INSTANCE_POLICY_NAME}",
    NAMES.instancePolicyName,
  ),
),
new ScenarioAction("createInstancePolicy", async (state) => {
  const client = new IAMClient({});
  const {
    Policy: { Arn },
  } = await client.send(
    new CreatePolicyCommand({
      PolicyName: NAMES.instancePolicyName,
      PolicyDocument: readFileSync(
        join(RESOURCES_PATH, "instance_policy.json"),
      ),
    }),
  );
  state.instancePolicyArn = Arn;
}),
new ScenarioOutput("createdInstancePolicy", (state) =>
  MESSAGES.createdInstancePolicy
    .replace("${INSTANCE_POLICY_NAME}", NAMES.instancePolicyName)
    .replace("${INSTANCE_POLICY_ARN}", state.instancePolicyArn),
),
new ScenarioOutput(
  "creatingInstanceRole",
  MESSAGES.creatingInstanceRole.replace(
    "${INSTANCE_ROLE_NAME}",
    NAMES.instanceRoleName,
  ),
),
new ScenarioAction("createInstanceRole", () => {
  const client = new IAMClient({});
  return client.send(
    new CreateRoleCommand({
      RoleName: NAMES.instanceRoleName,
      AssumeRolePolicyDocument: readFileSync(
        join(ROOT, "assume-role-policy.json"),
      ),
    }),
  );
}),
```

```
new ScenarioOutput(
  "createdInstanceRole",
  MESSAGES.createdInstanceRole.replace(
    "${INSTANCE_ROLE_NAME}",
    NAMES.instanceRoleName,
  ),
),
new ScenarioOutput(
  "attachingPolicyToRole",
  MESSAGES.attachingPolicyToRole
    .replace("${INSTANCE_ROLE_NAME}", NAMES.instanceRoleName)
    .replace("${INSTANCE_POLICY_NAME}", NAMES.instancePolicyName),
),
new ScenarioAction("attachPolicyToRole", async (state) => {
  const client = new IAMClient({});
  await client.send(
    new AttachRolePolicyCommand({
      RoleName: NAMES.instanceRoleName,
      PolicyArn: state.instancePolicyArn,
    }),
  );
}),
new ScenarioOutput(
  "attachedPolicyToRole",
  MESSAGES.attachedPolicyToRole
    .replace("${INSTANCE_POLICY_NAME}", NAMES.instancePolicyName)
    .replace("${INSTANCE_ROLE_NAME}", NAMES.instanceRoleName),
),
new ScenarioOutput(
  "creatingInstanceProfile",
  MESSAGES.creatingInstanceProfile.replace(
    "${INSTANCE_PROFILE_NAME}",
    NAMES.instanceProfileName,
  ),
),
new ScenarioAction("createInstanceProfile", async (state) => {
  const client = new IAMClient({});
  const {
    InstanceProfile: { Arn },
  } = await client.send(
    new CreateInstanceProfileCommand({
      InstanceProfileName: NAMES.instanceProfileName,
    }),
  );
});
```

```
state.instanceProfileArn = Arn;

await waitUntilInstanceProfileExists(
  { client },
  { InstanceProfileName: NAMES.instanceProfileName },
);
}),
new ScenarioOutput("createdInstanceProfile", (state) =>
  MESSAGES.createdInstanceProfile
    .replace("${INSTANCE_PROFILE_NAME}", NAMES.instanceProfileName)
    .replace("${INSTANCE_PROFILE_ARN}", state.instanceProfileArn),
),
new ScenarioOutput(
  "addingRoleToInstanceProfile",
  MESSAGES.addingRoleToInstanceProfile
    .replace("${INSTANCE_PROFILE_NAME}", NAMES.instanceProfileName)
    .replace("${INSTANCE_ROLE_NAME}", NAMES.instanceRoleName),
),
new ScenarioAction("addRoleToInstanceProfile", () => {
  const client = new IAMClient({});
  return client.send(
    new AddRoleToInstanceProfileCommand({
      RoleName: NAMES.instanceRoleName,
      InstanceProfileName: NAMES.instanceProfileName,
    }),
  );
}),
new ScenarioOutput(
  "addedRoleToInstanceProfile",
  MESSAGES.addedRoleToInstanceProfile
    .replace("${INSTANCE_PROFILE_NAME}", NAMES.instanceProfileName)
    .replace("${INSTANCE_ROLE_NAME}", NAMES.instanceRoleName),
),
...initParamsSteps,
new ScenarioOutput("creatingLaunchTemplate", MESSAGES.creatingLaunchTemplate),
new ScenarioAction("createLaunchTemplate", async () => {
  // snippet-start:[javascript.v3.wkflw.resilient.CreateLaunchTemplate]
  const ssmClient = new SSMClient({});
  const { Parameter } = await ssmClient.send(
    new GetParameterCommand({
      Name: "/aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x86_64-gp2",
    }),
  );
});
const ec2Client = new EC2Client({});
```

```

    await ec2Client.send(
      new CreateLaunchTemplateCommand({
        LaunchTemplateName: NAMES.launchTemplateName,
        LaunchTemplateData: {
          InstanceType: "t3.micro",
          ImageId: Parameter.Value,
          IamInstanceProfile: { Name: NAMES.instanceProfileName },
          UserData: readFileSync(
            join(RESOURCES_PATH, "server_startup_script.sh"),
          ).toString("base64"),
          KeyName: NAMES.keyPairName,
        },
      }),
      // snippet-end:[javascript.v3.wkflw.resilient.CreateLaunchTemplate]
    );
  }),
  new ScenarioOutput(
    "createdLaunchTemplate",
    MESSAGES.createdLaunchTemplate.replace(
      "${LAUNCH_TEMPLATE_NAME}",
      NAMES.launchTemplateName,
    ),
  ),
  new ScenarioOutput(
    "creatingAutoScalingGroup",
    MESSAGES.creatingAutoScalingGroup.replace(
      "${AUTO_SCALING_GROUP_NAME}",
      NAMES.autoScalingGroupName,
    ),
  ),
  new ScenarioAction("createAutoScalingGroup", async (state) => {
    const ec2Client = new EC2Client({});
    const { AvailabilityZones } = await ec2Client.send(
      new DescribeAvailabilityZonesCommand({}),
    );
    state.availabilityZoneNames = AvailabilityZones.map((az) => az.ZoneName);
    const autoScalingClient = new AutoScalingClient({});
    await retry({ intervalInMs: 1000, maxRetries: 30 }, () =>
      autoScalingClient.send(
        new CreateAutoScalingGroupCommand({
          AvailabilityZones: state.availabilityZoneNames,
          AutoScalingGroupName: NAMES.autoScalingGroupName,
          LaunchTemplate: {
            LaunchTemplateName: NAMES.launchTemplateName,

```

```

        Version: "$Default",
    },
    MinSize: 3,
    MaxSize: 3,
  )),
),
);
}),
new ScenarioOutput(
  "createdAutoScalingGroup",
  /**
   * @param {{ availabilityZoneNames: string[] }} state
   */
  (state) =>
    MESSAGES.createdAutoScalingGroup
      .replace("${AUTO_SCALING_GROUP_NAME}", NAMES.autoScalingGroupName)
      .replace(
        "${AVAILABILITY_ZONE_NAMES}",
        state.availabilityZoneNames.join(", "),
      ),
  ),
),
new ScenarioInput("confirmContinue", MESSAGES.confirmContinue, {
  type: "confirm",
}),
new ScenarioOutput("loadBalancer", MESSAGES.loadBalancer),
new ScenarioOutput("gettingVpc", MESSAGES.gettingVpc),
new ScenarioAction("getVpc", async (state) => {
  // snippet-start:[javascript.v3.wkflw.resilient.DescribeVpcs]
  const client = new EC2Client({});
  const { Vpcs } = await client.send(
    new DescribeVpcsCommand({
      Filters: [{ Name: "is-default", Values: ["true"] }]},
    ),
  );
  // snippet-end:[javascript.v3.wkflw.resilient.DescribeVpcs]
  state.defaultVpc = Vpcs[0].VpcId;
}),
new ScenarioOutput("gotVpc", (state) =>
  MESSAGES.gotVpc.replace("${VPC_ID}", state.defaultVpc),
),
new ScenarioOutput("gettingSubnets", MESSAGES.gettingSubnets),
new ScenarioAction("getSubnets", async (state) => {
  // snippet-start:[javascript.v3.wkflw.resilient.DescribeSubnets]
  const client = new EC2Client({});

```

```
const { Subnets } = await client.send(
  new DescribeSubnetsCommand({
    Filters: [
      { Name: "vpc-id", Values: [state.defaultVpc] },
      { Name: "availability-zone", Values: state.availabilityZoneNames },
      { Name: "default-for-az", Values: ["true"] },
    ],
  }),
);
// snippet-end:[javascript.v3.wkflw.resilient.DescribeSubnets]
state.subnets = Subnets.map((subnet) => subnet.SubnetId);
}),
new ScenarioOutput(
  "gotSubnets",
  /**
   * @param {{ subnets: string[] }} state
   */
  (state) =>
    MESSAGES.gotSubnets.replace("${SUBNETS}", state.subnets.join(", ")),
),
new ScenarioOutput(
  "creatingLoadBalancerTargetGroup",
  MESSAGES.creatingLoadBalancerTargetGroup.replace(
    "${TARGET_GROUP_NAME}",
    NAMES.loadBalancerTargetGroupName,
  ),
),
new ScenarioAction("createLoadBalancerTargetGroup", async (state) => {
  // snippet-start:[javascript.v3.wkflw.resilient.CreateTargetGroup]
  const client = new ElasticLoadBalancingV2Client({});
  const { TargetGroups } = await client.send(
    new CreateTargetGroupCommand({
      Name: NAMES.loadBalancerTargetGroupName,
      Protocol: "HTTP",
      Port: 80,
      HealthCheckPath: "/healthcheck",
      HealthCheckIntervalSeconds: 10,
      HealthCheckTimeoutSeconds: 5,
      HealthyThresholdCount: 2,
      UnhealthyThresholdCount: 2,
      VpcId: state.defaultVpc,
    }),
  );
  // snippet-end:[javascript.v3.wkflw.resilient.CreateTargetGroup]
```

```
    const targetGroup = TargetGroups[0];
    state.targetGroupArn = targetGroup.TargetGroupArn;
    state.targetGroupProtocol = targetGroup.Protocol;
    state.targetGroupPort = targetGroup.Port;
  }},
  new ScenarioOutput(
    "createdLoadBalancerTargetGroup",
    MESSAGES.createdLoadBalancerTargetGroup.replace(
      "${TARGET_GROUP_NAME}",
      NAMES.loadBalancerTargetGroupName,
    ),
  ),
  new ScenarioOutput(
    "creatingLoadBalancer",
    MESSAGES.creatingLoadBalancer.replace("${LB_NAME}", NAMES.loadBalancerName),
  ),
  new ScenarioAction("createLoadBalancer", async (state) => {
    // snippet-start:[javascript.v3.wkflw.resilient.CreateLoadBalancer]
    const client = new ElasticLoadBalancingV2Client({});
    const { LoadBalancers } = await client.send(
      new CreateLoadBalancerCommand({
        Name: NAMES.loadBalancerName,
        Subnets: state.subnets,
      })),
    );
    state.loadBalancerDns = LoadBalancers[0].DNSName;
    state.loadBalancerArn = LoadBalancers[0].LoadBalancerArn;
    await waitUntilLoadBalancerAvailable(
      { client },
      { Names: [NAMES.loadBalancerName] },
    );
    // snippet-end:[javascript.v3.wkflw.resilient.CreateLoadBalancer]
  })),
  new ScenarioOutput("createdLoadBalancer", (state) =>
    MESSAGES.createdLoadBalancer
      .replace("${LB_NAME}", NAMES.loadBalancerName)
      .replace("${DNS_NAME}", state.loadBalancerDns),
  ),
  new ScenarioOutput(
    "creatingListener",
    MESSAGES.creatingLoadBalancerListener
      .replace("${LB_NAME}", NAMES.loadBalancerName)
      .replace("${TARGET_GROUP_NAME}", NAMES.loadBalancerTargetGroupName),
  ),
```

```
new ScenarioAction("createListener", async (state) => {
  // snippet-start:[javascript.v3.wkflw.resilient.CreateListener]
  const client = new ElasticLoadBalancingV2Client({});
  const { Listeners } = await client.send(
    new CreateListenerCommand({
      LoadBalancerArn: state.loadBalancerArn,
      Protocol: state.targetGroupProtocol,
      Port: state.targetGroupPort,
      DefaultActions: [
        { Type: "forward", TargetGroupArn: state.targetGroupArn },
      ],
    }),
  );
  // snippet-end:[javascript.v3.wkflw.resilient.CreateListener]
  const listener = Listeners[0];
  state.loadBalancerListenerArn = listener.ListenerArn;
}),
new ScenarioOutput("createdListener", (state) =>
  MESSAGES.createdLoadBalancerListener.replace(
    "${LB_LISTENER_ARN}",
    state.loadBalancerListenerArn,
  ),
),
new ScenarioOutput(
  "attachingLoadBalancerTargetGroup",
  MESSAGES.attachingLoadBalancerTargetGroup
    .replace("${TARGET_GROUP_NAME}", NAMES.loadBalancerTargetGroupName)
    .replace("${AUTO_SCALING_GROUP_NAME}", NAMES.autoScalingGroupName),
),
new ScenarioAction("attachLoadBalancerTargetGroup", async (state) => {
  // snippet-start:[javascript.v3.wkflw.resilient.AttachTargetGroup]
  const client = new AutoScalingClient({});
  await client.send(
    new AttachLoadBalancerTargetGroupsCommand({
      AutoScalingGroupName: NAMES.autoScalingGroupName,
      TargetGroupARNs: [state.targetGroupArn],
    }),
  );
  // snippet-end:[javascript.v3.wkflw.resilient.AttachTargetGroup]
}),
new ScenarioOutput(
  "attachedLoadBalancerTargetGroup",
  MESSAGES.attachedLoadBalancerTargetGroup,
),
```



```

new ScenarioOutput("verifyingInboundPort", MESSAGES.verifyingInboundPort),
new ScenarioAction(
  "verifyInboundPort",
  /**
   *
   * @param {{ defaultSecurityGroup: import('@aws-sdk/client-
ec2').SecurityGroup}} state
   */
  async (state) => {
    const client = new EC2Client({});
    const { SecurityGroups } = await client.send(
      new DescribeSecurityGroupsCommand({
        Filters: [{ Name: "group-name", Values: ["default"] }],
      }),
    );
    if (!SecurityGroups) {
      state.verifyInboundPortError = new Error(MESSAGES.noSecurityGroups);
    }
    state.defaultSecurityGroup = SecurityGroups[0];

    /**
     * @type {string}
     */
    const ipResponse = (await axios.get("http://checkip.amazonaws.com")).data;
    state.myIp = ipResponse.trim();
    const myIpRules = state.defaultSecurityGroup.IpPermissions.filter(
      ({ IpRanges }) =>
        IpRanges.some(
          ({ CidrIp }) =>
            CidrIp.startsWith(state.myIp) || CidrIp === "0.0.0.0/0",
        ),
    )
      .filter(({ IpProtocol }) => IpProtocol === "tcp")
      .filter(({ FromPort }) => FromPort === 80);

    state.myIpRules = myIpRules;
  },
),
new ScenarioOutput(
  "verifiedInboundPort",
  /**
   * @param {{ myIpRules: any[] }} state
   */
  (state) => {

```

```

    if (state.myIpRules.length > 0) {
      return MESSAGES.foundIpRules.replace(
        "${IP_RULES}",
        JSON.stringify(state.myIpRules, null, 2),
      );
    } else {
      return MESSAGES.noIpRules;
    }
  },
),
new ScenarioInput(
  "shouldAddInboundRule",
  /**
   * @param {{ myIpRules: any[] }} state
   */
  (state) => {
    if (state.myIpRules.length > 0) {
      return false;
    } else {
      return MESSAGES.noIpRules;
    }
  },
  { type: "confirm" },
),
new ScenarioAction(
  "addInboundRule",
  /**
   * @param {{ defaultSecurityGroup: import('@aws-sdk/client-ec2').SecurityGroup }} state
   */
  async (state) => {
    if (!state.shouldAddInboundRule) {
      return;
    }

    const client = new EC2Client({});
    await client.send(
      new AuthorizeSecurityGroupIngressCommand({
        GroupId: state.defaultSecurityGroup.GroupId,
        CidrIp: `${state.myIp}/32`,
        FromPort: 80,
        ToPort: 80,
        IpProtocol: "tcp",
      })),

```

```

    );
  },
),
new ScenarioOutput("addedInboundRule", (state) => {
  if (state.shouldAddInboundRule) {
    return MESSAGES.addedInboundRule.replace("${IP_ADDRESS}", state.myIp);
  } else {
    return false;
  }
}),
new ScenarioOutput("verifyingEndpoint", (state) =>
  MESSAGES.verifyingEndpoint.replace("${DNS_NAME}", state.loadBalancerDns),
),
new ScenarioAction("verifyEndpoint", async (state) => {
  try {
    const response = await retry({ intervalInMs: 2000, maxRetries: 30 }, () =>
      axios.get(`http://${state.loadBalancerDns}`),
    );
    state.endpointResponse = JSON.stringify(response.data, null, 2);
  } catch (e) {
    state.verifyEndpointError = e;
  }
}),
new ScenarioOutput("verifiedEndpoint", (state) => {
  if (state.verifyEndpointError) {
    console.error(state.verifyEndpointError);
  } else {
    return MESSAGES.verifiedEndpoint.replace(
      "${ENDPOINT_RESPONSE}",
      state.endpointResponse,
    );
  }
}),
];

```

Menyusun langkah-langkah untuk menjalankan demo.

```

// Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
// SPDX-License-Identifier: Apache-2.0
import { readFileSync } from "node:fs";
import { join } from "node:path";

```

```
import axios from "axios";

import {
  DescribeTargetGroupsCommand,
  DescribeTargetHealthCommand,
  ElasticLoadBalancingV2Client,
} from "@aws-sdk/client-elastic-load-balancing-v2";
import {
  DescribeInstanceInformationCommand,
  PutParameterCommand,
  SSMClient,
  SendCommandCommand,
} from "@aws-sdk/client-ssm";
import {
  IAMClient,
  CreatePolicyCommand,
  CreateRoleCommand,
  AttachRolePolicyCommand,
  CreateInstanceProfileCommand,
  AddRoleToInstanceProfileCommand,
  waitUntilInstanceProfileExists,
} from "@aws-sdk/client-iam";
import {
  AutoScalingClient,
  DescribeAutoScalingGroupsCommand,
  TerminateInstanceInAutoScalingGroupCommand,
} from "@aws-sdk/client-auto-scaling";
import {
  DescribeIamInstanceProfileAssociationsCommand,
  EC2Client,
  RebootInstancesCommand,
  ReplaceIamInstanceProfileAssociationCommand,
} from "@aws-sdk/client-ec2";

import {
  ScenarioAction,
  ScenarioInput,
  ScenarioOutput,
} from "@aws-doc-sdk-examples/lib/scenario/scenario.js";
import { retry } from "@aws-doc-sdk-examples/lib/utils/util-timers.js";

import { MESSAGES, NAMES, RESOURCES_PATH } from "./constants.js";
import { findLoadBalancer } from "./shared.js";
```

```
const getRecommendation = new ScenarioAction(
  "getRecommendation",
  async (state) => {
    const loadBalancer = await findLoadBalancer(NAMES.loadBalancerName);
    if (loadBalancer) {
      state.loadBalancerDnsName = loadBalancer.DNSName;
      try {
        state.recommendation = (
          await axios.get(`http://${state.loadBalancerDnsName}`)
        ).data;
      } catch (e) {
        state.recommendation = e instanceof Error ? e.message : e;
      }
    } else {
      throw new Error(MESSAGES.demoFindLoadBalancerError);
    }
  },
);
```

```
const getRecommendationResult = new ScenarioOutput(
  "getRecommendationResult",
  (state) =>
    `Recommendation:\n${JSON.stringify(state.recommendation, null, 2)}`,
  { preformatted: true },
);
```

```
const getHealthCheck = new ScenarioAction("getHealthCheck", async (state) => {
  // snippet-start:[javascript.v3.wkflw.resilient.DescribeTargetGroups]
  const client = new ElasticLoadBalancingV2Client({});
  const { TargetGroups } = await client.send(
    new DescribeTargetGroupsCommand({
      Names: [NAMES.loadBalancerTargetGroupName],
    }),
  );
  // snippet-end:[javascript.v3.wkflw.resilient.DescribeTargetGroups]

  // snippet-start:[javascript.v3.wkflw.resilient.DescribeTargetHealth]
  const { TargetHealthDescriptions } = await client.send(
    new DescribeTargetHealthCommand({
      TargetGroupArn: TargetGroups[0].TargetGroupArn,
    }),
  );
  // snippet-end:[javascript.v3.wkflw.resilient.DescribeTargetHealth]
  state.targetHealthDescriptions = TargetHealthDescriptions;
});
```

```
});

const getHealthCheckResult = new ScenarioOutput(
  "getHealthCheckResult",
  /**
   * @param {{ targetHealthDescriptions: import('@aws-sdk/client-elastic-load-
  balancing-v2').TargetHealthDescription[]}} state
   */
  (state) => {
    const status = state.targetHealthDescriptions
      .map((th) => `${th.Target.Id}: ${th.TargetHealth.State}`)
      .join("\n");
    return `Health check:\n${status}`;
  },
  { preformatted: true },
);

const loadBalancerLoop = new ScenarioAction(
  "loadBalancerLoop",
  getRecommendation.action,
  {
    whileConfig: {
      whileFn: ({ loadBalancerCheck }) => loadBalancerCheck,
      input: new ScenarioInput(
        "loadBalancerCheck",
        MESSAGES.demoLoadBalancerCheck,
        {
          type: "confirm",
        },
      ),
      output: getRecommendationResult,
    },
  },
);

const healthCheckLoop = new ScenarioAction(
  "healthCheckLoop",
  getHealthCheck.action,
  {
    whileConfig: {
      whileFn: ({ healthCheck }) => healthCheck,
      input: new ScenarioInput("healthCheck", MESSAGES.demoHealthCheck, {
        type: "confirm",
      }),
    },
  },
);
```

```
        output: getHealthCheckResult,
      },
    ],
  );

const statusSteps = [
  getRecommendation,
  getRecommendationResult,
  getHealthCheck,
  getHealthCheckResult,
];

/**
 * @type {import('@aws-doc-sdk-examples/lib/scenario.js').Step[]}
 */
export const demoSteps = [
  new ScenarioOutput("header", MESSAGES.demoHeader, { header: true }),
  new ScenarioOutput("sanityCheck", MESSAGES.demoSanityCheck),
  ...statusSteps,
  new ScenarioInput(
    "brokenDependencyConfirmation",
    MESSAGES.demoBrokenDependencyConfirmation,
    { type: "confirm" },
  ),
  new ScenarioAction("brokenDependency", async (state) => {
    if (!state.brokenDependencyConfirmation) {
      process.exit();
    } else {
      const client = new SSMClient({});
      state.badTableName = `fake-table-${Date.now()}`;
      await client.send(
        new PutParameterCommand({
          Name: NAMES.ssmTableNameKey,
          Value: state.badTableName,
          Overwrite: true,
          Type: "String",
        }),
      );
    }
  }),
  new ScenarioOutput("testBrokenDependency", (state) =>
    MESSAGES.demoTestBrokenDependency.replace(
      "${TABLE_NAME}",
      state.badTableName,
    ),
  ),
];
```

```
    ),
  ),
  ...statusSteps,
  new ScenarioInput(
    "staticResponseConfirmation",
    MESSAGES.demoStaticResponseConfirmation,
    { type: "confirm" },
  ),
  new ScenarioAction("staticResponse", async (state) => {
    if (!state.staticResponseConfirmation) {
      process.exit();
    } else {
      const client = new SSMClient({});
      await client.send(
        new PutParameterCommand({
          Name: NAMES.ssmFailureResponseKey,
          Value: "static",
          Overwrite: true,
          Type: "String",
        }),
      );
    }
  }),
  new ScenarioOutput("testStaticResponse", MESSAGES.demoTestStaticResponse),
  ...statusSteps,
  new ScenarioInput(
    "badCredentialsConfirmation",
    MESSAGES.demoBadCredentialsConfirmation,
    { type: "confirm" },
  ),
  new ScenarioAction("badCredentialsExit", (state) => {
    if (!state.badCredentialsConfirmation) {
      process.exit();
    }
  }),
  new ScenarioAction("fixDynamoDBName", async () => {
    const client = new SSMClient({});
    await client.send(
      new PutParameterCommand({
        Name: NAMES.ssmTableNameKey,
        Value: NAMES.tableName,
        Overwrite: true,
        Type: "String",
      }),
    ),
  ),
```



```

    );
  }),
  new ScenarioAction(
    "badCredentials",
    /**
     * @param {{ targetInstance: import('@aws-sdk/client-auto-
scaling').Instance }} state
     */
    async (state) => {
      await createSsmOnlyInstanceProfile();
      const autoScalingClient = new AutoScalingClient({});
      const { AutoScalingGroups } = await autoScalingClient.send(
        new DescribeAutoScalingGroupsCommand({
          AutoScalingGroupNames: [NAMES.autoScalingGroupName],
        }),
      );
      state.targetInstance = AutoScalingGroups[0].Instances[0];
      // snippet-start:
[javascript.v3.wkflw.resilient.DescribeIamInstanceProfileAssociations]
      const ec2Client = new EC2Client({});
      const { IamInstanceProfileAssociations } = await ec2Client.send(
        new DescribeIamInstanceProfileAssociationsCommand({
          Filters: [
            { Name: "instance-id", Values: [state.targetInstance.InstanceId] },
          ],
        }),
      );
      // snippet-end:
[javascript.v3.wkflw.resilient.DescribeIamInstanceProfileAssociations]
      state.instanceProfileAssociationId =
        IamInstanceProfileAssociations[0].AssociationId;
      // snippet-start:
[javascript.v3.wkflw.resilient.ReplaceIamInstanceProfileAssociation]
      await retry({ intervalInMs: 1000, maxRetries: 30 }, () =>
        ec2Client.send(
          new ReplaceIamInstanceProfileAssociationCommand({
            AssociationId: state.instanceProfileAssociationId,
            IamInstanceProfile: { Name: NAMES.ssmOnlyInstanceProfileName },
          }),
        ),
      );
      // snippet-end:
[javascript.v3.wkflw.resilient.ReplaceIamInstanceProfileAssociation]

```

```
await ec2Client.send(
  new RebootInstancesCommand({
    InstanceIds: [state.targetInstance.InstanceId],
  }),
);

const ssmClient = new SSMClient({});
await retry({ intervalInMs: 20000, maxRetries: 15 }, async () => {
  const { InstanceInformationList } = await ssmClient.send(
    new DescribeInstanceInformationCommand({}),
  );

  const instance = InstanceInformationList.find(
    (info) => info.InstanceId === state.targetInstance.InstanceId,
  );

  if (!instance) {
    throw new Error("Instance not found.");
  }
});

await ssmClient.send(
  new SendCommandCommand({
    InstanceIds: [state.targetInstance.InstanceId],
    DocumentName: "AWS-RunShellScript",
    Parameters: { commands: ["cd / && sudo python3 server.py 80"] },
  }),
);
},
),
new ScenarioOutput(
  "testBadCredentials",
  /**
   * @param {{ targetInstance: import('@aws-sdk/client-
  ssm').InstanceInformation}} state
   */
  (state) =>
    MESSAGES.demoTestBadCredentials.replace(
      "${INSTANCE_ID}",
      state.targetInstance.InstanceId,
    ),
),
loadBalancerLoop,
new ScenarioInput(
```

```

    "deepHealthCheckConfirmation",
    MESSAGES.demoDeepHealthCheckConfirmation,
    { type: "confirm" },
  ),
  new ScenarioAction("deepHealthCheckExit", (state) => {
    if (!state.deepHealthCheckConfirmation) {
      process.exit();
    }
  }),
  new ScenarioAction("deepHealthCheck", async () => {
    const client = new SSMClient({});
    await client.send(
      new PutParameterCommand({
        Name: NAMES.ssmHealthCheckKey,
        Value: "deep",
        Overwrite: true,
        Type: "String",
      }),
    );
  }),
  new ScenarioOutput("testDeepHealthCheck", MESSAGES.demoTestDeepHealthCheck),
  healthCheckLoop,
  loadBalancerLoop,
  new ScenarioInput(
    "killInstanceConfirmation",
    /**
     * @param {{ targetInstance: import('@aws-sdk/client-
    ssm').InstanceInformation }} state
     */
    (state) =>
      MESSAGES.demoKillInstanceConfirmation.replace(
        "${INSTANCE_ID}",
        state.targetInstance.InstanceId,
      ),
    { type: "confirm" },
  ),
  new ScenarioAction("killInstanceExit", (state) => {
    if (!state.killInstanceConfirmation) {
      process.exit();
    }
  }),
  new ScenarioAction(
    "killInstance",
    /**

```

```

    * @param {{ targetInstance: import('@aws-sdk/client-
    ssm').InstanceInformation }} state
    */
    async (state) => {
      const client = new AutoScalingClient({});
      await client.send(
        new TerminateInstanceInAutoScalingGroupCommand({
          InstanceId: state.targetInstance.InstanceId,
          ShouldDecrementDesiredCapacity: false,
        }),
      );
    },
  ),
  new ScenarioOutput("testKillInstance", MESSAGES.demoTestKillInstance),
  healthCheckLoop,
  loadBalancerLoop,
  new ScenarioInput("failOpenConfirmation", MESSAGES.demoFailOpenConfirmation, {
    type: "confirm",
  }),
  new ScenarioAction("failOpenExit", (state) => {
    if (!state.failOpenConfirmation) {
      process.exit();
    }
  }),
  new ScenarioAction("failOpen", () => {
    const client = new SSMClient({});
    return client.send(
      new PutParameterCommand({
        Name: NAMES.ssmTableNameKey,
        Value: `fake-table-${Date.now()}`,
        Overwrite: true,
        Type: "String",
      }),
    );
  }),
  new ScenarioOutput("testFailOpen", MESSAGES.demoFailOpenTest),
  healthCheckLoop,
  loadBalancerLoop,
  new ScenarioInput(
    "resetTableConfirmation",
    MESSAGES.demoResetTableConfirmation,
    { type: "confirm" },
  ),
  new ScenarioAction("resetTableExit", (state) => {

```

```
    if (!state.resetTableConfirmation) {
      process.exit();
    }
  })),
  new ScenarioAction("resetTable", async () => {
    const client = new SSMClient({});
    await client.send(
      new PutParameterCommand({
        Name: NAMES.ssmTableNameKey,
        Value: NAMES.tableName,
        Overwrite: true,
        Type: "String",
      }),
    );
  }),
  new ScenarioOutput("testResetTable", MESSAGES.demoTestResetTable),
  healthCheckLoop,
  loadBalancerLoop,
];

async function createSsmOnlyInstanceProfile() {
  const iamClient = new IAMClient({});
  const { Policy } = await iamClient.send(
    new CreatePolicyCommand({
      PolicyName: NAMES.ssmOnlyPolicyName,
      PolicyDocument: readFileSync(
        join(RESOURCES_PATH, "ssm_only_policy.json"),
      ),
    }),
  );
  await iamClient.send(
    new CreateRoleCommand({
      RoleName: NAMES.ssmOnlyRoleName,
      AssumeRolePolicyDocument: JSON.stringify({
        Version: "2012-10-17",
        Statement: [
          {
            Effect: "Allow",
            Principal: { Service: "ec2.amazonaws.com" },
            Action: "sts:AssumeRole",
          },
        ],
      }),
    }),
  );
}
```

```
);
await iamClient.send(
  new AttachRolePolicyCommand({
    RoleName: NAMES.ssmOnlyRoleName,
    PolicyArn: Policy.Arn,
  }),
);
await iamClient.send(
  new AttachRolePolicyCommand({
    RoleName: NAMES.ssmOnlyRoleName,
    PolicyArn: "arn:aws:iam::aws:policy/AmazonSSMManagedInstanceCore",
  }),
);
// snippet-start:[javascript.v3.wkflw.resilient.CreateInstanceProfile]
const { InstanceProfile } = await iamClient.send(
  new CreateInstanceProfileCommand({
    InstanceProfileName: NAMES.ssmOnlyInstanceProfileName,
  }),
);
await waitUntilInstanceProfileExists(
  { client: iamClient },
  { InstanceProfileName: NAMES.ssmOnlyInstanceProfileName },
);
// snippet-end:[javascript.v3.wkflw.resilient.CreateInstanceProfile]
await iamClient.send(
  new AddRoleToInstanceProfileCommand({
    InstanceProfileName: NAMES.ssmOnlyInstanceProfileName,
    RoleName: NAMES.ssmOnlyRoleName,
  }),
);

return InstanceProfile;
}
```

Menyusun langkah-langkah untuk menghancurkan semua sumber daya.

```
// Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
// SPDX-License-Identifier: Apache-2.0
import { unlinkSync } from "node:fs";

import { DynamoDBClient, DeleteTableCommand } from "@aws-sdk/client-dynamodb";
import {
```

```
    EC2Client,
    DeleteKeyPairCommand,
    DeleteLaunchTemplateCommand,
} from "@aws-sdk/client-ec2";
import {
    IAMClient,
    DeleteInstanceProfileCommand,
    RemoveRoleFromInstanceProfileCommand,
    DeletePolicyCommand,
    DeleteRoleCommand,
    DetachRolePolicyCommand,
    paginateListPolicies,
} from "@aws-sdk/client-iam";
import {
    AutoScalingClient,
    DeleteAutoScalingGroupCommand,
    TerminateInstanceInAutoScalingGroupCommand,
    UpdateAutoScalingGroupCommand,
    paginateDescribeAutoScalingGroups,
} from "@aws-sdk/client-auto-scaling";
import {
    DeleteLoadBalancerCommand,
    DeleteTargetGroupCommand,
    DescribeTargetGroupsCommand,
    ElasticLoadBalancingV2Client,
} from "@aws-sdk/client-elastic-load-balancing-v2";

import {
    ScenarioOutput,
    ScenarioInput,
    ScenarioAction,
} from "@aws-doc-sdk-examples/lib/scenario/index.js";
import { retry } from "@aws-doc-sdk-examples/lib/utils/util-timers.js";

import { MESSAGES, NAMES } from "./constants.js";
import { findLoadBalancer } from "./shared.js";

/**
 * @type {import('@aws-doc-sdk-examples/lib/scenario.js').Step[]}
 */
export const destroySteps = [
    new ScenarioInput("destroy", MESSAGES.destroy, { type: "confirm" }),
    new ScenarioAction(
        "abort",
```

```
(state) => state.destroy === false && process.exit(),
),
new ScenarioAction("deleteTable", async (c) => {
  try {
    const client = new DynamoDBClient({});
    await client.send(new DeleteTableCommand({ TableName: NAMES.tableName }));
  } catch (e) {
    c.deleteTableError = e;
  }
}),
new ScenarioOutput("deleteTableResult", (state) => {
  if (state.deleteTableError) {
    console.error(state.deleteTableError);
    return MESSAGES.deleteTableError.replace(
      "${TABLE_NAME}",
      NAMES.tableName,
    );
  } else {
    return MESSAGES.deletedTable.replace("${TABLE_NAME}", NAMES.tableName);
  }
}),
new ScenarioAction("deleteKeyPair", async (state) => {
  try {
    const client = new EC2Client({});
    await client.send(
      new DeleteKeyPairCommand({ KeyName: NAMES.keyPairName }),
    );
    unlinkSync(`${NAMES.keyPairName}.pem`);
  } catch (e) {
    state.deleteKeyPairError = e;
  }
}),
new ScenarioOutput("deleteKeyPairResult", (state) => {
  if (state.deleteKeyPairError) {
    console.error(state.deleteKeyPairError);
    return MESSAGES.deleteKeyPairError.replace(
      "${KEY_PAIR_NAME}",
      NAMES.keyPairName,
    );
  } else {
    return MESSAGES.deletedKeyPair.replace(
      "${KEY_PAIR_NAME}",
      NAMES.keyPairName,
    );
  }
});
```



```
    }
  })),
  new ScenarioAction("detachPolicyFromRole", async (state) => {
    try {
      const client = new IAMClient({});
      const policy = await findPolicy(NAMES.instancePolicyName);

      if (!policy) {
        state.detachPolicyFromRoleError = new Error(
          `Policy ${NAMES.instancePolicyName} not found.`
        );
      } else {
        await client.send(
          new DetachRolePolicyCommand({
            RoleName: NAMES.instanceRoleName,
            PolicyArn: policy.Arn,
          })
        );
      }
    } catch (e) {
      state.detachPolicyFromRoleError = e;
    }
  })),
  new ScenarioOutput("detachedPolicyFromRole", (state) => {
    if (state.detachPolicyFromRoleError) {
      console.error(state.detachPolicyFromRoleError);
      return MESSAGES.detachPolicyFromRoleError
        .replace("${INSTANCE_POLICY_NAME}", NAMES.instancePolicyName)
        .replace("${INSTANCE_ROLE_NAME}", NAMES.instanceRoleName);
    } else {
      return MESSAGES.detachedPolicyFromRole
        .replace("${INSTANCE_POLICY_NAME}", NAMES.instancePolicyName)
        .replace("${INSTANCE_ROLE_NAME}", NAMES.instanceRoleName);
    }
  })),
  new ScenarioAction("deleteInstancePolicy", async (state) => {
    const client = new IAMClient({});
    const policy = await findPolicy(NAMES.instancePolicyName);

    if (!policy) {
      state.deletePolicyError = new Error(
        `Policy ${NAMES.instancePolicyName} not found.`
      );
    } else {
```

```
    return client.send(
      new DeletePolicyCommand({
        PolicyArn: policy.Arn,
      }),
    );
  }
}),
new ScenarioOutput("deletePolicyResult", (state) => {
  if (state.deletePolicyError) {
    console.error(state.deletePolicyError);
    return MESSAGES.deletePolicyError.replace(
      "${INSTANCE_POLICY_NAME}",
      NAMES.instancePolicyName,
    );
  } else {
    return MESSAGES.deletedPolicy.replace(
      "${INSTANCE_POLICY_NAME}",
      NAMES.instancePolicyName,
    );
  }
}),
new ScenarioAction("removeRoleFromInstanceProfile", async (state) => {
  try {
    const client = new IAMClient({});
    await client.send(
      new RemoveRoleFromInstanceProfileCommand({
        RoleName: NAMES.instanceRoleName,
        InstanceProfileName: NAMES.instanceProfileName,
      }),
    );
  } catch (e) {
    state.removeRoleFromInstanceProfileError = e;
  }
}),
new ScenarioOutput("removeRoleFromInstanceProfileResult", (state) => {
  if (state.removeRoleFromInstanceProfile) {
    console.error(state.removeRoleFromInstanceProfileError);
    return MESSAGES.removeRoleFromInstanceProfileError
      .replace("${INSTANCE_PROFILE_NAME}", NAMES.instanceProfileName)
      .replace("${INSTANCE_ROLE_NAME}", NAMES.instanceRoleName);
  } else {
    return MESSAGES.removedRoleFromInstanceProfile
      .replace("${INSTANCE_PROFILE_NAME}", NAMES.instanceProfileName)
      .replace("${INSTANCE_ROLE_NAME}", NAMES.instanceRoleName);
  }
});
```

```
    }
  )),
  new ScenarioAction("deleteInstanceRole", async (state) => {
    try {
      const client = new IAMClient({});
      await client.send(
        new DeleteRoleCommand({
          RoleName: NAMES.instanceRoleName,
        }),
      );
    } catch (e) {
      state.deleteInstanceRoleError = e;
    }
  )),
  new ScenarioOutput("deleteInstanceRoleResult", (state) => {
    if (state.deleteInstanceRoleError) {
      console.error(state.deleteInstanceRoleError);
      return MESSAGES.deleteInstanceRoleError.replace(
        "${INSTANCE_ROLE_NAME}",
        NAMES.instanceRoleName,
      );
    } else {
      return MESSAGES.deletedInstanceRole.replace(
        "${INSTANCE_ROLE_NAME}",
        NAMES.instanceRoleName,
      );
    }
  )),
  new ScenarioAction("deleteInstanceProfile", async (state) => {
    try {
      // snippet-start:[javascript.v3.wkflw.resilient.DeleteInstanceProfile]
      const client = new IAMClient({});
      await client.send(
        new DeleteInstanceProfileCommand({
          InstanceProfileName: NAMES.instanceProfileName,
        }),
      );
      // snippet-end:[javascript.v3.wkflw.resilient.DeleteInstanceProfile]
    } catch (e) {
      state.deleteInstanceProfileError = e;
    }
  )),
  new ScenarioOutput("deleteInstanceProfileResult", (state) => {
    if (state.deleteInstanceProfileError) {
```

```
    console.error(state.deleteInstanceProfileError);
    return MESSAGES.deleteInstanceProfileError.replace(
      "${INSTANCE_PROFILE_NAME}",
      NAMES.instanceProfileName,
    );
  } else {
    return MESSAGES.deletedInstanceProfile.replace(
      "${INSTANCE_PROFILE_NAME}",
      NAMES.instanceProfileName,
    );
  }
}),
new ScenarioAction("deleteAutoScalingGroup", async (state) => {
  try {
    await terminateGroupInstances(NAMES.autoScalingGroupName);
    await retry({ intervalInMs: 60000, maxRetries: 60 }, async () => {
      await deleteAutoScalingGroup(NAMES.autoScalingGroupName);
    });
  } catch (e) {
    state.deleteAutoScalingGroupError = e;
  }
}),
new ScenarioOutput("deleteAutoScalingGroupResult", (state) => {
  if (state.deleteAutoScalingGroupError) {
    console.error(state.deleteAutoScalingGroupError);
    return MESSAGES.deleteAutoScalingGroupError.replace(
      "${AUTO_SCALING_GROUP_NAME}",
      NAMES.autoScalingGroupName,
    );
  } else {
    return MESSAGES.deletedAutoScalingGroup.replace(
      "${AUTO_SCALING_GROUP_NAME}",
      NAMES.autoScalingGroupName,
    );
  }
}),
new ScenarioAction("deleteLaunchTemplate", async (state) => {
  const client = new EC2Client({});
  try {
    // snippet-start:[javascript.v3.wkflw.resilient.DeleteLaunchTemplate]
    await client.send(
      new DeleteLaunchTemplateCommand({
        LaunchTemplateName: NAMES.launchTemplateName,
      }),
    );
  }
});
```

```
    );
    // snippet-end:[javascript.v3.wkflw.resilient.DeleteLaunchTemplate]
  } catch (e) {
    state.deleteLaunchTemplateError = e;
  }
}),
new ScenarioOutput("deleteLaunchTemplateResult", (state) => {
  if (state.deleteLaunchTemplateError) {
    console.error(state.deleteLaunchTemplateError);
    return MESSAGES.deleteLaunchTemplateError.replace(
      "${LAUNCH_TEMPLATE_NAME}",
      NAMES.launchTemplateName,
    );
  } else {
    return MESSAGES.deletedLaunchTemplate.replace(
      "${LAUNCH_TEMPLATE_NAME}",
      NAMES.launchTemplateName,
    );
  }
}),
new ScenarioAction("deleteLoadBalancer", async (state) => {
  try {
    // snippet-start:[javascript.v3.wkflw.resilient.DeleteLoadBalancer]
    const client = new ElasticLoadBalancingV2Client({});
    const loadBalancer = await findLoadBalancer(NAMES.loadBalancerName);
    await client.send(
      new DeleteLoadBalancerCommand({
        LoadBalancerArn: loadBalancer.LoadBalancerArn,
      }),
    );
    await retry({ intervalInMs: 1000, maxRetries: 60 }, async () => {
      const lb = await findLoadBalancer(NAMES.loadBalancerName);
      if (lb) {
        throw new Error("Load balancer still exists.");
      }
    });
    // snippet-end:[javascript.v3.wkflw.resilient.DeleteLoadBalancer]
  } catch (e) {
    state.deleteLoadBalancerError = e;
  }
}),
new ScenarioOutput("deleteLoadBalancerResult", (state) => {
  if (state.deleteLoadBalancerError) {
    console.error(state.deleteLoadBalancerError);
  }
});
```

```
        return MESSAGES.deleteLoadBalancerError.replace(
            "${LB_NAME}",
            NAMES.loadBalancerName,
        );
    } else {
        return MESSAGES.deletedLoadBalancer.replace(
            "${LB_NAME}",
            NAMES.loadBalancerName,
        );
    }
}),
new ScenarioAction("deleteLoadBalancerTargetGroup", async (state) => {
    // snippet-start:[javascript.v3.wkflw.resilient.DeleteTargetGroup]
    const client = new ElasticLoadBalancingV2Client({});
    try {
        const { TargetGroups } = await client.send(
            new DescribeTargetGroupsCommand({
                Names: [NAMES.loadBalancerTargetGroupName],
            }),
        );
        await retry({ intervalInMs: 1000, maxRetries: 30 }, () =>
            client.send(
                new DeleteTargetGroupCommand({
                    TargetGroupArn: TargetGroups[0].TargetGroupArn,
                }),
            ),
        );
    } catch (e) {
        state.deleteLoadBalancerTargetGroupError = e;
    }
    // snippet-end:[javascript.v3.wkflw.resilient.DeleteTargetGroup]
}),
new ScenarioOutput("deleteLoadBalancerTargetGroupResult", (state) => {
    if (state.deleteLoadBalancerTargetGroupError) {
        console.error(state.deleteLoadBalancerTargetGroupError);
        return MESSAGES.deleteLoadBalancerTargetGroupError.replace(
            "${TARGET_GROUP_NAME}",
            NAMES.loadBalancerTargetGroupName,
        );
    } else {
        return MESSAGES.deletedLoadBalancerTargetGroup.replace(
            "${TARGET_GROUP_NAME}",
            NAMES.loadBalancerTargetGroupName,
        );
    }
});
```

```
    );
  }
 )),
  new ScenarioAction("detachSsmOnlyRoleFromProfile", async (state) => {
    try {
      const client = new IAMClient({});
      await client.send(
        new RemoveRoleFromInstanceProfileCommand({
          InstanceProfileName: NAMES.ssmOnlyInstanceProfileName,
          RoleName: NAMES.ssmOnlyRoleName,
        }),
      );
    } catch (e) {
      state.detachSsmOnlyRoleFromProfileError = e;
    }
  }),
  new ScenarioOutput("detachSsmOnlyRoleFromProfileResult", (state) => {
    if (state.detachSsmOnlyRoleFromProfileError) {
      console.error(state.detachSsmOnlyRoleFromProfileError);
      return MESSAGES.detachSsmOnlyRoleFromProfileError
        .replace("${ROLE_NAME}", NAMES.ssmOnlyRoleName)
        .replace("${PROFILE_NAME}", NAMES.ssmOnlyInstanceProfileName);
    } else {
      return MESSAGES.detachedSsmOnlyRoleFromProfile
        .replace("${ROLE_NAME}", NAMES.ssmOnlyRoleName)
        .replace("${PROFILE_NAME}", NAMES.ssmOnlyInstanceProfileName);
    }
  }),
  new ScenarioAction("detachSsmOnlyCustomRolePolicy", async (state) => {
    try {
      const iamClient = new IAMClient({});
      const ssmOnlyPolicy = await findPolicy(NAMES.ssmOnlyPolicyName);
      await iamClient.send(
        new DetachRolePolicyCommand({
          RoleName: NAMES.ssmOnlyRoleName,
          PolicyArn: ssmOnlyPolicy.Arn,
        }),
      );
    } catch (e) {
      state.detachSsmOnlyCustomRolePolicyError = e;
    }
  }),
  new ScenarioOutput("detachSsmOnlyCustomRolePolicyResult", (state) => {
    if (state.detachSsmOnlyCustomRolePolicyError) {
```

```
    console.error(state.detachSsmOnlyCustomRolePolicyError);
    return MESSAGES.detachSsmOnlyCustomRolePolicyError
      .replace("${ROLE_NAME}", NAMES.ssmOnlyRoleName)
      .replace("${POLICY_NAME}", NAMES.ssmOnlyPolicyName);
  } else {
    return MESSAGES.detachedSsmOnlyCustomRolePolicy
      .replace("${ROLE_NAME}", NAMES.ssmOnlyRoleName)
      .replace("${POLICY_NAME}", NAMES.ssmOnlyPolicyName);
  }
}),
new ScenarioAction("detachSsmOnlyAWSRolePolicy", async (state) => {
  try {
    const iamClient = new IAMClient({});
    await iamClient.send(
      new DetachRolePolicyCommand({
        RoleName: NAMES.ssmOnlyRoleName,
        PolicyArn: "arn:aws:iam::aws:policy/AmazonSSMManagedInstanceCore",
      }),
    );
  } catch (e) {
    state.detachSsmOnlyAWSRolePolicyError = e;
  }
}),
new ScenarioOutput("detachSsmOnlyAWSRolePolicyResult", (state) => {
  if (state.detachSsmOnlyAWSRolePolicyError) {
    console.error(state.detachSsmOnlyAWSRolePolicyError);
    return MESSAGES.detachSsmOnlyAWSRolePolicyError
      .replace("${ROLE_NAME}", NAMES.ssmOnlyRoleName)
      .replace("${POLICY_NAME}", "AmazonSSMManagedInstanceCore");
  } else {
    return MESSAGES.detachedSsmOnlyAWSRolePolicy
      .replace("${ROLE_NAME}", NAMES.ssmOnlyRoleName)
      .replace("${POLICY_NAME}", "AmazonSSMManagedInstanceCore");
  }
}),
new ScenarioAction("deleteSsmOnlyInstanceProfile", async (state) => {
  try {
    const iamClient = new IAMClient({});
    await iamClient.send(
      new DeleteInstanceProfileCommand({
        InstanceProfileName: NAMES.ssmOnlyInstanceProfileName,
      }),
    );
  } catch (e) {
```



```
    state.deleteSsmOnlyInstanceProfileError = e;
  }
}),
new ScenarioOutput("deleteSsmOnlyInstanceProfileResult", (state) => {
  if (state.deleteSsmOnlyInstanceProfileError) {
    console.error(state.deleteSsmOnlyInstanceProfileError);
    return MESSAGES.deleteSsmOnlyInstanceProfileError.replace(
      "${INSTANCE_PROFILE_NAME}",
      NAMES.ssmOnlyInstanceProfileName,
    );
  } else {
    return MESSAGES.deletedSsmOnlyInstanceProfile.replace(
      "${INSTANCE_PROFILE_NAME}",
      NAMES.ssmOnlyInstanceProfileName,
    );
  }
}),
new ScenarioAction("deleteSsmOnlyPolicy", async (state) => {
  try {
    const iamClient = new IAMClient({});
    const ssmOnlyPolicy = await findPolicy(NAMES.ssmOnlyPolicyName);
    await iamClient.send(
      new DeletePolicyCommand({
        PolicyArn: ssmOnlyPolicy.Arn,
      }),
    );
  } catch (e) {
    state.deleteSsmOnlyPolicyError = e;
  }
}),
new ScenarioOutput("deleteSsmOnlyPolicyResult", (state) => {
  if (state.deleteSsmOnlyPolicyError) {
    console.error(state.deleteSsmOnlyPolicyError);
    return MESSAGES.deleteSsmOnlyPolicyError.replace(
      "${POLICY_NAME}",
      NAMES.ssmOnlyPolicyName,
    );
  } else {
    return MESSAGES.deletedSsmOnlyPolicy.replace(
      "${POLICY_NAME}",
      NAMES.ssmOnlyPolicyName,
    );
  }
}),
}),
```

```
new ScenarioAction("deleteSsmOnlyRole", async (state) => {
  try {
    const iamClient = new IAMClient({});
    await iamClient.send(
      new DeleteRoleCommand({
        RoleName: NAMES.ssmOnlyRoleName,
      }),
    );
  } catch (e) {
    state.deleteSsmOnlyRoleError = e;
  }
}),
new ScenarioOutput("deleteSsmOnlyRoleResult", (state) => {
  if (state.deleteSsmOnlyRoleError) {
    console.error(state.deleteSsmOnlyRoleError);
    return MESSAGES.deleteSsmOnlyRoleError.replace(
      "${ROLE_NAME}",
      NAMES.ssmOnlyRoleName,
    );
  } else {
    return MESSAGES.deletedSsmOnlyRole.replace(
      "${ROLE_NAME}",
      NAMES.ssmOnlyRoleName,
    );
  }
}),
];

/**
 * @param {string} policyName
 */
async function findPolicy(policyName) {
  const client = new IAMClient({});
  const paginatedPolicies = paginateListPolicies({ client }, {});
  for await (const page of paginatedPolicies) {
    const policy = page.Policies.find((p) => p.PolicyName === policyName);
    if (policy) {
      return policy;
    }
  }
}

/**
 * @param {string} groupName
```

```
*/
async function deleteAutoScalingGroup(groupName) {
  const client = new AutoScalingClient({});
  try {
    await client.send(
      new DeleteAutoScalingGroupCommand({
        AutoScalingGroupName: groupName,
      }),
    );
  } catch (err) {
    if (!(err instanceof Error)) {
      throw err;
    } else {
      console.log(err.name);
      throw err;
    }
  }
}

/**
 * @param {string} groupName
 */
async function terminateGroupInstances(groupName) {
  const autoScalingClient = new AutoScalingClient({});
  const group = await findAutoScalingGroup(groupName);
  await autoScalingClient.send(
    new UpdateAutoScalingGroupCommand({
      AutoScalingGroupName: group.AutoScalingGroupName,
      MinSize: 0,
    }),
  );
  for (const i of group.Instances) {
    await retry({ intervalInMs: 1000, maxRetries: 30 }, () =>
      autoScalingClient.send(
        new TerminateInstanceInAutoScalingGroupCommand({
          InstanceId: i.InstanceId,
          ShouldDecrementDesiredCapacity: true,
        }),
      ),
    );
  }
}

async function findAutoScalingGroup(groupName) {
```

```
const client = new AutoScalingClient({});
const paginatedGroups = paginateDescribeAutoScalingGroups({ client }, {});
for await (const page of paginatedGroups) {
  const group = page.AutoScalingGroups.find(
    (g) => g.AutoScalingGroupName === groupName,
  );
  if (group) {
    return group;
  }
}
throw new Error(`Auto scaling group ${groupName} not found.`);
}
```

- Untuk detail API, lihat topik berikut di Referensi API AWS SDK for JavaScript .
 - [AttachLoadBalancerTargetGroups](#)
 - [CreateAutoScalingGroup](#)
 - [CreateInstanceProfile](#)
 - [CreateLaunchTemplate](#)
 - [CreateListener](#)
 - [CreateLoadBalancer](#)
 - [CreateTargetGroup](#)
 - [DeleteAutoScalingGroup](#)
 - [DeleteInstanceProfile](#)
 - [DeleteLaunchTemplate](#)
 - [DeleteLoadBalancer](#)
 - [DeleteTargetGroup](#)
 - [DescribeAutoScalingGroups](#)
 - [DescribeAvailabilityZones](#)
 - [DescribeInstanceProfileAssociations](#)
 - [DescribeInstances](#)
 - [DescribeLoadBalancers](#)
 - [DescribeSubnets](#)
 - [DescribeTargetGroups](#)
 - [DescribeTargetHealth](#)

- [DescribeVpcs](#)
- [RebootInstances](#)
- [ReplacelamInstanceProfileAssociation](#)
- [TerminateInstanceInAutoScalingGroup](#)
- [UpdateAutoScalingGroup](#)

Python

SDK untuk Python (Boto3)

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturannya dan menjalankannya di [Repositori Contoh Kode AWS](#).

Menjalankan skenario interaktif di prompt perintah.

```
class Runner:
    def __init__(
        self, resource_path, recommendation, autoscaler, loadbalancer,
        param_helper
    ):
        self.resource_path = resource_path
        self.recommendation = recommendation
        self.autoscaler = autoscaler
        self.loadbalancer = loadbalancer
        self.param_helper = param_helper
        self.protocol = "HTTP"
        self.port = 80
        self.ssh_port = 22

    def deploy(self):
        recommendations_path = f"{self.resource_path}/recommendations.json"
        startup_script = f"{self.resource_path}/server_startup_script.sh"
        instance_policy = f"{self.resource_path}/instance_policy.json"

        print(
            "\nFor this demo, we'll use the AWS SDK for Python (Boto3) to create
            several AWS resources\n"
```

```
        "to set up a load-balanced web service endpoint and explore some ways
to make it resilient\n"
        "against various kinds of failures.\n\n"
        "Some of the resources create by this demo are:\n"
    )
    print(
        "\t* A DynamoDB table that the web service depends on to provide
book, movie, and song recommendations."
    )
    print(
        "\t* An EC2 launch template that defines EC2 instances that each
contain a Python web server."
    )
    print(
        "\t* An EC2 Auto Scaling group that manages EC2 instances across
several Availability Zones."
    )
    print(
        "\t* An Elastic Load Balancing (ELB) load balancer that targets the
Auto Scaling group to distribute requests."
    )
    print("-" * 88)
    q.ask("Press Enter when you're ready to start deploying resources.")

    print(
        f"Creating and populating a DynamoDB table named
'{self.recommendation.table_name}'."
    )
    self.recommendation.create()
    self.recommendation.populate(recommendations_path)
    print("-" * 88)

    print(
        f"Creating an EC2 launch template that runs '{startup_script}' when
an instance starts.\n"
        f"This script starts a Python web server defined in the `server.py`
script. The web server\n"
        f"listens to HTTP requests on port 80 and responds to requests to '/'
and to '/healthcheck'.\n"
        f"For demo purposes, this server is run as the root user. In
production, the best practice is to\n"
        f"run a web server, such as Apache, with least-privileged
credentials.\n"
    )
)
```

```
print(
    f"The template also defines an IAM policy that each instance uses to
    assume a role that grants\n"
    f"permissions to access the DynamoDB recommendation table and Systems
    Manager parameters\n"
    f"that control the flow of the demo.\n"
)
self.autoscaler.create_template(startup_script, instance_policy)
print("-" * 88)

print(
    f"Creating an EC2 Auto Scaling group that maintains three EC2
    instances, each in a different\n"
    f"Availability Zone."
)
zones = self.autoscaler.create_group(3)
print("-" * 88)
print(
    "At this point, you have EC2 instances created. Once each instance
    starts, it listens for\n"
    "HTTP requests. You can see these instances in the console or
    continue with the demo."
)
print("-" * 88)
q.ask("Press Enter when you're ready to continue.")

print(f"Creating variables that control the flow of the demo.\n")
self.param_helper.reset()

print(
    "\nCreating an Elastic Load Balancing target group and load balancer.
    The target group\n"
    "defines how the load balancer connects to instances. The load
    balancer provides a\n"
    "single endpoint where clients connect and dispatches requests to
    instances in the group.\n"
)
vpc = self.autoscaler.get_default_vpc()
subnets = self.autoscaler.get_subnets(vpc["VpcId"], zones)
target_group = self.loadbalancer.create_target_group(
    self.protocol, self.port, vpc["VpcId"]
)
self.loadbalancer.create_load_balancer(
    [subnet["SubnetId"] for subnet in subnets], target_group
```

```
)
self.autoscaler.attach_load_balancer_target_group(target_group)
print(f"Verifying access to the load balancer endpoint...")
lb_success = self.loadbalancer.verify_load_balancer_endpoint()
if not lb_success:
    print(
        "Couldn't connect to the load balancer, verifying that the port
is open..."
    )
    current_ip_address = requests.get(
        "http://checkip.amazonaws.com"
    ).text.strip()
    sec_group, port_is_open = self.autoscaler.verify_inbound_port(
        vpc, self.port, current_ip_address
    )
    sec_group, ssh_port_is_open = self.autoscaler.verify_inbound_port(
        vpc, self.ssh_port, current_ip_address
    )
    if not port_is_open:
        print(
            "For this example to work, the default security group for
your default VPC must\n"
            "allows access from this computer. You can either add it
automatically from this\n"
            "example or add it yourself using the AWS Management Console.
\n"
        )
        if q.ask(
            f"Do you want to add a rule to security group
{sec_group['GroupId']} to allow\n"
            f"inbound traffic on port {self.port} from your computer's IP
address of {current_ip_address}? (y/n) ",
            q.is_yesno,
        ):
            self.autoscaler.open_inbound_port(
                sec_group["GroupId"], self.port, current_ip_address
            )
    if not ssh_port_is_open:
        if q.ask(
            f"Do you want to add a rule to security group
{sec_group['GroupId']} to allow\n"
            f"inbound SSH traffic on port {self.ssh_port} for debugging
from your computer's IP address of {current_ip_address}? (y/n) ",
            q.is_yesno,
```



```

        ):
            self.autoscaler.open_inbound_port(
                sec_group["GroupId"], self.ssh_port, current_ip_address
            )
            lb_success = self.loadbalancer.verify_load_balancer_endpoint()
        if lb_success:
            print("Your load balancer is ready. You can access it by browsing to:
\n")
            print(f"\thttp://{self.loadbalancer.endpoint()}\n")
        else:
            print(
                "Couldn't get a successful response from the load balancer
endpoint. Troubleshoot by\n"
                "manually verifying that your VPC and security group are
configured correctly and that\n"
                "you can successfully make a GET request to the load balancer
endpoint:\n"
            )
            print(f"\thttp://{self.loadbalancer.endpoint()}\n")
        print("-" * 88)
        q.ask("Press Enter when you're ready to continue with the demo.")

    def demo_choices(self):
        actions = [
            "Send a GET request to the load balancer endpoint.",
            "Check the health of load balancer targets.",
            "Go to the next part of the demo.",
        ]
        choice = 0
        while choice != 2:
            print("-" * 88)
            print(
                "\nSee the current state of the service by selecting one of the
following choices:\n"
            )
            choice = q.choose("\nWhich action would you like to take? ", actions)
            print("-" * 88)
            if choice == 0:
                print("Request:\n")
                print(f"GET http://{self.loadbalancer.endpoint()}")
                response = requests.get(f"http://{self.loadbalancer.endpoint()}")
                print("\nResponse:\n")
                print(f"{response.status_code}")
                if response.headers.get("content-type") == "application/json":

```

```

        pp(response.json())
    elif choice == 1:
        print("\nChecking the health of load balancer targets:\n")
        health = self.loadbalancer.check_target_health()
        for target in health:
            state = target["TargetHealth"]["State"]
            print(
                f"\tTarget {target['Target']['Id']} on port
{target['Target']['Port']} is {state}"
            )
            if state != "healthy":
                print(
                    f"\t\t{target['TargetHealth']['Reason']}:
{target['TargetHealth']['Description']}\n"
                )
            print(
                f"\nNote that it can take a minute or two for the health
check to update\n"
                f"after changes are made.\n"
            )
        elif choice == 2:
            print("\nOkay, let's move on.")
            print("-" * 88)

    def demo(self):
        ssm_only_policy = f"{self.resource_path}/ssm_only_policy.json"

        print("\nResetting parameters to starting values for demo.\n")
        self.param_helper.reset()

        print(
            "\nThis part of the demonstration shows how to toggle different parts
of the system\n"
            "to create situations where the web service fails, and shows how
using a resilient\n"
            "architecture can keep the web service running in spite of these
failures."
        )
        print("-" * 88)

        print(
            "At the start, the load balancer endpoint returns recommendations and
reports that all targets are healthy."
        )

```

```
self.demo_choices()

print(
    f"The web service running on the EC2 instances gets recommendations
    by querying a DynamoDB table.\n"
    f"The table name is contained in a Systems Manager parameter named
    '{self.param_helper.table}'.\n"
    f"To simulate a failure of the recommendation service, let's set this
    parameter to name a non-existent table.\n"
)
self.param_helper.put(self.param_helper.table, "this-is-not-a-table")
print(
    "\nNow, sending a GET request to the load balancer endpoint returns a
    failure code. But, the service reports as\n"
    "healthy to the load balancer because shallow health checks don't
    check for failure of the recommendation service."
)
self.demo_choices()

print(
    f"Instead of failing when the recommendation service fails, the web
    service can return a static response.\n"
    f"While this is not a perfect solution, it presents the customer with
    a somewhat better experience than failure.\n"
)
self.param_helper.put(self.param_helper.failure_response, "static")
print(
    f"\nNow, sending a GET request to the load balancer endpoint returns
    a static response.\n"
    f"The service still reports as healthy because health checks are
    still shallow.\n"
)
self.demo_choices()

print("Let's reinstate the recommendation service.\n")
self.param_helper.put(self.param_helper.table,
self.recommendation.table_name)
print(
    "\nLet's also substitute bad credentials for one of the instances in
    the target group so that it can't\n"
    "access the DynamoDB recommendation table.\n"
)
self.autoscaler.create_instance_profile(
    ssm_only_policy,
```

```
        self.autoscaler.bad_creds_policy_name,
        self.autoscaler.bad_creds_role_name,
        self.autoscaler.bad_creds_profile_name,
        ["AmazonSSMManagedInstanceCore"],
    )
    instances = self.autoscaler.get_instances()
    bad_instance_id = instances[0]
    instance_profile = self.autoscaler.get_instance_profile(bad_instance_id)
    print(
        f"\nReplacing the profile for instance {bad_instance_id} with a
profile that contains\n"
        f"bad credentials...\n"
    )
    self.autoscaler.replace_instance_profile(
        bad_instance_id,
        self.autoscaler.bad_creds_profile_name,
        instance_profile["AssociationId"],
    )
    print(
        "Now, sending a GET request to the load balancer endpoint returns
either a recommendation or a static response,\n"
        "depending on which instance is selected by the load balancer.\n"
    )
    self.demo_choices()

    print(
        "\nLet's implement a deep health check. For this demo, a deep health
check tests whether\n"
        "the web service can access the DynamoDB table that it depends on for
recommendations. Note that\n"
        "the deep health check is only for ELB routing and not for Auto
Scaling instance health.\n"
        "This kind of deep health check is not recommended for Auto Scaling
instance health, because it\n"
        "risks accidental termination of all instances in the Auto Scaling
group when a dependent service fails.\n"
    )
    print(
        "By implementing deep health checks, the load balancer can detect
when one of the instances is failing\n"
        "and take that instance out of rotation.\n"
    )
    self.param_helper.put(self.param_helper.health_check, "deep")
    print(
```

```
        f"\nNow, checking target health indicates that the instance with bad
credentials ({bad_instance_id})\n"
        f"is unhealthy. Note that it might take a minute or two for the load
balancer to detect the unhealthy \n"
        f"instance. Sending a GET request to the load balancer endpoint
always returns a recommendation, because\n"
        "the load balancer takes unhealthy instances out of its rotation.\n"
    )
    self.demo_choices()

    print(
        "\nBecause the instances in this demo are controlled by an auto
scaler, the simplest way to fix an unhealthy\n"
        "instance is to terminate it and let the auto scaler start a new
instance to replace it.\n"
    )
    self.autoscaler.terminate_instance(bad_instance_id)
    print(
        "\nEven while the instance is terminating and the new instance is
starting, sending a GET\n"
        "request to the web service continues to get a successful
recommendation response because\n"
        "the load balancer routes requests to the healthy instances. After
the replacement instance\n"
        "starts and reports as healthy, it is included in the load balancing
rotation.\n"
        "\nNote that terminating and replacing an instance typically takes
several minutes, during which time you\n"
        "can see the changing health check status until the new instance is
running and healthy.\n"
    )
    self.demo_choices()

    print(
        "\nIf the recommendation service fails now, deep health checks mean
all instances report as unhealthy.\n"
    )
    self.param_helper.put(self.param_helper.table, "this-is-not-a-table")
    print(
        "\nWhen all instances are unhealthy, the load balancer continues to
route requests even to\n"
        "unhealthy instances, allowing them to fail open and return a static
response rather than fail\n"
        "closed and report failure to the customer."
```

```
    )
    self.demo_choices()
    self.param_helper.reset()

def destroy(self):
    print(
        "This concludes the demo of how to build and manage a resilient
service.\n"
        "To keep things tidy and to avoid unwanted charges on your account,
we can clean up all AWS resources\n"
        "that were created for this demo."
    )
    if q.ask("Do you want to clean up all demo resources? (y/n) ",
q.is_yesno):
        self.loadbalancer.delete_load_balancer()
        self.loadbalancer.delete_target_group()
        self.autoscaler.delete_group()
        self.autoscaler.delete_key_pair()
        self.autoscaler.delete_template()
        self.autoscaler.delete_instance_profile(
            self.autoscaler.bad_creds_profile_name,
            self.autoscaler.bad_creds_role_name,
        )
        self.recommendation.destroy()
    else:
        print(
            "Okay, we'll leave the resources intact.\n"
            "Don't forget to delete them when you're done with them or you
might incur unexpected charges."
        )

def main():
    parser = argparse.ArgumentParser()
    parser.add_argument(
        "--action",
        required=True,
        choices=["all", "deploy", "demo", "destroy"],
        help="The action to take for the demo. When 'all' is specified, resources
are\n"
        "deployed, the demo is run, and resources are destroyed.",
    )
    parser.add_argument(
        "--resource_path",
```

```
        default="../../../../workflows/resilient_service/resources",
        help="The path to resource files used by this example, such as IAM
policies and\n"
        "instance scripts.",
    )
    args = parser.parse_args()

    print("-" * 88)
    print(
        "Welcome to the demonstration of How to Build and Manage a Resilient
Service!"
    )
    print("-" * 88)

    prefix = "doc-example-resilience"
    recommendation = RecommendationService.from_client(
        "doc-example-recommendation-service"
    )
    autoscaler = AutoScaler.from_client(prefix)
    loadbalancer = LoadBalancer.from_client(prefix)
    param_helper = ParameterHelper.from_client(recommendation.table_name)
    runner = Runner(
        args.resource_path, recommendation, autoscaler, loadbalancer,
param_helper
    )
    actions = [args.action] if args.action != "all" else ["deploy", "demo",
"destroy"]
    for action in actions:
        if action == "deploy":
            runner.deploy()
        elif action == "demo":
            runner.demo()
        elif action == "destroy":
            runner.destroy()

    print("-" * 88)
    print("Thanks for watching!")
    print("-" * 88)

if __name__ == "__main__":
    logging.basicConfig(level=logging.INFO, format="%(levelname)s: %(message)s")
    main()
```

Membuat kelas yang menggabungkan tindakan Penskalaan Otomatis dan Amazon EC2.

```
class AutoScaler:
    """
    Encapsulates Amazon EC2 Auto Scaling and EC2 management actions.
    """

    def __init__(
        self,
        resource_prefix,
        inst_type,
        ami_param,
        autoscaling_client,
        ec2_client,
        ssm_client,
        iam_client,
    ):
        """
        :param resource_prefix: The prefix for naming AWS resources that are
        created by this class.
        :param inst_type: The type of EC2 instance to create, such as t3.micro.
        :param ami_param: The Systems Manager parameter used to look up the AMI
        that is
            created.
        :param autoscaling_client: A Boto3 EC2 Auto Scaling client.
        :param ec2_client: A Boto3 EC2 client.
        :param ssm_client: A Boto3 Systems Manager client.
        :param iam_client: A Boto3 IAM client.
        """
        self.inst_type = inst_type
        self.ami_param = ami_param
        self.autoscaling_client = autoscaling_client
        self.ec2_client = ec2_client
        self.ssm_client = ssm_client
        self.iam_client = iam_client
        self.launch_template_name = f"{resource_prefix}-template"
        self.group_name = f"{resource_prefix}-group"
        self.instance_policy_name = f"{resource_prefix}-pol"
        self.instance_role_name = f"{resource_prefix}-role"
        self.instance_profile_name = f"{resource_prefix}-prof"
        self.bad_creds_policy_name = f"{resource_prefix}-bc-pol"
```



```
self.bad_creds_role_name = f"{resource_prefix}-bc-role"
self.bad_creds_profile_name = f"{resource_prefix}-bc-prof"
self.key_pair_name = f"{resource_prefix}-key-pair"

@classmethod
def from_client(cls, resource_prefix):
    """
    Creates this class from Boto3 clients.

    :param resource_prefix: The prefix for naming AWS resources that are
    created by this class.
    """
    as_client = boto3.client("autoscaling")
    ec2_client = boto3.client("ec2")
    ssm_client = boto3.client("ssm")
    iam_client = boto3.client("iam")
    return cls(
        resource_prefix,
        "t3.micro",
        "/aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x86_64-gp2",
        as_client,
        ec2_client,
        ssm_client,
        iam_client,
    )

def create_instance_profile(
    self, policy_file, policy_name, role_name, profile_name,
    aws_managed_policies=()
):
    """
    Creates a policy, role, and profile that is associated with instances
    created by
    this class. An instance's associated profile defines a role that is
    assumed by the
    instance. The role has attached policies that specify the AWS permissions
    granted to
    clients that run on the instance.

    :param policy_file: The name of a JSON file that contains the policy
    definition to
        create and attach to the role.
    :param policy_name: The name to give the created policy.
```

```

        :param role_name: The name to give the created role.
        :param profile_name: The name to the created profile.
        :param aws_managed_policies: Additional AWS-managed policies that are
attached to
                                the role, such as
AmazonSSMManagedInstanceCore to grant
                                use of Systems Manager to send commands to
the instance.
        :return: The ARN of the profile that is created.
        """
    assume_role_doc = {
        "Version": "2012-10-17",
        "Statement": [
            {
                "Effect": "Allow",
                "Principal": {"Service": "ec2.amazonaws.com"},
                "Action": "sts:AssumeRole",
            }
        ],
    }
    with open(policy_file) as file:
        instance_policy_doc = file.read()

    policy_arn = None
    try:
        pol_response = self.iam_client.create_policy(
            PolicyName=policy_name, PolicyDocument=instance_policy_doc
        )
        policy_arn = pol_response["Policy"]["Arn"]
        log.info("Created policy with ARN %s.", policy_arn)
    except ClientError as err:
        if err.response["Error"]["Code"] == "EntityAlreadyExists":
            log.info("Policy %s already exists, nothing to do.", policy_name)
            list_pol_response = self.iam_client.list_policies(Scope="Local")
            for pol in list_pol_response["Policies"]:
                if pol["PolicyName"] == policy_name:
                    policy_arn = pol["Arn"]
                    break
        if policy_arn is None:
            raise AutoScalerError(f"Couldn't create policy {policy_name}:
{err}")

    try:
        self.iam_client.create_role(

```

```
        RoleName=role_name,
AssumeRolePolicyDocument=json.dumps(assume_role_doc)
    )
    self.iam_client.attach_role_policy(RoleName=role_name,
PolicyArn=policy_arn)
    for aws_policy in aws_managed_policies:
        self.iam_client.attach_role_policy(
            RoleName=role_name,
            PolicyArn=f"arn:aws:iam::aws:policy/{aws_policy}",
        )
    log.info("Created role %s and attached policy %s.", role_name,
policy_arn)
    except ClientError as err:
        if err.response["Error"]["Code"] == "EntityAlreadyExists":
            log.info("Role %s already exists, nothing to do.", role_name)
        else:
            raise AutoScalerError(f"Couldn't create role {role_name}: {err}")

    try:
        profile_response = self.iam_client.create_instance_profile(
            InstanceProfileName=profile_name
        )
        waiter = self.iam_client.get_waiter("instance_profile_exists")
        waiter.wait(InstanceProfileName=profile_name)
        time.sleep(10) # wait a little longer
        profile_arn = profile_response["InstanceProfile"]["Arn"]
        self.iam_client.add_role_to_instance_profile(
            InstanceProfileName=profile_name, RoleName=role_name
        )
        log.info("Created profile %s and added role %s.", profile_name,
role_name)
    except ClientError as err:
        if err.response["Error"]["Code"] == "EntityAlreadyExists":
            prof_response = self.iam_client.get_instance_profile(
                InstanceProfileName=profile_name
            )
            profile_arn = prof_response["InstanceProfile"]["Arn"]
            log.info(
                "Instance profile %s already exists, nothing to do.",
profile_name
            )
        else:
            raise AutoScalerError(
```

```
        f"Couldn't create profile {profile_name} and attach it to
role\n"
        f"{role_name}: {err}"
    )
    return profile_arn

def get_instance_profile(self, instance_id):
    """
    Gets data about the profile associated with an instance.

    :param instance_id: The ID of the instance to look up.
    :return: The profile data.
    """
    try:
        response =
self.ec2_client.describe_iam_instance_profile_associations(
            Filters=[{"Name": "instance-id", "Values": [instance_id]}]
        )
    except ClientError as err:
        raise AutoScalerError(
            f"Couldn't get instance profile association for instance
{instance_id}: {err}"
        )
    else:
        return response["IamInstanceProfileAssociations"][0]

def replace_instance_profile(
    self, instance_id, new_instance_profile_name, profile_association_id
):
    """
    Replaces the profile associated with a running instance. After the
profile is
replaced, the instance is rebooted to ensure that it uses the new
profile. When
the instance is ready, Systems Manager is used to restart the Python web
server.

    :param instance_id: The ID of the instance to update.
    :param new_instance_profile_name: The name of the new profile to
associate with
                                the specified instance.
    """
```

```

        :param profile_association_id: The ID of the existing profile association
for the
                                instance.
"""
try:
    self.ec2_client.replace_iam_instance_profile_association(
        IamInstanceProfile={"Name": new_instance_profile_name},
        AssociationId=profile_association_id,
    )
    log.info(
        "Replaced instance profile for association %s with profile %s.",
        profile_association_id,
        new_instance_profile_name,
    )
    time.sleep(5)
    inst_ready = False
    tries = 0
    while not inst_ready:
        if tries % 6 == 0:
            self.ec2_client.reboot_instances(InstanceIds=[instance_id])
            log.info(
                "Rebooting instance %s and waiting for it to be
ready.",
                instance_id,
            )
            tries += 1
            time.sleep(10)
            response = self.ssm_client.describe_instance_information()
            for info in response["InstanceInformationList"]:
                if info["InstanceId"] == instance_id:
                    inst_ready = True
    self.ssm_client.send_command(
        InstanceIds=[instance_id],
        DocumentName="AWS-RunShellScript",
        Parameters={"commands": ["cd / && sudo python3 server.py 80"]},
    )
    log.info("Restarted the Python web server on instance %s.",
instance_id)
except ClientError as err:
    raise AutoScalerError(
        f"Couldn't replace instance profile for association
{profile_association_id}: {err}"
    )

```

```

def delete_instance_profile(self, profile_name, role_name):
    """
    Detaches a role from an instance profile, detaches policies from the
role,
and deletes all the resources.

:param profile_name: The name of the profile to delete.
:param role_name: The name of the role to delete.
    """
    try:
        self.iam_client.remove_role_from_instance_profile(
            InstanceProfileName=profile_name, RoleName=role_name
        )

self.iam_client.delete_instance_profile(InstanceProfileName=profile_name)
        log.info("Deleted instance profile %s.", profile_name)
        attached_policies = self.iam_client.list_attached_role_policies(
            RoleName=role_name
        )
        for pol in attached_policies["AttachedPolicies"]:
            self.iam_client.detach_role_policy(
                RoleName=role_name, PolicyArn=pol["PolicyArn"]
            )
            if not pol["PolicyArn"].startswith("arn:aws:iam::aws"):
                self.iam_client.delete_policy(PolicyArn=pol["PolicyArn"])
                log.info("Detached and deleted policy %s.", pol["PolicyName"])
            self.iam_client.delete_role(RoleName=role_name)
            log.info("Deleted role %s.", role_name)
    except ClientError as err:
        if err.response["Error"]["Code"] == "NoSuchEntity":
            log.info(
                "Instance profile %s doesn't exist, nothing to do.",
profile_name
            )
        else:
            raise AutoScalerError(
                f"Couldn't delete instance profile {profile_name} or detach "
                f"policies and delete role {role_name}: {err}"
            )

def create_key_pair(self, key_pair_name):
    """

```

```
Creates a new key pair.

:param key_pair_name: The name of the key pair to create.
:return: The newly created key pair.
"""
try:
    response = self.ec2_client.create_key_pair(KeyName=key_pair_name)
    with open(f"{key_pair_name}.pem", "w") as file:
        file.write(response["KeyMaterial"])
    chmod(f"{key_pair_name}.pem", 0o600)
    log.info("Created key pair %s.", key_pair_name)
except ClientError as err:
    raise AutoScalerError(f"Couldn't create key pair {key_pair_name}:
{err}")

def delete_key_pair(self):
    """
    Deletes a key pair.

    :param key_pair_name: The name of the key pair to delete.
    """
    try:
        self.ec2_client.delete_key_pair(KeyName=self.key_pair_name)
        remove(f"{self.key_pair_name}.pem")
        log.info("Deleted key pair %s.", self.key_pair_name)
    except ClientError as err:
        raise AutoScalerError(
            f"Couldn't delete key pair {self.key_pair_name}: {err}"
        )
    except FileNotFoundError:
        log.info("Key pair %s doesn't exist, nothing to do.",
self.key_pair_name)
    except PermissionError:
        log.info(
            "Inadequate permissions to delete key pair %s.",
self.key_pair_name
        )
    except Exception as err:
        raise AutoScalerError(
            f"Couldn't delete key pair {self.key_pair_name}: {err}"
        )
```

```

def create_template(self, server_startup_script_file, instance_policy_file):
    """
    Creates an Amazon EC2 launch template to use with Amazon EC2 Auto
    Scaling. The
    launch template specifies a Bash script in its user data field that runs
    after
    the instance is started. This script installs Python packages and starts
    a
    Python web server on the instance.

    :param server_startup_script_file: The path to a Bash script file that is
    run
                                     when an instance starts.
    :param instance_policy_file: The path to a file that defines a
    permissions policy
                                to create and attach to the instance
    profile.
    :return: Information about the newly created template.
    """
    template = {}
    try:
        self.create_key_pair(self.key_pair_name)
        self.create_instance_profile(
            instance_policy_file,
            self.instance_policy_name,
            self.instance_role_name,
            self.instance_profile_name,
        )
        with open(server_startup_script_file) as file:
            start_server_script = file.read()
        ami_latest = self.ssm_client.get_parameter(Name=self.ami_param)
        ami_id = ami_latest["Parameter"]["Value"]
        lt_response = self.ec2_client.create_launch_template(
            LaunchTemplateName=self.launch_template_name,
            LaunchTemplateData={
                "InstanceType": self.inst_type,
                "ImageId": ami_id,
                "IamInstanceProfile": {"Name": self.instance_profile_name},
                "UserData": base64.b64encode(
                    start_server_script.encode(encoding="utf-8")
                ).decode(encoding="utf-8"),
                "KeyName": self.key_pair_name,
            },
        )

```



```
        template = lt_response["LaunchTemplate"]
        log.info(
            "Created launch template %s for AMI %s on %s.",
            self.launch_template_name,
            ami_id,
            self.inst_type,
        )
    except ClientError as err:
        if (
            err.response["Error"]["Code"]
            == "InvalidLaunchTemplateName.AlreadyExistsException"
        ):
            log.info(
                "Launch template %s already exists, nothing to do.",
                self.launch_template_name,
            )
        else:
            raise AutoScalerError(
                f"Couldn't create launch template
                {self.launch_template_name}: {err}."
            )
        return template

def delete_template(self):
    """
    Deletes a launch template.
    """
    try:
        self.ec2_client.delete_launch_template(
            LaunchTemplateName=self.launch_template_name
        )
        self.delete_instance_profile(
            self.instance_profile_name, self.instance_role_name
        )
        log.info("Launch template %s deleted.", self.launch_template_name)
    except ClientError as err:
        if (
            err.response["Error"]["Code"]
            == "InvalidLaunchTemplateName.NotFoundException"
        ):
            log.info(
                "Launch template %s does not exist, nothing to do.",
                self.launch_template_name,
```

```
        )
    else:
        raise AutoScalerError(
            f"Couldn't delete launch template
{self.launch_template_name}: {err}."
        )

def get_availability_zones(self):
    """
    Gets a list of Availability Zones in the AWS Region of the Amazon EC2
    client.

    :return: The list of Availability Zones for the client Region.
    """
    try:
        response = self.ec2_client.describe_availability_zones()
        zones = [zone["ZoneName"] for zone in response["AvailabilityZones"]]
    except ClientError as err:
        raise AutoScalerError(f"Couldn't get availability zones: {err}.")
    else:
        return zones

def create_group(self, group_size):
    """
    Creates an EC2 Auto Scaling group with the specified size.

    :param group_size: The number of instances to set for the minimum and
    maximum in
        the group.
    :return: The list of Availability Zones specified for the group.
    """
    zones = []
    try:
        zones = self.get_availability_zones()
        self.autoscaling_client.create_auto_scaling_group(
            AutoScalingGroupName=self.group_name,
            AvailabilityZones=zones,
            LaunchTemplate={
                "LaunchTemplateName": self.launch_template_name,
                "Version": "$Default",
            },
            MinSize=group_size,
```

```
        MaxSize=group_size,
    )
    log.info(
        "Created EC2 Auto Scaling group %s with availability zones %s.",
        self.launch_template_name,
        zones,
    )
except ClientError as err:
    if err.response["Error"]["Code"] == "AlreadyExists":
        log.info(
            "EC2 Auto Scaling group %s already exists, nothing to do.",
            self.group_name,
        )
    else:
        raise AutoScalerError(
            f"Couldn't create EC2 Auto Scaling group {self.group_name}:
{err}")
    )
return zones

def get_instances(self):
    """
    Gets data about the instances in the EC2 Auto Scaling group.

    :return: Data about the instances.
    """
    try:
        as_response = self.autoscaling_client.describe_auto_scaling_groups(
            AutoScalingGroupNames=[self.group_name]
        )
        instance_ids = [
            i["InstanceId"]
            for i in as_response["AutoScalingGroups"][0]["Instances"]
        ]
    except ClientError as err:
        raise AutoScalerError(
            f"Couldn't get instances for Auto Scaling group
{self.group_name}: {err}")
    )
    else:
        return instance_ids
```

```

def terminate_instance(self, instance_id):
    """
    Terminates and instances in an EC2 Auto Scaling group. After an instance
    is
    terminated, it can no longer be accessed.

    :param instance_id: The ID of the instance to terminate.
    """
    try:
        self.autoscaling_client.terminate_instance_in_auto_scaling_group(
            InstanceId=instance_id, ShouldDecrementDesiredCapacity=False
        )
        log.info("Terminated instance %s.", instance_id)
    except ClientError as err:
        raise AutoScalerError(f"Couldn't terminate instance {instance_id}:
{err}")

def attach_load_balancer_target_group(self, lb_target_group):
    """
    Attaches an Elastic Load Balancing (ELB) target group to this EC2 Auto
    Scaling group.
    The target group specifies how the load balancer forward requests to the
    instances
    in the group.

    :param lb_target_group: Data about the ELB target group to attach.
    """
    try:
        self.autoscaling_client.attach_load_balancer_target_groups(
            AutoScalingGroupName=self.group_name,
            TargetGroupARNs=[lb_target_group["TargetGroupArn"]],
        )
        log.info(
            "Attached load balancer target group %s to auto scaling group
%s.",
            lb_target_group["TargetGroupName"],
            self.group_name,
        )
    except ClientError as err:
        raise AutoScalerError(
            f"Couldn't attach load balancer target group
{lb_target_group['TargetGroupName']}\n"
            f"to auto scaling group {self.group_name}"
        )

```

```
def _try_terminate_instance(self, inst_id):
    stopping = False
    log.info(f"Stopping {inst_id}.")
    while not stopping:
        try:
            self.autoscaling_client.terminate_instance_in_auto_scaling_group(
                InstanceId=inst_id, ShouldDecrementDesiredCapacity=True
            )
            stopping = True
        except ClientError as err:
            if err.response["Error"]["Code"] == "ScalingActivityInProgress":
                log.info("Scaling activity in progress for %s. Waiting...",
inst_id)
                time.sleep(10)
            else:
                raise AutoScalerError(f"Couldn't stop instance {inst_id}:
{err}.")

def _try_delete_group(self):
    """
    Tries to delete the EC2 Auto Scaling group. If the group is in use or in
progress,
the function waits and retries until the group is successfully deleted.
    """
    stopped = False
    while not stopped:
        try:
            self.autoscaling_client.delete_auto_scaling_group(
                AutoScalingGroupName=self.group_name
            )
            stopped = True
            log.info("Deleted EC2 Auto Scaling group %s.", self.group_name)
        except ClientError as err:
            if (
                err.response["Error"]["Code"] == "ResourceInUse"
                or err.response["Error"]["Code"] ==
"ScalingActivityInProgress"
            ):
                log.info(
                    "Some instances are still running. Waiting for them to
stop..."
                )
```

```
        time.sleep(10)
    else:
        raise AutoScalerError(
            f"Couldn't delete group {self.group_name}: {err}."
        )

def delete_group(self):
    """
    Terminates all instances in the group, deletes the EC2 Auto Scaling
    group.
    """
    try:
        response = self.autoscaling_client.describe_auto_scaling_groups(
            AutoScalingGroupNames=[self.group_name]
        )
        groups = response.get("AutoScalingGroups", [])
        if len(groups) > 0:
            self.autoscaling_client.update_auto_scaling_group(
                AutoScalingGroupName=self.group_name, MinSize=0
            )
            instance_ids = [inst["InstanceId"] for inst in groups[0]
["Instances"]]
            for inst_id in instance_ids:
                self._try_terminate_instance(inst_id)
                self._try_delete_group()
        else:
            log.info("No groups found named %s, nothing to do.",
self.group_name)
    except ClientError as err:
        raise AutoScalerError(f"Couldn't delete group {self.group_name}:
{err}.")

def get_default_vpc(self):
    """
    Gets the default VPC for the account.

    :return: Data about the default VPC.
    """
    try:
        response = self.ec2_client.describe_vpcs(
            Filters=[{"Name": "is-default", "Values": ["true"]}])
    except ClientError as err:
```

```

        raise AutoScalerError(f"Couldn't get default VPC: {err}")
    else:
        return response["Vpcs"][0]

def verify_inbound_port(self, vpc, port, ip_address):
    """
    Verify the default security group of the specified VPC allows ingress
    from this
    computer. This can be done by allowing ingress from this computer's IP
    address. In some situations, such as connecting from a corporate network,
    you
    must instead specify a prefix list ID. You can also temporarily open the
    port to
    any IP address while running this example. If you do, be sure to remove
    public
    access when you're done.

    :param vpc: The VPC used by this example.
    :param port: The port to verify.
    :param ip_address: This computer's IP address.
    :return: The default security group of the specific VPC, and a value that
    indicates
        whether the specified port is open.
    """
    try:
        response = self.ec2_client.describe_security_groups(
            Filters=[
                {"Name": "group-name", "Values": ["default"]},
                {"Name": "vpc-id", "Values": [vpc["VpcId"]]},
            ]
        )
        sec_group = response["SecurityGroups"][0]
        port_is_open = False
        log.info("Found default security group %s.", sec_group["GroupId"])
        for ip_perm in sec_group["IpPermissions"]:
            if ip_perm.get("FromPort", 0) == port:
                log.info("Found inbound rule: %s", ip_perm)
                for ip_range in ip_perm["IpRanges"]:
                    cidr = ip_range.get("CidrIp", "")
                    if cidr.startswith(ip_address) or cidr == "0.0.0.0/0":
                        port_is_open = True
                if ip_perm["PrefixListIds"]:
                    port_is_open = True
    
```

```

        if not port_is_open:
            log.info(
                "The inbound rule does not appear to be open to
either this computer's IP\n"
                "address of %s, to all IP addresses (0.0.0.0/0), or
to a prefix list ID.",
                ip_address,
            )
        else:
            break
    except ClientError as err:
        raise AutoScalerError(
            f"Couldn't verify inbound rule for port {port} for VPC
{vpc['VpcId']}: {err}"
        )
    else:
        return sec_group, port_is_open

def open_inbound_port(self, sec_group_id, port, ip_address):
    """
    Add an ingress rule to the specified security group that allows access on
the
    specified port from the specified IP address.

    :param sec_group_id: The ID of the security group to modify.
    :param port: The port to open.
    :param ip_address: The IP address that is granted access.
    """
    try:
        self.ec2_client.authorize_security_group_ingress(
            GroupId=sec_group_id,
            CidrIp=f"{ip_address}/32",
            FromPort=port,
            ToPort=port,
            IpProtocol="tcp",
        )
        log.info(
            "Authorized ingress to %s on port %s from %s.",
            sec_group_id,
            port,
            ip_address,
        )
    except ClientError as err:

```



```

        raise AutoScalerError(
            f"Couldn't authorize ingress to {sec_group_id} on port {port}
from {ip_address}: {err}"
        )

def get_subnets(self, vpc_id, zones):
    """
    Gets the default subnets in a VPC for a specified list of Availability
    Zones.

    :param vpc_id: The ID of the VPC to look up.
    :param zones: The list of Availability Zones to look up.
    :return: The list of subnets found.
    """
    try:
        response = self.ec2_client.describe_subnets(
            Filters=[
                {"Name": "vpc-id", "Values": [vpc_id]},
                {"Name": "availability-zone", "Values": zones},
                {"Name": "default-for-az", "Values": ["true"]},
            ]
        )
        subnets = response["Subnets"]
        log.info("Found %s subnets for the specified zones.", len(subnets))
    except ClientError as err:
        raise AutoScalerError(f"Couldn't get subnets: {err}")
    else:
        return subnets

```

Membuat kelas yang menggabungkan tindakan Penyeimbangan Beban Elastis.

```

class LoadBalancer:
    """Encapsulates Elastic Load Balancing (ELB) actions."""

    def __init__(self, target_group_name, load_balancer_name, elb_client):
        """
        :param target_group_name: The name of the target group associated with
        the load balancer.

```

```
    :param load_balancer_name: The name of the load balancer.
    :param elb_client: A Boto3 Elastic Load Balancing client.
    """
    self.target_group_name = target_group_name
    self.load_balancer_name = load_balancer_name
    self.elb_client = elb_client
    self._endpoint = None

    @classmethod
    def from_client(cls, resource_prefix):
        """
        Creates this class from a Boto3 client.

        :param resource_prefix: The prefix to give to AWS resources created by
        this class.
        """
        elb_client = boto3.client("elbv2")
        return cls(f"{resource_prefix}-tg", f"{resource_prefix}-lb", elb_client)

    def endpoint(self):
        """
        Gets the HTTP endpoint of the load balancer.

        :return: The endpoint.
        """
        if self._endpoint is None:
            try:
                response = self.elb_client.describe_load_balancers(
                    Names=[self.load_balancer_name]
                )
                self._endpoint = response["LoadBalancers"][0]["DNSName"]
            except ClientError as err:
                raise LoadBalancerError(
                    f"Couldn't get the endpoint for load balancer
                    {self.load_balancer_name}: {err}")
            return self._endpoint

    def create_target_group(self, protocol, port, vpc_id):
        """
        Creates an Elastic Load Balancing target group. The target group
        specifies how
```

the load balancer forward requests to instances in the group and how instance health is checked.

To speed up this demo, the health check is configured with shortened times and lower thresholds. In production, you might want to decrease the sensitivity of your health checks to avoid unwanted failures.

```

:param protocol: The protocol to use to forward requests, such as 'HTTP'.
:param port: The port to use to forward requests, such as 80.
:param vpc_id: The ID of the VPC in which the load balancer exists.
:return: Data about the newly created target group.
"""
try:
    response = self.elb_client.create_target_group(
        Name=self.target_group_name,
        Protocol=protocol,
        Port=port,
        HealthCheckPath="/healthcheck",
        HealthCheckIntervalSeconds=10,
        HealthCheckTimeoutSeconds=5,
        HealthyThresholdCount=2,
        UnhealthyThresholdCount=2,
        VpcId=vpc_id,
    )
    target_group = response["TargetGroups"][0]
    log.info("Created load balancing target group %s.",
self.target_group_name)
except ClientError as err:
    raise LoadBalancerError(
        f"Couldn't create load balancing target group
{self.target_group_name}: {err}")
)
else:
    return target_group

def delete_target_group(self):
    """
    Deletes the target group.
    """
    done = False

```

```

    while not done:
        try:
            response = self.elb_client.describe_target_groups(
                Names=[self.target_group_name]
            )
            tg_arn = response["TargetGroups"][0]["TargetGroupArn"]
            self.elb_client.delete_target_group(TargetGroupArn=tg_arn)
            log.info(
                "Deleted load balancing target group %s.",
                self.target_group_name
            )
            done = True
        except ClientError as err:
            if err.response["Error"]["Code"] == "TargetGroupNotFound":
                log.info(
                    "Load balancer target group %s not found, nothing to
do.",
                    self.target_group_name,
                )
                done = True
            elif err.response["Error"]["Code"] == "ResourceInUse":
                log.info(
                    "Target group not yet released from load balancer,
waiting..."
                )
                time.sleep(10)
            else:
                raise LoadBalancerError(
                    f"Couldn't delete load balancing target group
{self.target_group_name}: {err}"
                )

    def create_load_balancer(self, subnet_ids, target_group):
        """
        Creates an Elastic Load Balancing load balancer that uses the specified
subnets
and forwards requests to the specified target group.

:param subnet_ids: A list of subnets to associate with the load balancer.
:param target_group: An existing target group that is added as a listener
to the
                    load balancer.
:return: Data about the newly created load balancer.

```

```
"""
try:
    response = self.elb_client.create_load_balancer(
        Name=self.load_balancer_name, Subnets=subnet_ids
    )
    load_balancer = response["LoadBalancers"][0]
    log.info("Created load balancer %s.", self.load_balancer_name)
    waiter = self.elb_client.get_waiter("load_balancer_available")
    log.info("Waiting for load balancer to be available...")
    waiter.wait(Names=[self.load_balancer_name])
    log.info("Load balancer is available!")
    self.elb_client.create_listener(
        LoadBalancerArn=load_balancer["LoadBalancerArn"],
        Protocol=target_group["Protocol"],
        Port=target_group["Port"],
        DefaultActions=[
            {
                "Type": "forward",
                "TargetGroupArn": target_group["TargetGroupArn"],
            }
        ],
    )
    log.info(
        "Created listener to forward traffic from load balancer %s to
target group %s.",
        self.load_balancer_name,
        target_group["TargetGroupName"],
    )
except ClientError as err:
    raise LoadBalancerError(
        f"Failed to create load balancer {self.load_balancer_name}"
        f"and add a listener for target group
{target_group['TargetGroupName']}: {err}"
    )
else:
    self._endpoint = load_balancer["DNSName"]
    return load_balancer

def delete_load_balancer(self):
    """
    Deletes a load balancer.
    """
    try:
```

```
        response = self.elb_client.describe_load_balancers(
            Names=[self.load_balancer_name]
        )
        lb_arn = response["LoadBalancers"][0]["LoadBalancerArn"]
        self.elb_client.delete_load_balancer(LoadBalancerArn=lb_arn)
        log.info("Deleted load balancer %s.", self.load_balancer_name)
        waiter = self.elb_client.get_waiter("load_balancers_deleted")
        log.info("Waiting for load balancer to be deleted...")
        waiter.wait(Names=[self.load_balancer_name])
    except ClientError as err:
        if err.response["Error"]["Code"] == "LoadBalancerNotFound":
            log.info(
                "Load balancer %s does not exist, nothing to do.",
                self.load_balancer_name,
            )
        else:
            raise LoadBalancerError(
                f"Couldn't delete load balancer {self.load_balancer_name}:"
                {err}"
            )

    def verify_load_balancer_endpoint(self):
        """
        Verify this computer can successfully send a GET request to the load
        balancer endpoint.
        """
        success = False
        retries = 3
        while not success and retries > 0:
            try:
                lb_response = requests.get(f"http://{self.endpoint()}")
                log.info(
                    "Got response %s from load balancer endpoint.",
                    lb_response.status_code,
                )
                if lb_response.status_code == 200:
                    success = True
            else:
                retries = 0
        except requests.exceptions.ConnectionError:
            log.info(
                "Got connection error from load balancer endpoint,
                retrying..."
            )
```

```

        )
        retries -= 1
        time.sleep(10)
    return success

def check_target_health(self):
    """
    Checks the health of the instances in the target group.

    :return: The health status of the target group.
    """
    try:
        tg_response = self.elb_client.describe_target_groups(
            Names=[self.target_group_name]
        )
        health_response = self.elb_client.describe_target_health(
            TargetGroupArn=tg_response["TargetGroups"][0]["TargetGroupArn"]
        )
    except ClientError as err:
        raise LoadBalancerError(
            f"Couldn't check health of {self.target_group_name} targets:
{err}"
        )
    else:
        return health_response["TargetHealthDescriptions"]

```

Membuat kelas yang menggunakan DynamoDB untuk menyimulasikan layanan yang direkomendasikan.

```

class RecommendationService:
    """
    Encapsulates a DynamoDB table to use as a service that recommends books,
    movies,
    and songs.
    """

    def __init__(self, table_name, dynamodb_client):
        """
        :param table_name: The name of the DynamoDB recommendations table.

```

```

        :param dynamodb_client: A Boto3 DynamoDB client.
        """
        self.table_name = table_name
        self.dynamodb_client = dynamodb_client

    @classmethod
    def from_client(cls, table_name):
        """
        Creates this class from a Boto3 client.

        :param table_name: The name of the DynamoDB recommendations table.
        """
        ddb_client = boto3.client("dynamodb")
        return cls(table_name, ddb_client)

    def create(self):
        """
        Creates a DynamoDB table to use a recommendation service. The table has a
        hash key named 'MediaType' that defines the type of media recommended,
        such as
        Book or Movie, and a range key named 'ItemId' that, combined with the
        MediaType,
        forms a unique identifier for the recommended item.

        :return: Data about the newly created table.
        """
        try:
            response = self.dynamodb_client.create_table(
                TableName=self.table_name,
                AttributeDefinitions=[
                    {"AttributeName": "MediaType", "AttributeType": "S"},
                    {"AttributeName": "ItemId", "AttributeType": "N"},
                ],
                KeySchema=[
                    {"AttributeName": "MediaType", "KeyType": "HASH"},
                    {"AttributeName": "ItemId", "KeyType": "RANGE"},
                ],
                ProvisionedThroughput={"ReadCapacityUnits": 5,
"WriteCapacityUnits": 5},
            )
            log.info("Creating table %s...", self.table_name)
            waiter = self.dynamodb_client.get_waiter("table_exists")
            waiter.wait(TableName=self.table_name)
            log.info("Table %s created.", self.table_name)

```



```
except ClientError as err:
    if err.response["Error"]["Code"] == "ResourceInUseException":
        log.info("Table %s exists, nothing to be do.", self.table_name)
    else:
        raise RecommendationServiceError(
            self.table_name, f"ClientError when creating table: {err}."
        )
else:
    return response

def populate(self, data_file):
    """
    Populates the recommendations table from a JSON file.

    :param data_file: The path to the data file.
    """
    try:
        with open(data_file) as data:
            items = json.load(data)
            batch = [{"PutRequest": {"Item": item}} for item in items]
            self.dynamodb_client.batch_write_item(RequestItems={self.table_name:
batch})
            log.info(
                "Populated table %s with items from %s.", self.table_name,
data_file
            )
    except ClientError as err:
        raise RecommendationServiceError(
            self.table_name, f"Couldn't populate table from {data_file}:
{err}"
        )

def destroy(self):
    """
    Deletes the recommendations table.
    """
    try:
        self.dynamodb_client.delete_table(TableName=self.table_name)
        log.info("Deleting table %s...", self.table_name)
        waiter = self.dynamodb_client.get_waiter("table_not_exists")
        waiter.wait(TableName=self.table_name)
        log.info("Table %s deleted.", self.table_name)
    except ClientError as err:
        if err.response["Error"]["Code"] == "ResourceNotFoundException":
```

```

        log.info("Table %s does not exist, nothing to do.",
self.table_name)
    else:
        raise RecommendationServiceError(
            self.table_name, f"ClientError when deleting table: {err}."
        )

```

Membuat kelas yang menggabungkan tindakan Systems Manager.

```

class ParameterHelper:
    """
    Encapsulates Systems Manager parameters. This example uses these parameters
    to drive
    the demonstration of resilient architecture, such as failure of a dependency
    or
    how the service responds to a health check.
    """

    table = "doc-example-resilient-architecture-table"
    failure_response = "doc-example-resilient-architecture-failure-response"
    health_check = "doc-example-resilient-architecture-health-check"

    def __init__(self, table_name, ssm_client):
        """
        :param table_name: The name of the DynamoDB table that is used as a
        recommendation
                           service.
        :param ssm_client: A Boto3 Systems Manager client.
        """
        self.ssm_client = ssm_client
        self.table_name = table_name

    @classmethod
    def from_client(cls, table_name):
        ssm_client = boto3.client("ssm")
        return cls(table_name, ssm_client)

    def reset(self):
        """
        Resets the Systems Manager parameters to starting values for the demo.

```

```
a
    """
    These are the name of the DynamoDB recommendation table, no response when
    dependency fails, and shallow health checks.
    """
    self.put(self.table, self.table_name)
    self.put(self.failure_response, "none")
    self.put(self.health_check, "shallow")

def put(self, name, value):
    """
    Sets the value of a named Systems Manager parameter.

    :param name: The name of the parameter.
    :param value: The new value of the parameter.
    """
    try:
        self.ssm_client.put_parameter(
            Name=name, Value=value, Overwrite=True, Type="String"
        )
        log.info("Setting demo parameter %s to '%s'.", name, value)
    except ClientError as err:
        raise ParameterHelperError(
            f"Couldn't set parameter {name} to {value}: {err}"
        )
```

- Untuk detail API, lihat topik berikut di Referensi API AWS SDK for Python (Boto3).
 - [AttachLoadBalancerTargetGroups](#)
 - [CreateAutoScalingGroup](#)
 - [CreateInstanceProfile](#)
 - [CreateLaunchTemplate](#)
 - [CreateListener](#)
 - [CreateLoadBalancer](#)
 - [CreateTargetGroup](#)
 - [DeleteAutoScalingGroup](#)
 - [DeleteInstanceProfile](#)
 - [DeleteLaunchTemplate](#)

- [DeleteLoadBalancer](#)
- [DeleteTargetGroup](#)
- [DescribeAutoScalingGroups](#)
- [DescribeAvailabilityZones](#)
- [DescribeIamInstanceProfileAssociations](#)
- [DescribeInstances](#)
- [DescribeLoadBalancers](#)
- [DescribeSubnets](#)
- [DescribeTargetGroups](#)
- [DescribeTargetHealth](#)
- [DescribeVpcs](#)
- [RebootInstances](#)
- [ReplacelamInstanceProfileAssociation](#)
- [TerminateInstanceInAutoScalingGroup](#)
- [UpdateAutoScalingGroup](#)

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Memulai instans Amazon EC2 menggunakan SDK AWS

Contoh-contoh kode berikut ini menunjukkan cara untuk melakukan:

- Membuat pasangan kunci dan grup keamanan.
- Memilih Amazon Machine Image (AMI) dan tipe instans yang kompatibel, lalu membuat instans.
- Menghentikan dan memulai ulang instans.
- Kaitkan alamat IP Elastis dengan instans Anda.
- Menghubungkan instans Anda dengan SSH, lalu membersihkan sumber daya.

.NET

AWS SDK for .NET

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturannya dan menjalankannya di [Repositori Contoh Kode AWS](#).

Menjalankan skenario di prompt perintah.

```
/// <summary>
/// Show Amazon Elastic Compute Cloud (Amazon EC2) Basics actions.
/// </summary>
public class EC2Basics
{
    /// <summary>
    /// Perform the actions defined for the Amazon EC2 Basics scenario.
    /// </summary>
    /// <param name="args">Command line arguments.</param>
    /// <returns>A Task object.</returns>
    static async Task Main(string[] args)
    {
        // Set up dependency injection for Amazon EC2 and Amazon Simple Systems
        // Management Service.
        using var host =
        Microsoft.Extensions.Hosting.Host.CreateDefaultBuilder(args)
            .ConfigureServices((_, services) =>
                services.AddAWSService<IAmazonEC2>()
                    .AddAWSService<IAmazonSimpleSystemsManagement>()
                    .AddTransient<EC2Wrapper>()
                    .AddTransient<SsmWrapper>()
            )
            .Build();

        // Now the client is available for injection.
        var ec2Client = host.Services.GetRequiredService<IAmazonEC2>();
        var ec2Methods = new EC2Wrapper(ec2Client);

        var ssmClient =
        host.Services.GetRequiredService<IAmazonSimpleSystemsManagement>();
        var ssmMethods = new SsmWrapper(ssmClient);
```

```
var uiMethods = new UiMethods();

var uniqueName = Guid.NewGuid().ToString();
var keyPairName = "mvp-example-key-pair" + uniqueName;
var groupName = "ec2-scenario-group" + uniqueName;
var groupDescription = "A security group created for the EC2 Basics
scenario.";

// Start the scenario.
uiMethods.DisplayOverview();
uiMethods.PressEnter();

// Create the key pair.
uiMethods.DisplayTitle("Create RSA key pair");
Console.Write("Let's create an RSA key pair that you can be use to ");
Console.WriteLine("securely connect to your EC2 instance.");
var keyPair = await ec2Methods.CreateKeyPair(keyPairName);

// Save key pair information to a temporary file.
var tempFileName = ec2Methods.SaveKeyPair(keyPair);

Console.WriteLine($"Created the key pair: {keyPair.KeyName} and saved it
to: {tempFileName}");
string? answer;
do
{
    Console.Write("Would you like to list your existing key pairs? ");
    answer = Console.ReadLine();
} while (answer!.ToLower() != "y" && answer.ToLower() != "n");

if (answer == "y")
{
    // List existing key pairs.
    uiMethods.DisplayTitle("Existing key pairs");

    // Passing an empty string to the DescribeKeyPairs method will return
    // a list of all existing key pairs.
    var keyPairs = await ec2Methods.DescribeKeyPairs("");
    keyPairs.ForEach(kp =>
    {
        Console.WriteLine($"{kp.KeyName} created at: {kp.CreateTime}
Fingerprint: {kp.KeyFingerprint}");
    });
}
```

```
    uiMethods.PressEnter();

    // Create the security group.
    Console.WriteLine("Let's create a security group to manage access to your
instance.");
    var secGroupId = await ec2Methods.CreateSecurityGroup(groupName,
groupDescription);
    Console.WriteLine("Let's add rules to allow all HTTP and HTTPS inbound
traffic and to allow SSH only from your current IP address.");

    uiMethods.DisplayTitle("Security group information");
    var secGroups = await ec2Methods.DescribeSecurityGroups(secGroupId);

    Console.WriteLine($"Created security group {groupName} in your default
VPC.");
    secGroups.ForEach(group =>
    {
        ec2Methods.DisplaySecurityGroupInfoAsync(group);
    });
    uiMethods.PressEnter();

    Console.WriteLine("Now we'll authorize the security group we just created
so that it can");
    Console.WriteLine("access the EC2 instances you create.");
    var success = await ec2Methods.AuthorizeSecurityGroupIngress(groupName);

    secGroups = await ec2Methods.DescribeSecurityGroups(secGroupId);
    Console.WriteLine($"Now let's look at the permissions again.");
    secGroups.ForEach(group =>
    {
        ec2Methods.DisplaySecurityGroupInfoAsync(group);
    });
    uiMethods.PressEnter();

    // Get list of available Amazon Linux 2 Amazon Machine Images (AMIs).
    var parameters = await ssmMethods.GetParametersByPath("/aws/service/ami-
amazon-linux-latest");

    List<string> imageIds = parameters.Select(param => param.Value).ToList();

    var images = await ec2Methods.DescribeImages(imageIds);

    var i = 1;
    images.ForEach(image =>
```

```
{
    Console.WriteLine($"{i++}\t{image.Description}");
});

int choice;
bool validNumber = false;

do
{
    Console.Write("Please select an image: ");
    var selImage = Console.ReadLine();
    validNumber = int.TryParse(selImage, out choice);
} while (!validNumber);

var selectedImage = images[choice - 1];

// Display available instance types.
uiMethods.DisplayTitle("Instance Types");
var instanceTypes = await
ec2Methods.DescribeInstanceTypes(selectedImage.Architecture);

i = 1;
instanceTypes.ForEach(instanceType =>
{
    Console.WriteLine($"{i++}\t{instanceType.InstanceType}");
});

do
{
    Console.Write("Please select an instance type: ");
    var selImage = Console.ReadLine();
    validNumber = int.TryParse(selImage, out choice);
} while (!validNumber);

var selectedInstanceType = instanceTypes[choice - 1].InstanceType;

// Create an EC2 instance.
uiMethods.DisplayTitle("Creating an EC2 Instance");
var instanceId = await ec2Methods.RunInstances(selectedImage.ImageId,
selectedInstanceType, keyPairName, secGroupId);
Console.Write("Waiting for the instance to start.");
var isRunning = false;
do
{
```



```
        isRunning = await ec2Methods.WaitForInstanceState(instanceId,
InstanceStateName.Running);
    } while (!isRunning);

    uiMethods.PressEnter();

    var instance = await ec2Methods.DescribeInstance(instanceId);
    uiMethods.DisplayTitle("New Instance Information");
    ec2Methods.DisplayInstanceInformation(instance);

    Console.WriteLine("\nYou can use SSH to connect to your instance. For
example:");
    Console.WriteLine($"\"tssh -i {tempFileName} ec2-
user@{instance.PublicIpAddress}\"");

    uiMethods.PressEnter();

    Console.WriteLine("Now we'll stop the instance and then start it again to
see what's changed.");

    await ec2Methods.StopInstances(instanceId);
    var hasStopped = false;
    do
    {
        hasStopped = await ec2Methods.WaitForInstanceState(instanceId,
InstanceStateName.Stopped);
    } while (!hasStopped);

    Console.WriteLine("\nThe instance has stopped.");

    Console.WriteLine("Now let's start it up again.");
    await ec2Methods.StartInstances(instanceId);
    Console.Write("Waiting for instance to start. ");

    isRunning = false;
    do
    {
        isRunning = await ec2Methods.WaitForInstanceState(instanceId,
InstanceStateName.Running);
    } while (!isRunning);

    Console.WriteLine("\nLet's see what changed.");

    instance = await ec2Methods.DescribeInstance(instanceId);
```

```
    uiMethods.DisplayTitle("New Instance Information");
    ec2Methods.DisplayInstanceInformation(instance);

    Console.WriteLine("\nNotice the change in the SSH information:");
    Console.WriteLine($"\\tssh -i {tempFileName} ec2-
user@{instance.PublicIpAddress}");

    uiMethods.PressEnter();

    Console.WriteLine("Now we will stop the instance again. Then we will
create and associate an");
    Console.WriteLine("Elastic IP address to use with our instance.");

    await ec2Methods.StopInstances(instanceId);
    hasStopped = false;
    do
    {
        hasStopped = await ec2Methods.WaitForInstanceState(instanceId,
InstanceStateName.Stopped);
    } while (!hasStopped);

    Console.WriteLine("\nThe instance has stopped.");
    uiMethods.PressEnter();

    uiMethods.DisplayTitle("Allocate Elastic IP address");
    Console.WriteLine("You can allocate an Elastic IP address and associate
it with your instance\nto keep a consistent IP address even when your instance
restarts.");
    var allocationId = await ec2Methods.AllocateAddress();
    Console.WriteLine("Now we will associate the Elastic IP address with our
instance.");
    var associationId = await ec2Methods.AssociateAddress(allocationId,
instanceId);

    // Start the instance again.
    Console.WriteLine("Now let's start the instance again.");
    await ec2Methods.StartInstances(instanceId);
    Console.WriteLine("Waiting for instance to start. ");

    isRunning = false;
    do
    {
        isRunning = await ec2Methods.WaitForInstanceState(instanceId,
InstanceStateName.Running);
```

```
    } while (!isRunning);

    Console.WriteLine("\nLet's see what changed.");

    instance = await ec2Methods.DescribeInstance(instanceId);
    uiMethods.DisplayTitle("Instance information");
    ec2Methods.DisplayInstanceInformation(instance);

    Console.WriteLine("\nHere is the SSH information:");
    Console.WriteLine($"\"tssh -i {tempFileName} ec2-
user@{instance.PublicIpAddress}");

    Console.WriteLine("Let's stop and start the instance again.");
    uiMethods.PressEnter();

    await ec2Methods.StopInstances(instanceId);

    hasStopped = false;
    do
    {
        hasStopped = await ec2Methods.WaitForInstanceState(instanceId,
InstanceStateName.Stopped);
    } while (!hasStopped);

    Console.WriteLine("\nThe instance has stopped.");

    Console.WriteLine("Now let's start it up again.");
    await ec2Methods.StartInstances(instanceId);
    Console.WriteLine("Waiting for instance to start. ");

    isRunning = false;
    do
    {
        isRunning = await ec2Methods.WaitForInstanceState(instanceId,
InstanceStateName.Running);
    } while (!isRunning);

    instance = await ec2Methods.DescribeInstance(instanceId);
    uiMethods.DisplayTitle("New Instance Information");
    ec2Methods.DisplayInstanceInformation(instance);
    Console.WriteLine("Note that the IP address did not change this time.");
    uiMethods.PressEnter();

    uiMethods.DisplayTitle("Clean up resources");
```

```
Console.WriteLine("Now let's clean up the resources we created.");

// Terminate the instance.
Console.WriteLine("Terminating the instance we created.");
var stateChange = await ec2Methods.TerminateInstances(instanceId);

// Wait for the instance state to be terminated.
var hasTerminated = false;
do
{
    hasTerminated = await ec2Methods.WaitForInstanceState(instanceId,
InstanceStateName.Terminated);
} while (!hasTerminated);

Console.WriteLine($"\\nThe instance {instanceId} has been terminated.");
Console.WriteLine("Now we can disassociate the Elastic IP address and
release it.");

// Disassociate the Elastic IP address.
var disassociated = ec2Methods.DisassociateIp(associationId);

// Delete the Elastic IP address.
var released = ec2Methods.ReleaseAddress(allocationId);

// Delete the security group.
Console.WriteLine($"Deleting the Security Group: {groupName}.");
success = await ec2Methods.DeleteSecurityGroup(secGroupId);
if (success)
{
    Console.WriteLine($"Successfully deleted {groupName}.");
}

// Delete the RSA key pair.
Console.WriteLine($"Deleting the key pair: {keyPairName}");
await ec2Methods.DeleteKeyPair(keyPairName);
Console.WriteLine("Deleting the temporary file with the key
information.");
ec2Methods.DeleteTempFile(tempFileName);
uiMethods.PressEnter();

uiMethods.DisplayTitle("EC2 Basics Scenario completed.");
uiMethods.PressEnter();
}
```

```
}
```

Menentukan kelas yang menggabungkan tindakan EC2.

```
/// <summary>
/// Methods of this class perform Amazon Elastic Compute Cloud (Amazon EC2).
/// </summary>
public class EC2Wrapper
{
    private readonly IAmazonEC2 _amazonEC2;

    public EC2Wrapper(IAmazonEC2 amazonService)
    {
        _amazonEC2 = amazonService;
    }

    /// <summary>
    /// Allocate an Elastic IP address.
    /// </summary>
    /// <returns>The allocation Id of the allocated address.</returns>
    public async Task<string> AllocateAddress()
    {
        var request = new AllocateAddressRequest();

        var response = await _amazonEC2.AllocateAddressAsync(request);
        return response.AllocationId;
    }

    /// <summary>
    /// Associate an Elastic IP address to an EC2 instance.
    /// </summary>
    /// <param name="allocationId">The allocation Id of an Elastic IP address.</
param>
    /// <param name="instanceId">The instance Id of the EC2 instance to
    /// associate the address with.</param>
    /// <returns>The association Id that represents
    /// the association of the Elastic IP address with an instance.</returns>
    public async Task<string> AssociateAddress(string allocationId, string
instanceId)
    {
        var request = new AssociateAddressRequest
        {
```

```

        AllocationId = allocationId,
        InstanceId = instanceId
    };

    var response = await _amazonEC2.AssociateAddressAsync(request);
    return response.AssociationId;
}

/// <summary>
/// Authorize the local computer ingress to EC2 instances associated
/// with the virtual private cloud (VPC) security group.
/// </summary>
/// <param name="groupName">The name of the security group.</param>
/// <returns>A Boolean value indicating the success of the action.</returns>
public async Task<bool> AuthorizeSecurityGroupIngress(string groupName)
{
    // Get the IP address for the local computer.
    var ipAddress = await GetIpAddress();
    Console.WriteLine($"Your IP address is: {ipAddress}");
    var ipRanges = new List<IpRange> { new IpRange { CidrIp =
$"{ipAddress}/32" } };
    var permission = new IpPermission
    {
        Ipv4Ranges = ipRanges,
        IpProtocol = "tcp",
        FromPort = 22,
        ToPort = 22
    };
    var permissions = new List<IpPermission> { permission };
    var response = await _amazonEC2.AuthorizeSecurityGroupIngressAsync(
        new AuthorizeSecurityGroupIngressRequest(groupName, permissions));
    return response.HttpStatusCode == HttpStatusCode.OK;
}

/// <summary>
/// Authorize the local computer for ingress to
/// the Amazon EC2 SecurityGroup.
/// </summary>
/// <returns>The IPv4 address of the computer running the scenario.</returns>
private static async Task<string> GetIpAddress()
{
    var httpClient = new HttpClient();
    var ipString = await httpClient.GetStringAsync("https://
checkip.amazonaws.com");

```

```
    // The IP address is returned with a new line
    // character on the end. Trim off the whitespace and
    // return the value to the caller.
    return ipString.Trim();
}

/// <summary>
/// Create an Amazon EC2 key pair.
/// </summary>
/// <param name="keyPairName">The name for the new key pair.</param>
/// <returns>The Amazon EC2 key pair created.</returns>
public async Task<KeyPair?> CreateKeyPair(string keyPairName)
{
    var request = new CreateKeyPairRequest
    {
        KeyName = keyPairName,
    };

    var response = await _amazonEC2.CreateKeyPairAsync(request);

    if (response.HttpStatusCode == HttpStatusCode.OK)
    {
        var kp = response.KeyPair;
        return kp;
    }
    else
    {
        Console.WriteLine("Could not create key pair.");
        return null;
    }
}

/// <summary>
/// Save KeyPair information to a temporary file.
/// </summary>
/// <param name="keyPair">The name of the key pair.</param>
/// <returns>The full path to the temporary file.</returns>
public string SaveKeyPair(KeyPair keyPair)
{
    var tempPath = Path.GetTempPath();
    var tempFileName = $"{tempPath}\\{Path.GetRandomFileName()}";
    var pemFileName = Path.ChangeExtension(tempFileName, "pem");
}
```

```
// Save the key pair to a file in a temporary folder.
using var stream = new FileStream(pemFileName, FileMode.Create);
using var writer = new StreamWriter(stream);
writer.WriteLine(keyPair.KeyMaterial);

return pemFileName;
}

/// <summary>
/// Create an Amazon EC2 security group.
/// </summary>
/// <param name="groupName">The name for the new security group.</param>
/// <param name="groupDescription">A description of the new security group.</
param>
/// <returns>The group Id of the new security group.</returns>
public async Task<string> CreateSecurityGroup(string groupName, string
groupDescription)
{
    var response = await _amazonEC2.CreateSecurityGroupAsync(
        new CreateSecurityGroupRequest(groupName, groupDescription));

    return response.GroupId;
}

/// <summary>
/// Create a new Amazon EC2 VPC.
/// </summary>
/// <param name="cidrBlock">The CIDR block for the new security group.</
param>
/// <returns>The VPC Id of the new VPC.</returns>
public async Task<string?> CreateVPC(string cidrBlock)
{
    try
    {
        var response = await _amazonEC2.CreateVpcAsync(new CreateVpcRequest
        {
            CidrBlock = cidrBlock,
        });

        Vpc vpc = response.Vpc;
        Console.WriteLine($"Created VPC with ID: {vpc.VpcId}.");
        return vpc.VpcId;
    }
}
```



```
    }
    catch (AmazonEC2Exception ex)
    {
        Console.WriteLine($"Couldn't create VPC because: {ex.Message}");
        return null;
    }
}

/// <summary>
/// Delete an Amazon EC2 key pair.
/// </summary>
/// <param name="keyPairName">The name of the key pair to delete.</param>
/// <returns>A Boolean value indicating the success of the action.</returns>
public async Task<bool> DeleteKeyPair(string keyPairName)
{
    try
    {
        await _amazonEC2.DeleteKeyPairAsync(new
DeleteKeyPairRequest(keyPairName)).ConfigureAwait(false);
        return true;
    }
    catch (Exception ex)
    {
        Console.WriteLine($"Couldn't delete the key pair because:
{ex.Message}");
        return false;
    }
}

/// <summary>
/// Delete the temporary file where the key pair information was saved.
/// </summary>
/// <param name="tempFileName">The path to the temporary file.</param>
public void DeleteTempFile(string tempFileName)
{
    if (File.Exists(tempFileName))
    {
        File.Delete(tempFileName);
    }
}

/// <summary>
/// Delete an Amazon EC2 security group.
/// </summary>
```

```
/// <param name="groupName">The name of the group to delete.</param>
/// <returns>A Boolean value indicating the success of the action.</returns>
public async Task<bool> DeleteSecurityGroup(string groupId)
{
    var response = await _amazonEC2.DeleteSecurityGroupAsync(new
DeleteSecurityGroupRequest { GroupId = groupId });
    return response.HttpStatusCode == HttpStatusCode.OK;
}

/// <summary>
/// Delete an Amazon EC2 VPC.
/// </summary>
/// <returns>A Boolean value indicating the success of the action.</returns>
public async Task<bool> DeleteVpc(string vpcId)
{
    var request = new DeleteVpcRequest
    {
        VpcId = vpcId,
    };

    var response = await _amazonEC2.DeleteVpcAsync(request);

    return response.HttpStatusCode == System.Net.HttpStatusCode.OK;
}

/// <summary>
/// Get information about existing Amazon EC2 images.
/// </summary>
/// <returns>A list of image information.</returns>
public async Task<List<Image>> DescribeImages(List<string>? imageIds)
{
    var request = new DescribeImagesRequest();
    if (imageIds is not null)
    {
        // If the imageIds list is not null, add the list
        // to the request object.
        request.ImageIds = imageIds;
    }

    var response = await _amazonEC2.DescribeImagesAsync(request);
    return response.Images;
}

/// <summary>
```

```
/// Display the information returned by DescribeImages.
/// </summary>
/// <param name="images">The list of image information to display.</param>
public void DisplayImageInfo(List<Image> images)
{
    images.ForEach(image =>
    {
        Console.WriteLine($"{image.Name} Created on: {image.CreationDate}");
    });
}

/// <summary>
/// Get information about an Amazon EC2 instance.
/// </summary>
/// <param name="instanceId">The instance Id of the EC2 instance.</param>
/// <returns>An EC2 instance.</returns>
public async Task<Instance> DescribeInstance(string instanceId)
{
    var response = await _amazonEC2.DescribeInstancesAsync(
        new DescribeInstancesRequest { InstanceIds = new List<string>
{ instanceId } });
    return response.Reservations[0].Instances[0];
}

/// <summary>
/// Display EC2 instance information.
/// </summary>
/// <param name="instance">The instance Id of the EC2 instance.</param>
public void DisplayInstanceInformation(Instance instance)
{
    Console.WriteLine($"ID: {instance.InstanceId}");
    Console.WriteLine($"Image ID: {instance.ImageId}");
    Console.WriteLine($"{instance.InstanceType}");
    Console.WriteLine($"Key Name: {instance.KeyName}");
    Console.WriteLine($"VPC ID: {instance.VpcId}");
    Console.WriteLine($"Public IP: {instance.PublicIpAddress}");
    Console.WriteLine($"State: {instance.State.Name}");
}

/// <summary>
/// Get information about existing EC2 images.
/// </summary>
/// <returns>Async task.</returns>
```

```
public async Task DescribeInstances()
{
    // List all EC2 instances.
    await GetInstanceDescriptions();

    string tagName = "IncludeInList";
    string tagValue = "Yes";
    await GetInstanceDescriptionsFiltered(tagName, tagValue);
}

/// <summary>
/// Get information for all existing Amazon EC2 instances.
/// </summary>
/// <returns>Async task.</returns>
public async Task GetInstanceDescriptions()
{
    Console.WriteLine("Showing all instances:");
    var paginator = _amazonEC2.Paginators.DescribeInstances(new
DescribeInstancesRequest());

    await foreach (var response in paginator.Responses)
    {
        foreach (var reservation in response.Reservations)
        {
            foreach (var instance in reservation.Instances)
            {
                Console.Write($"Instance ID: {instance.InstanceId}");
                Console.WriteLine($"\\tCurrent State: {instance.State.Name}");
            }
        }
    }
}

/// <summary>
/// Get information about EC2 instances filtered by a tag name and value.
/// </summary>
/// <param name="tagName">The name of the tag to filter on.</param>
/// <param name="tagValue">The value of the tag to look for.</param>
/// <returns>Async task.</returns>
public async Task GetInstanceDescriptionsFiltered(string tagName, string
tagValue)
{
    // This tag filters the results of the instance list.
    var filters = new List<Filter>
```

```

    {
        new Filter
        {
            Name = $"tag:{tagName}",
            Values = new List<string>
            {
                tagValue,
            },
        },
    };
    var request = new DescribeInstancesRequest
    {
        Filters = filters,
    };

    Console.WriteLine("\nShowing instances with tag: \"IncludeInList\" set to
    \"Yes\".");
    var paginator = _amazonEC2.Paginators.DescribeInstances(request);

    await foreach (var response in paginator.Responses)
    {
        foreach (var reservation in response.Reservations)
        {
            foreach (var instance in reservation.Instances)
            {
                Console.WriteLine($"Instance ID: {instance.InstanceId} ");
                Console.WriteLine($"Current State: {instance.State.Name}");
            }
        }
    }
}

/// <summary>
/// Describe the instance types available.
/// </summary>
/// <returns>A list of instance type information.</returns>
public async Task<List<InstanceTypeInfo>>
DescribeInstanceTypes(ArchitectureValues architecture)
{
    var request = new DescribeInstanceTypesRequest();

    var filters = new List<Filter>
        { new Filter("processor-info.supported-architecture", new
List<string> { architecture.ToString() }) };

```

```
filters.Add(new Filter("instance-type", new() { "*.micro", "*.small" }));

request.Filters = filters;
var instanceTypes = new List<InstanceTypeInfo>();

var paginator = _amazonEC2.Paginators.DescribeInstanceTypes(request);
await foreach (var instanceType in paginator.InstanceTypes)
{
    instanceTypes.Add(instanceType);
}
return instanceTypes;
}

/// <summary>
/// Display the instance type information returned by
DescribeInstanceTypesAsync.
/// </summary>
/// <param name="instanceTypes">The list of instance type information.</
param>
public void DisplayInstanceTypeInfo(List<InstanceTypeInfo> instanceTypes)
{
    instanceTypes.ForEach(type =>
    {
        Console.WriteLine($"{type.InstanceType}\t{type.MemoryInfo}");
    });
}

/// <summary>
/// Get information about an Amazon EC2 key pair.
/// </summary>
/// <param name="keyPairName">The name of the key pair.</param>
/// <returns>A list of key pair information.</returns>
public async Task<List<KeyPairInfo>> DescribeKeyPairs(string keyPairName)
{
    var request = new DescribeKeyPairsRequest();
    if (!string.IsNullOrEmpty(keyPairName))
    {
        request = new DescribeKeyPairsRequest
        {
            KeyNames = new List<string> { keyPairName }
        };
    }
    var response = await _amazonEC2.DescribeKeyPairsAsync(request);
    return response.KeyPairs.ToList();
}
```

```
}

/// <summary>
/// Retrieve information for an Amazon EC2 security group.
/// </summary>
/// <param name="groupId">The Id of the Amazon EC2 security group.</param>
/// <returns>A list of security group information.</returns>
public async Task<List<SecurityGroup>> DescribeSecurityGroups(string groupId)
{
    var request = new DescribeSecurityGroupsRequest();
    var groupIds = new List<string> { groupId };
    request.GroupIds = groupIds;

    var response = await _amazonEC2.DescribeSecurityGroupsAsync(request);
    return response.SecurityGroups;
}

/// <summary>
/// Display the information returned by the call to
/// DescribeSecurityGroupsAsync.
/// </summary>
/// <param name="securityGroup">A list of security group information.</param>
public void DisplaySecurityGroupInfoAsync(SecurityGroup securityGroup)
{
    Console.WriteLine($"{securityGroup.GroupName}");
    Console.WriteLine("Ingress permissions:");
    securityGroup.IpPermissions.ForEach(permission =>
    {
        Console.WriteLine($"  \tFromPort: {permission.FromPort}");
        Console.WriteLine($"  \tIpProtocol: {permission.IpProtocol}");

        Console.WriteLine($"  \tIpv4Ranges: ");
        permission.Ipv4Ranges.ForEach(range =>
        { Console.WriteLine($"  \t{range.CidrIp} "); });

        Console.WriteLine($"  \n\tIpv6Ranges:");
        permission.Ipv6Ranges.ForEach(range =>
        { Console.WriteLine($"  \t{range.CidrIpv6} "); });

        Console.WriteLine($"  \n\tPrefixListIds: ");
        permission.PrefixListIds.ForEach(id => Console.WriteLine($"  \t{id.Id} "));

        Console.WriteLine($"  \n\tTo Port: {permission.ToPort}");
    });
}
```

```
});
Console.WriteLine("Egress permissions:");
securityGroup.IpPermissionsEgress.ForEach(permission =>
{
    Console.WriteLine($"\\tFromPort: {permission.FromPort}");
    Console.WriteLine($"\\tIpProtocol: {permission.IpProtocol}");

    Console.WriteLine($"\\tIpv4Ranges: ");
    permission.Ipv4Ranges.ForEach(range =>
{ Console.WriteLine($"{range.CidrIp} "); });

    Console.WriteLine($"\\n\\tIpv6Ranges:");
    permission.Ipv6Ranges.ForEach(range =>
{ Console.WriteLine($"{range.CidrIpv6} "); });

    Console.WriteLine($"\\n\\tPrefixListIds: ");
    permission.PrefixListIds.ForEach(id => Console.WriteLine($"{id.Id} "));

    Console.WriteLine($"\\n\\tTo Port: {permission.ToPort}");
});
}

/// <summary>
/// Disassociate an Elastic IP address from an EC2 instance.
/// </summary>
/// <param name="associationId">The association Id.</param>
/// <returns>A Boolean value indicating the success of the action.</returns>
public async Task<bool> DisassociateIp(string associationId)
{
    var response = await _amazonEC2.DisassociateAddressAsync(
        new DisassociateAddressRequest { AssociationId = associationId });
    return response.HttpStatusCode == HttpStatusCode.OK;
}

/// <summary>
/// Retrieve a list of available Amazon Linux images.
/// </summary>
/// <returns>A list of image information.</returns>
public async Task<List<Image>> GetEC2AmiList()
{
    var filter = new Filter { Name = "architecture", Values = new
List<string> { "x86_64" } };
    var filters = new List<Filter> { filter };
}
```



```
        var response = await _amazonEC2.DescribeImagesAsync(new
DescribeImagesRequest { Filters = filters });
        return response.Images;
    }

    /// <summary>
    /// Reboot EC2 instances.
    /// </summary>
    /// <param name="ec2InstanceId">The instance Id of the instances that will be
rebooted.</param>
    /// <returns>Async task.</returns>
    public async Task RebootInstances(string ec2InstanceId)
    {
        var request = new RebootInstancesRequest
        {
            InstanceIds = new List<string> { ec2InstanceId },
        };

        var response = await _amazonEC2.RebootInstancesAsync(request);
        if (response.HttpStatusCode == System.Net.HttpStatusCode.OK)
        {
            Console.WriteLine("Instances successfully rebooted.");
        }
        else
        {
            Console.WriteLine("Could not reboot one or more instances.");
        }
    }

    /// <summary>
    /// Release an Elastic IP address.
    /// </summary>
    /// <param name="allocationId">The allocation Id of the Elastic IP address.</
param>
    /// <returns>A Boolean value indicating the success of the action.</returns>
    public async Task<bool> ReleaseAddress(string allocationId)
    {
        var request = new ReleaseAddressRequest
        {
            AllocationId = allocationId
        };

        var response = await _amazonEC2.ReleaseAddressAsync(request);
        return response.HttpStatusCode == HttpStatusCode.OK;
    }
}
```

```
}

/// <summary>
/// Create and run an EC2 instance.
/// </summary>
/// <param name="ImageId">The image Id of the image used as a basis for the
/// EC2 instance.</param>
/// <param name="instanceType">The instance type of the EC2 instance to
create.</param>
/// <param name="keyName">The name of the key pair to associate with the
/// instance.</param>
/// <param name="groupId">The Id of the Amazon EC2 security group that will
be
/// allowed to interact with the new EC2 instance.</param>
/// <returns>The instance Id of the new EC2 instance.</returns>
public async Task<string> RunInstances(string imageId, string instanceType,
string keyName, string groupId)
{
    var request = new RunInstancesRequest
    {
        ImageId = imageId,
        InstanceType = instanceType,
        KeyName = keyName,
        MinCount = 1,
        MaxCount = 1,
        SecurityGroupIds = new List<string> { groupId }
    };
    var response = await _amazonEC2.RunInstancesAsync(request);
    return response.Reservation.Instances[0].InstanceId;
}

/// <summary>
/// Start an EC2 instance.
/// </summary>
/// <param name="ec2InstanceId">The instance Id of the Amazon EC2 instance
/// to start.</param>
/// <returns>Async task.</returns>
public async Task StartInstances(string ec2InstanceId)
{
    var request = new StartInstancesRequest
    {
        InstanceIds = new List<string> { ec2InstanceId },
    };
};
```

```
var response = await _amazonEC2.StartInstancesAsync(request);

if (response.StartingInstances.Count > 0)
{
    var instances = response.StartingInstances;
    instances.ForEach(i =>
    {
        Console.WriteLine($"Successfully started the EC2 instance with
instance ID: {i.InstanceId}.");
    });
}

/// <summary>
/// Stop an EC2 instance.
/// </summary>
/// <param name="ec2InstanceId">The instance Id of the EC2 instance to
/// stop.</param>
/// <returns>Async task.</returns>
public async Task StopInstances(string ec2InstanceId)
{
    // In addition to the list of instance Ids, the
    // request can also include the following properties:
    //     Force      When true, forces the instances to
    //                 stop but you must check the integrity
    //                 of the file system. Not recommended on
    //                 Windows instances.
    //     Hibernate  When true, hibernates the instance if the
    //                 instance was enabled for hibernation when
    //                 it was launched.
    var request = new StopInstancesRequest
    {
        InstanceIds = new List<string> { ec2InstanceId },
    };

    var response = await _amazonEC2.StopInstancesAsync(request);

    if (response.StoppingInstances.Count > 0)
    {
        var instances = response.StoppingInstances;
        instances.ForEach(i =>
        {
            Console.WriteLine($"Successfully stopped the EC2 Instance " +
```

```
        $"with InstanceID: {i.InstanceId}.");
    });
}
}

/// <summary>
/// Terminate an EC2 instance.
/// </summary>
/// <param name="ec2InstanceId">The instance Id of the EC2 instance
/// to terminate.</param>
/// <returns>Async task.</returns>
public async Task<List<InstanceStateChange>> TerminateInstances(string
ec2InstanceId)
{
    var request = new TerminateInstancesRequest
    {
        InstanceIds = new List<string> { ec2InstanceId }
    };

    var response = await _amazonEC2.TerminateInstancesAsync(request);
    return response.TerminatingInstances;
}

/// <summary>
/// Wait until an EC2 instance is in a specified state.
/// </summary>
/// <param name="instanceId">The instance Id.</param>
/// <param name="stateName">The state to wait for.</param>
/// <returns>A Boolean value indicating the success of the action.</returns>
public async Task<bool> WaitForInstanceState(string instanceId,
InstanceStateName stateName)
{
    var request = new DescribeInstancesRequest
    {
        InstanceIds = new List<string> { instanceId }
    };

    // Wait until the instance is running.
    var hasState = false;
    do
    {
        // Wait 5 seconds.
        Thread.Sleep(5000);
```

```
        // Check for the desired state.
        var response = await _amazonEC2.DescribeInstancesAsync(request);
        var instance = response.Reservations[0].Instances[0];
        hasState = instance.State.Name == stateName;
        Console.WriteLine(". ");
    } while (!hasState);

    return hasState;
}
}
```

- Untuk detail API, lihat topik berikut di Referensi API AWS SDK for .NET .
 - [AllocateAddress](#)
 - [AssociateAddress](#)
 - [AuthorizeSecurityGroupIngress](#)
 - [CreateKeyPair](#)
 - [CreateSecurityGroup](#)
 - [DeleteKeyPair](#)
 - [DeleteSecurityGroup](#)
 - [DescribeImages](#)
 - [DescribeInstanceTypes](#)
 - [DescribeInstances](#)
 - [DescribeKeyPairs](#)
 - [DescribeSecurityGroups](#)
 - [DisassociateAddress](#)
 - [ReleaseAddress](#)
 - [RunInstances](#)
 - [StartInstances](#)
 - [StopInstances](#)
 - [TerminateInstances](#)
 - [UnmonitorInstances](#)

Bash

AWS CLI dengan skrip Bash

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankan di [Repositori Contoh AWS Kode](#).

Jalankan skenario interaktif di penggugah/prompt perintah.

```
#####
# function get_started_with_ec2_instances
#
# Runs an interactive scenario that shows how to get started using EC2 instances.
#
# "EC2 access" permissions are needed to run this code.
#
# Returns:
# 0 - If successful.
# 1 - If an error occurred.
#####
function get_started_with_ec2_instances() {
    # Requires version 4 for mapfile.
    local required_version=4.0

    # Get the current Bash version
    # Check if BASH_VERSION is set
    local current_version
    if [[ -n "$BASH_VERSION" ]]; then
        # Convert BASH_VERSION to a number for comparison
        current_version=$BASH_VERSION
    else
        # Get the current Bash version using the bash command
        current_version=$(bash --version | head -n 1 | awk '{ print $4 }')
    fi

    # Convert version strings to numbers for comparison
    local required_version_num current_version_num
    required_version_num=$(echo "$required_version" | awk -F. '{ print ($1 * 10000)
+ ($2 * 100) + $3 }')
```

```
current_version_num=$(echo "$current_version" | awk -F. '{ print ($1 * 10000) + ($2 * 100) + $3 }')

# Compare versions
if ((current_version_num < required_version_num)); then
    echo "Error: This script requires Bash version $required_version or higher."
    echo "Your current Bash version is number is $current_version."
    exit 1
fi

{
    if [ "$EC2_OPERATIONS_SOURCED" != "True" ]; then

        source ./ec2_operations.sh
    fi
}

echo_repeat "*" 88
echo "Welcome to the Amazon Elastic Compute Cloud (Amazon EC2) get started with instances demo."
echo_repeat "*" 88
echo

echo "Let's create an RSA key pair that you can be use to securely connect to "
echo "your EC2 instance."

echo -n "Enter a unique name for your key: "
get_input
local key_name
key_name=$get_input_result

local temp_dir
temp_dir=$(mktemp -d)
local key_file_name="$temp_dir/${key_name}.pem"

if ec2_create_keypair -n "${key_name}" -f "${key_file_name}"; then
    echo "Created a key pair $key_name and saved the private key to $key_file_name"
    echo
else
    errecho "The key pair failed to create. This demo will exit."
    return 1
fi
```

```
chmod 400 "${key_file_name}"

if yes_no_input "Do you want to list some of your key pairs? (y/n) "; then
  local keys_and_fingerprints
  keys_and_fingerprints="$(ec2_describe_key_pairs)" && {
    local image_name_and_id
    while IFS=$'\n' read -r image_name_and_id; do
      local entries
      IFS=$'\t' read -ra entries <<<"$image_name_and_id"
      echo "Found rsa key ${entries[0]} with fingerprint:"
      echo "    ${entries[1]}"
    done <<<"$keys_and_fingerprints"
  }
fi

echo_repeat "*" 88
echo_repeat "*" 88

echo "Let's create a security group to manage access to your instance."
echo -n "Enter a unique name for your security group: "
get_input
local security_group_name
security_group_name=$get_input_result
local security_group_id
security_group_id=$(ec2_create_security_group -n "$security_group_name" \
  -d "Security group for EC2 instance") || {
  errecho "The security failed to create. This demo will exit."
  clean_up "$key_name" "$key_file_name"
  return 1
}

echo "Security group created with ID $security_group_id"
echo

local public_ip
public_ip=$(curl -s http://checkip.amazonaws.com)

echo "Let's add a rule to allow SSH only from your current IP address."
echo "Your public IP address is $public_ip"
echo -n "press return to add this rule to your security group."
get_input
```



```

if ! ec2_authorize_security_group_ingress -g "$security_group_id" -i
"$public_ip" -p tcp -f 22 -t 22; then
    errecho "The security group rules failed to update. This demo will exit."
    clean_up "$key_name" "$key_file_name" "$security_group_id"
    return 1
fi

echo "Security group rules updated"

local security_group_description
security_group_description="$(ec2_describe_security_groups -g
"${security_group_id}")" || {
    errecho "Failed to describe security groups. This demo will exit."
    clean_up "$key_name" "$key_file_name" "$security_group_id"
    return 1
}

mapfile -t parameters <<<"$security_group_description"
IFS=$'\t' read -ra entries <<<"${parameters[0]}"
echo "Security group: ${entries[0]}"
echo "    ID: ${entries[1]}"
echo "    VPC: ${entries[2]}"
echo "Inbound permissions:"
IFS=$'\t' read -ra entries <<<"${parameters[1]}"
echo "    IpProtocol: ${entries[0]}"
echo "    FromPort: ${entries[1]}"
echo "    ToPort: ${entries[2]}"
echo "    CidrIp: ${parameters[2]}"

local parameters
parameters="$(ssm_get_parameters_by_path -p "/aws/service/ami-amazon-linux-
latest")" || {
    errecho "Failed to get parameters. This demo will exit."
    clean_up "$key_name" "$key_file_name" "$security_group_id"
    return 1
}

local image_ids=""
mapfile -t parameters <<<"$parameters"
for image_name_and_id in "${parameters[@]"; do
    IFS=$'\t' read -ra values <<<"$image_name_and_id"
    if [[ "${values[0]}" == *"amzn2"* ]]; then
        image_ids+="${values[1]} "

```

```

    fi
done

local images
images="$(ec2_describe_images -i "$image_ids")" || {
    errecho "Failed to describe images. This demo will exit."
    clean_up "$key_name" "$key_file_name" "$security_group_id"
    return 1
}

new_line_and_tab_to_list "$images"
local images=("${list_result[@]}")

# Get the size of the array
local images_count=${#images[@]}

if ((images_count == 0)); then
    errecho "No images found. This demo will exit."
    clean_up "$key_name" "$key_file_name" "$security_group_id"
    return 1
fi

echo_repeat "*" 88
echo_repeat "*" 88

echo "Let's create an instance from an Amazon Linux 2 AMI. Here are some
options:"
for ((i = 0; i < images_count; i += 3)); do
    echo "$(((i / 3) + 1)) - ${images[$i]}"
done

integer_input "Please enter the number of the AMI you want to use: " 1
"$((images_count / 3))"
local choice=$get_input_result
choice=$((choice - 1) * 3)

echo "Great choice."
echo

local architecture=${images[$((choice + 1))]}
local image_id=${images[$((choice + 2))]}
echo "Here are some instance types that support the ${architecture}
architecture of the image:"

```

```

response="$(ec2_describe_instance_types -a "${architecture}" -t
"*.*micro,*.*small")" || {
    errecho "Failed to describe instance types. This demo will exit."
    clean_up "$key_name" "$key_file_name" "$security_group_id"
    return 1
}

local instance_types
mapfile -t instance_types <<<"$response"

# Get the size of the array
local instance_types_count=${#instance_types[@]}

echo "Here are some options:"
for ((i = 0; i < instance_types_count; i++)); do
    echo "$((i + 1)) - ${instance_types[$i]}"
done

integer_input "Which one do you want to use? " 1 "${#instance_types[@]}"
"
choice=$get_input_result
local instance_type=${instance_types[$((choice - 1))]}
echo "Another great choice."
echo

echo "Creating your instance and waiting for it to start..."
local instance_id
instance_id=$(ec2_run_instances -i "$image_id" -t "$instance_type" -k
"$key_name" -s "$security_group_id") || {
    errecho "Failed to run instance. This demo will exit."
    clean_up "$key_name" "$key_file_name" "$security_group_id"
    return 1
}

ec2_wait_for_instance_running -i "$instance_id"
echo "Your instance is ready:"
echo

local instance_details
instance_details="$(ec2_describe_instances -i "${instance_id}")"

echo
print_instance_details "${instance_details}"

```

```
local public_ip
public_ip=$(echo "${instance_details}" | awk '{print $6}')
echo
echo "You can use SSH to connect to your instance"
echo "If the connection attempt times out, you might have to manually update
the SSH ingress rule"
echo "for your IP address in the AWS Management Console."
connect_to_instance "$key_file_name" "$public_ip"

echo -n "Press Enter when you're ready to continue the demo: "
get_input

echo_repeat "*" 88
echo_repeat "*" 88

echo "Let's stop and start your instance to see what changes."
echo "Stopping your instance and waiting until it's stopped..."
ec2_stop_instances -i "$instance_id"
ec2_wait_for_instance_stopped -i "$instance_id"

echo "Your instance is stopped. Restarting..."

ec2_start_instances -i "$instance_id"
ec2_wait_for_instance_running -i "$instance_id"

echo "Your instance is running again."
local instance_details
instance_details="$(ec2_describe_instances -i "${instance_id}")"

print_instance_details "${instance_details}"

public_ip=$(echo "${instance_details}" | awk '{print $6}')

echo "Every time your instance is restarted, its public IP address changes"
connect_to_instance "$key_file_name" "$public_ip"

echo -n "Press Enter when you're ready to continue the demo: "
get_input

echo_repeat "*" 88
echo_repeat "*" 88

echo "You can allocate an Elastic IP address and associate it with your
instance"
```

```
echo "to keep a consistent IP address even when your instance restarts."

local result
result=$(ec2_allocate_address -d vpc) || {
    errecho "Failed to allocate an address. This demo will exit."
    clean_up "$key_name" "$key_file_name" "$security_group_id" "$instance_id"
    return 1
}

local elastic_ip allocation_id
elastic_ip=$(echo "$result" | awk '{print $1}')
allocation_id=$(echo "$result" | awk '{print $2}')

echo "Allocated static Elastic IP address: $elastic_ip"

local association_id
association_id=$(ec2_associate_address -i "$instance_id" -a "$allocation_id")
|| {
    errecho "Failed to associate an address. This demo will exit."
    clean_up "$key_name" "$key_file_name" "$security_group_id" "$instance_id"
    "$allocation_id"
    return 1
}

echo "Associated your Elastic IP with your instance."
echo "You can now use SSH to connect to your instance by using the Elastic IP."
connect_to_instance "$key_file_name" "$elastic_ip"

echo -n "Press Enter when you're ready to continue the demo: "
get_input

echo_repeat "*" 88
echo_repeat "*" 88

echo "Let's stop and start your instance to see what changes."
echo "Stopping your instance and waiting until it's stopped..."
ec2_stop_instances -i "$instance_id"
ec2_wait_for_instance_stopped -i "$instance_id"

echo "Your instance is stopped. Restarting..."

ec2_start_instances -i "$instance_id"
ec2_wait_for_instance_running -i "$instance_id"
```

```

echo "Your instance is running again."
local instance_details
instance_details="$(ec2_describe_instances -i "${instance_id}")"

print_instance_details "${instance_details}"

echo "Because you have associated an Elastic IP with your instance, you can"
echo "connect by using a consistent IP address after the instance restarts."
connect_to_instance "$key_file_name" "$elastic_ip"

echo -n "Press Enter when you're ready to continue the demo: "
get_input

echo_repeat "*" 88
echo_repeat "*" 88

if yes_no_input "Do you want to delete the resources created in this demo: (y/
n) "; then
    clean_up "$key_name" "$key_file_name" "$security_group_id" "$instance_id" \
        "$allocation_id" "$association_id"
else
    echo "The following resources were not deleted."
    echo "Key pair: $key_name"
    echo "Key file: $key_file_name"
    echo "Security group: $security_group_id"
    echo "Instance: $instance_id"
    echo "Elastic IP address: $elastic_ip"
fi
}

#####
# function clean_up
#
# This function cleans up the created resources.
# $1 - The name of the ec2 key pair to delete.
# $2 - The name of the key file to delete.
# $3 - The ID of the security group to delete.
# $4 - The ID of the instance to terminate.
# $5 - The ID of the elastic IP address to release.
# $6 - The ID of the elastic IP address to disassociate.
#
# Returns:
# 0 - If successful.
# 1 - If an error occurred.

```

```
#####  
function clean_up() {  
    local result=0  
    local key_pair_name=$1  
    local key_file_name=$2  
    local security_group_id=$3  
    local instance_id=$4  
    local allocation_id=$5  
    local association_id=$6  
  
    if [ -n "$association_id" ]; then  
        # bashsupport disable=BP2002  
        if (ec2_disassociate_address -a "$association_id"); then  
            echo "Disassociated elastic IP address with ID $association_id"  
        else  
            errecho "The elastic IP address disassociation failed."  
            result=1  
        fi  
    fi  
  
    if [ -n "$allocation_id" ]; then  
        # bashsupport disable=BP2002  
        if (ec2_release_address -a "$allocation_id"); then  
            echo "Released elastic IP address with ID $allocation_id"  
        else  
            errecho "The elastic IP address release failed."  
            result=1  
        fi  
    fi  
  
    if [ -n "$instance_id" ]; then  
        # bashsupport disable=BP2002  
        if (ec2_terminate_instances -i "$instance_id"); then  
            echo "Started terminating instance with ID $instance_id"  
  
            ec2_wait_for_instance_terminated -i "$instance_id"  
        else  
            errecho "The instance terminate failed."  
            result=1  
        fi  
    fi  
  
    if [ -n "$security_group_id" ]; then  
        # bashsupport disable=BP2002
```

```

    if (ec2_delete_security_group -i "$security_group_id"); then
        echo "Deleted security group with ID $security_group_id"
    else
        errecho "The security group delete failed."
        result=1
    fi
fi

if [ -n "$key_pair_name" ]; then
    # bashsupport disable=BP2002
    if (ec2_delete_keypair -n "$key_pair_name"); then
        echo "Deleted key pair named $key_pair_name"
    else
        errecho "The key pair delete failed."
        result=1
    fi
fi

if [ -n "$key_file_name" ]; then
    rm -f "$key_file_name"
fi

return $result
}

#####
# function ssm_get_parameters_by_path
#
# This function retrieves one or more parameters from the AWS Systems Manager
# Parameter Store
# by specifying a parameter path.
#
# Parameters:
#     -p parameter_path - The path of the parameter(s) to retrieve.
#
# And:
#     0 - If successful.
#     1 - If it fails.
#####
function ssm_get_parameters_by_path() {
    local parameter_path response
    local option OPTARG # Required to use getopt command in a function.

    # bashsupport disable=BP5008

```



```
function usage() {
    echo "function ssm_get_parameters_by_path"
    echo "Retrieves one or more parameters from the AWS Systems Manager Parameter
Store by specifying a parameter path."
    echo "  -p parameter_path - The path of the parameter(s) to retrieve."
    echo ""
}

# Retrieve the calling parameters.
while getopts "p:h" option; do
    case "${option}" in
        p) parameter_path="${OPTARG}" ;;
        h)
            usage
            return 0
            ;;
        \?)
            echo "Invalid parameter"
            usage
            return 1
            ;;
    esac
done
export OPTIND=1

if [[ -z "$parameter_path" ]]; then
    errecho "ERROR: You must provide a parameter path with the -p parameter."
    usage
    return 1
fi

response=$(aws ssm get-parameters-by-path \
    --path "$parameter_path" \
    --query "Parameters[*].[Name, Value]" \
    --output text) || {
    aws_cli_error_log $?
    errecho "ERROR: AWS reports get-parameters-by-path operation failed.
$response"
    return 1
}

echo "$response"

return 0
```

```
}

#####
# function print_instance_details
#
# This function prints the details of an Amazon Elastic Compute Cloud (Amazon
# EC2) instance.
#
# Parameters:
#     instance_details - The instance details in the format "InstanceId ImageId
#     InstanceType KeyName VpcId PublicIpAddress State.Name".
#
# Returns:
#     0 - If successful.
#     1 - If it fails.
#####
function print_instance_details() {
    local instance_details="$1"

    if [[ -z "${instance_details}" ]]; then
        echo "Error: Missing required instance details argument."
        return 1
    fi

    local instance_id image_id instance_type key_name vpc_id public_ip state
    instance_id=$(echo "${instance_details}" | awk '{print $1}')
    image_id=$(echo "${instance_details}" | awk '{print $2}')
    instance_type=$(echo "${instance_details}" | awk '{print $3}')
    key_name=$(echo "${instance_details}" | awk '{print $4}')
    vpc_id=$(echo "${instance_details}" | awk '{print $5}')
    public_ip=$(echo "${instance_details}" | awk '{print $6}')
    state=$(echo "${instance_details}" | awk '{print $7}')

    echo "    ID: ${instance_id}"
    echo "    Image ID: ${image_id}"
    echo "    Instance type: ${instance_type}"
    echo "    Key name: ${key_name}"
    echo "    VPC ID: ${vpc_id}"
    echo "    Public IP: ${public_ip}"
    echo "    State: ${state}"

    return 0
}
```

```
#####
# function connect_to_instance
#
# This function displays the public IP address of an Amazon Elastic Compute Cloud
  (Amazon EC2) instance and prompts the user to connect to the instance via SSH.
#
# Parameters:
#     $1 - The name of the key file used to connect to the instance.
#     $2 - The public IP address of the instance.
#
# Returns:
#     None
#####
function connect_to_instance() {
    local key_file_name="$1"
    local public_ip="$2"

    # Validate the input parameters
    if [[ -z "$key_file_name" ]]; then
        echo "ERROR: You must provide a key file name as the first argument." >&2
        return 1
    fi

    if [[ -z "$public_ip" ]]; then
        echo "ERROR: You must provide a public IP address as the second argument."
    >&2
        return 1
    fi

    # Display the public IP address and connection command
    echo "To connect, run the following command:"
    echo "    ssh -i ${key_file_name} ec2-user@${public_ip}"

    # Prompt the user to connect to the instance
    if yes_no_input "Do you want to connect now? (y/n) "; then
        echo "After you have connected, you can return to this example by typing
'exit'"
        ssh -i "${key_file_name}" ec2-user@"${public_ip}"
    fi
}

#####
# function get_input
#
```

```

# This function gets user input from the command line.
#
# Outputs:
#   User input to stdout.
#
# Returns:
#   0
#####
function get_input() {

    if [ -z "${mock_input+x}" ]; then
        read -r get_input_result
    else

        if [ "$mock_input_array_index" -lt ${#mock_input_array[@]} ]; then
            get_input_result="${mock_input_array[$mock_input_array_index]}"
            # bashsupport disable=BP2001
            # shellcheck disable=SC2206
            ((mock_input_array_index++))
            echo -n "$get_input_result"
        else
            echo "MOCK_INPUT_ARRAY has no more elements" 1>&2
            return 1
        fi
    fi

    return 0
}

#####
# function yes_no_input
#
# This function requests a yes/no answer from the user, following to a prompt.
#
# Parameters:
#   $1 - The prompt.
#
# Returns:
#   0 - If yes.
#   1 - If no.
#####
function yes_no_input() {
    if [ -z "$1" ]; then
        echo "Internal error yes_no_input"
    fi
}

```

```

    return 1
fi

local index=0
local response="N"
while [[ $index -lt 10 ]]; do
    index=$((index + 1))
    echo -n "$1"
    if ! get_input; then
        return 1
    fi
    response=$(echo "$get_input_result" | tr '[:upper:]' '[:lower:]')
    if [ "$response" = "y" ] || [ "$response" = "n" ]; then
        break
    else
        echo -e "\nPlease enter or 'y' or 'n'."
    fi
fi
done

echo

if [ "$response" = "y" ]; then
    return 0
else
    return 1
fi
}

#####
# function integer_input
#
# This function prompts the user to enter an integer within a specified range
# and validates the input.
#
# Parameters:
#     $1 - The prompt message to display to the user.
#     $2 - The minimum value of the accepted range.
#     $3 - The maximum value of the accepted range.
#
# Returns:
#     The valid integer input from the user.
#     If the input is invalid or out of range, the function will continue
#     prompting the user until a valid input is provided.
#####

```

```

function integer_input() {
    local prompt="$1"
    local min_value="$2"
    local max_value="$3"
    local input=""

    while true; do
        # Display the prompt message and wait for user input
        echo -n "$prompt"

        if ! get_input; then
            return 1
        fi

        input="$get_input_result"

        # Check if the input is a valid integer
        if [[ "$input" =~ ^-?[0-9]+$ ]]; then
            # Check if the input is within the specified range
            if ((input >= min_value && input <= max_value)); then
                return 0
            else
                echo "Error: Input, $input, must be between $min_value and $max_value."
            fi
        else
            echo "Error: Invalid input- $input. Please enter an integer."
        fi
    done
}

#####
# function new_line_and_tab_to_list
#
# This function takes a string input containing newlines and tabs, and
# converts it into a list (array) of elements.
#
# Parameters:
#     $1 - The input string containing newlines and tabs.
#
# Returns:
#     The resulting list (array) is stored in the global variable
#     'list_result'.
#####
function new_line_and_tab_to_list() {
    local input=$1

```

```

export list_result

list_result=()
mapfile -t lines <<<"$input"
local line
for line in "${lines[@]"; do
    IFS=$'\t' read -ra parameters <<<"$line"
    list_result+=("${parameters[@]}")
done
}

#####
# function echo_repeat
#
# This function prints a string 'n' times to stdout.
#
# Parameters:
#     $1 - The string.
#     $2 - Number of times to print the string.
#
# Outputs:
#     String 'n' times to stdout.
#
# Returns:
#     0
#####
function echo_repeat() {
    local end=$2
    for ((i = 0; i < end; i++)); do
        echo -n "$1"
    done
    echo
}

```

Fungsi DynamoDB yang digunakan dalam skenario ini.

```

#####
# function ec2_create_keypair
#
# This function creates an Amazon Elastic Compute Cloud (Amazon EC2) ED25519 or
# 2048-bit RSA key pair
# and writes it to a file.

```

```

#
# Parameters:
#   -n key_pair_name - A key pair name.
#   -f file_path - File to store the key pair.
#
# And:
#   0 - If successful.
#   1 - If it fails.
#####
function ec2_create_keypair() {
    local key_pair_name file_path response
    local option OPTARG # Required to use getopt command in a function.

    # bashsupport disable=BP5008
    function usage() {
        echo "function ec2_create_keypair"
        echo "Creates an Amazon Elastic Compute Cloud (Amazon EC2) ED25519 or 2048-bit RSA key pair"
        echo " and writes it to a file."
        echo "  -n key_pair_name - A key pair name."
        echo "  -f file_path - File to store the key pair."
        echo ""
    }

    # Retrieve the calling parameters.
    while getopt "n:f:h" option; do
        case "${option}" in
            n) key_pair_name="${OPTARG}" ;;
            f) file_path="${OPTARG}" ;;
            h)
                usage
                return 0
                ;;
            \?)
                echo "Invalid parameter"
                usage
                return 1
                ;;
        esac
    done
    export OPTIND=1

    if [[ -z "$key_pair_name" ]]; then
        errecho "ERROR: You must provide a key name with the -n parameter."
    fi
}

```



```

usage
return 1
fi

if [[ -z "$file_path" ]]; then
    errecho "ERROR: You must provide a file path with the -f parameter."
    usage
    return 1
fi

response=$(aws ec2 create-key-pair \
    --key-name "$key_pair_name" \
    --query 'KeyMaterial' \
    --output text) || {
    aws_cli_error_log ${?}
    errecho "ERROR: AWS reports create-access-key operation failed.$response"
    return 1
}

if [[ -n "$file_path" ]]; then
    echo "$response" >"$file_path"
fi

return 0
}

#####
# function ec2_describe_key_pairs
#
# This function describes one or more Amazon Elastic Compute Cloud (Amazon EC2)
# key pairs.
#
# Parameters:
#     -h - Display help.
#
# And:
#     0 - If successful.
#     1 - If it fails.
#####
function ec2_describe_key_pairs() {
    local option OPTARG # Required to use getopt command in a function.

    # bashsupport disable=BP5008
    function usage() {

```

```
    echo "function ec2_describe_key_pairs"
    echo "Describes one or more Amazon Elastic Compute Cloud (Amazon EC2) key
pairs."
    echo "  -h - Display help."
    echo ""
}

# Retrieve the calling parameters.
while getopts "h" option; do
    case "${option}" in
        h)
            usage
            return 0
            ;;
        \?)
            echo "Invalid parameter"
            usage
            return 1
            ;;
    esac
done
export OPTIND=1

local response

response=$(aws ec2 describe-key-pairs \
  --query 'KeyPairs[*].[KeyName, KeyFingerprint]' \
  --output text) || {
    aws_cli_error_log ${?}
    errecho "ERROR: AWS reports describe-key-pairs operation failed.${response}"
    return 1
}

echo "${response}"

return 0
}

#####
# function ec2_create_security_group
#
# This function creates an Amazon Elastic Compute Cloud (Amazon EC2) security
group.
#
```

```
# Parameters:
#     -n security_group_name - The name of the security group.
#     -d security_group_description - The description of the security group.
#
# Returns:
#     The ID of the created security group, or an error message if the
#     operation fails.
# And:
#     0 - If successful.
#     1 - If it fails.
#
#####
function ec2_create_security_group() {
    local security_group_name security_group_description response

    # Function to display usage information
    function usage() {
        echo "function ec2_create_security_group"
        echo "Creates an Amazon Elastic Compute Cloud (Amazon EC2) security group."
        echo "  -n security_group_name - The name of the security group."
        echo "  -d security_group_description - The description of the security
group."
        echo ""
    }

    # Parse the command-line arguments
    while getopts "n:d:h" option; do
        case "${option}" in
            n) security_group_name="${OPTARG}" ;;
            d) security_group_description="${OPTARG}" ;;
            h)
                usage
                return 0
                ;;
            \?)
                echo "Invalid parameter"
                usage
                return 1
                ;;
        esac
    done
    export OPTIND=1

    # Validate the input parameters
```

```

if [[ -z "$security_group_name" ]]; then
    errecho "ERROR: You must provide a security group name with the -n
parameter."
    return 1
fi

if [[ -z "$security_group_description" ]]; then
    errecho "ERROR: You must provide a security group description with the -d
parameter."
    return 1
fi

# Create the security group
response=$(aws ec2 create-security-group \
    --group-name "$security_group_name" \
    --description "$security_group_description" \
    --query "GroupId" \
    --output text) || {
    aws_cli_error_log ${?}
    errecho "ERROR: AWS reports create-security-group operation failed."
    errecho "$response"
    return 1
}

echo "$response"
return 0
}

#####
# function ec2_describe_security_groups
#
# This function describes one or more Amazon Elastic Compute Cloud (Amazon EC2)
# security groups.
#
# Parameters:
#     -g security_group_id - The ID of the security group to describe
#     (optional).
#
# And:
#     0 - If successful.
#     1 - If it fails.
#####
function ec2_describe_security_groups() {
    local security_group_id response

```

```
local option OPTARG # Required to use getopt command in a function.

# bashsupport disable=BP5008
function usage() {
    echo "function ec2_describe_security_groups"
    echo "Describes one or more Amazon Elastic Compute Cloud (Amazon EC2)
security groups."
    echo "  -g security_group_id - The ID of the security group to describe
(optional)."
    echo ""
}

# Retrieve the calling parameters.
while getopt "g:h" option; do
    case "${option}" in
        g) security_group_id="${OPTARG}" ;;
        h)
            usage
            return 0
            ;;
        \?)
            echo "Invalid parameter"
            usage
            return 1
            ;;
    esac
done
export OPTIND=1

local query="SecurityGroups[*].[GroupName, GroupId, VpcId, IpPermissions[*].
[IpProtocol, FromPort, ToPort, IpRanges[*].CidrIp]]"

if [[ -n "$security_group_id" ]]; then
    response=$(aws ec2 describe-security-groups --group-ids "$security_group_id"
--query "${query}" --output text)
else
    response=$(aws ec2 describe-security-groups --query "${query}" --output text)
fi

local error_code=${?}

if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
```

```

    errecho "ERROR: AWS reports describe-security-groups operation failed.
$response"
    return 1
fi

echo "$response"

return 0
}

#####
# function ec2_authorize_security_group_ingress
#
# This function authorizes an ingress rule for an Amazon Elastic Compute Cloud
# (Amazon EC2) security group.
#
# Parameters:
#     -g security_group_id - The ID of the security group.
#     -i ip_address - The IP address or CIDR block to authorize.
#     -p protocol - The protocol to authorize (e.g., tcp, udp, icmp).
#     -f from_port - The start of the port range to authorize.
#     -t to_port - The end of the port range to authorize.
#
# And:
#     0 - If successful.
#     1 - If it fails.
#####
function ec2_authorize_security_group_ingress() {
    local security_group_id ip_address protocol from_port to_port response
    local option OPTARG # Required to use getopt command in a function.

    # bashsupport disable=BP5008
    function usage() {
        echo "function ec2_authorize_security_group_ingress"
        echo "Authorizes an ingress rule for an Amazon Elastic Compute Cloud (Amazon
EC2) security group."
        echo " -g security_group_id - The ID of the security group."
        echo " -i ip_address - The IP address or CIDR block to authorize."
        echo " -p protocol - The protocol to authorize (e.g., tcp, udp, icmp)."
        echo " -f from_port - The start of the port range to authorize."
        echo " -t to_port - The end of the port range to authorize."
        echo ""
    }
}

```

```
# Retrieve the calling parameters.
while getopts "g:i:p:f:t:h" option; do
  case "${option}" in
    g) security_group_id="${OPTARG}" ;;
    i) ip_address="${OPTARG}" ;;
    p) protocol="${OPTARG}" ;;
    f) from_port="${OPTARG}" ;;
    t) to_port="${OPTARG}" ;;
    h)
      usage
      return 0
      ;;
    \?)
      echo "Invalid parameter"
      usage
      return 1
      ;;
  esac
done
export OPTIND=1

if [[ -z "$security_group_id" ]]; then
  errecho "ERROR: You must provide a security group ID with the -g parameter."
  usage
  return 1
fi

if [[ -z "$ip_address" ]]; then
  errecho "ERROR: You must provide an IP address or CIDR block with the -i
parameter."
  usage
  return 1
fi

if [[ -z "$protocol" ]]; then
  errecho "ERROR: You must provide a protocol with the -p parameter."
  usage
  return 1
fi

if [[ -z "$from_port" ]]; then
  errecho "ERROR: You must provide a start port with the -f parameter."
  usage
  return 1
fi
```

```

fi

if [[ -z "$to_port" ]]; then
    errecho "ERROR: You must provide an end port with the -t parameter."
    usage
    return 1
fi

response=$(aws ec2 authorize-security-group-ingress \
    --group-id "$security_group_id" \
    --cidr "${ip_address}/32" \
    --protocol "$protocol" \
    --port "$from_port-$to_port" \
    --output text) || {
    aws_cli_error_log ${?}
    errecho "ERROR: AWS reports authorize-security-group-ingress operation
failed.$response"
    return 1
}

return 0
}

#####
# function ec2_describe_images
#
# This function describes one or more Amazon Elastic Compute Cloud (Amazon EC2)
images.
#
# Parameters:
#     -i image_ids - A space-separated list of image IDs (optional).
#     -h - Display help.
#
# And:
#     0 - If successful.
#     1 - If it fails.
#####
function ec2_describe_images() {
    local image_ids response
    local option OPTARG # Required to use getopt command in a function.

    # bashsupport disable=BP5008
    function usage() {
        echo "function ec2_describe_images"
    }
}

```



```
    echo "Describes one or more Amazon Elastic Compute Cloud (Amazon EC2)
images."
    echo "  -i image_ids - A space-separated list of image IDs (optional)."
    echo "  -h - Display help."
    echo ""
}

# Retrieve the calling parameters.
while getopts "i:h" option; do
    case "${option}" in
        i) image_ids="${OPTARG}" ;;
        h)
            usage
            return 0
            ;;
        \?)
            echo "Invalid parameter"
            usage
            return 1
            ;;
    esac
done
export OPTIND=1

local aws_cli_args=()

if [[ -n "$image_ids" ]]; then
    # shellcheck disable=SC2206
    aws_cli_args+=("--image-ids" $image_ids)
fi

response=$(aws ec2 describe-images \
    "${aws_cli_args[@]}" \
    --query 'Images[*].[Description,Architecture,ImageId]' \
    --output text) || {
    aws_cli_error_log ${?}
    errecho "ERROR: AWS reports describe-images operation failed.$response"
    return 1
}

echo "$response"

return 0
}
```

```
#####
# ec2_describe_instance_types
#
# This function describes EC2 instance types filtered by processor architecture
# and optionally by instance type. It takes the following arguments:
#
# -a, --architecture ARCHITECTURE Specify the processor architecture (e.g.,
# x86_64)
# -t, --type INSTANCE_TYPE Comma-separated list of instance types (e.g.,
# t2.micro)
# -h, --help Show the usage help
#
# The function prints the instance type and supported architecture for each
# matching instance type.
#####
function ec2_describe_instance_types() {
    local architecture=""
    local instance_types=""

    # bashsupport disable=BP5008
    function usage() {
        echo "Usage: ec2_describe_instance_types [-a|--architecture ARCHITECTURE] [-t|--type INSTANCE_TYPE] [-h|--help]"
        echo " -a, --architecture ARCHITECTURE Specify the processor architecture (e.g., x86_64)"
        echo " -t, --type INSTANCE_TYPE Comma-separated list of instance types (e.g., t2.micro)"
        echo " -h, --help Show this help message"
    }

    while [[ $# -gt 0 ]]; do
        case "$1" in
            -a | --architecture)
                architecture="$2"
                shift 2
                ;;
            -t | --type)
                instance_types="$2"
                shift 2
                ;;
            -h | --help)
                usage
                return 0
        esac
    done
}

```

```
    ;;
    *)
        echo "Unknown argument: $1"
        return 1
    ;;
esac
done

if [[ -z "$architecture" ]]; then
    errecho "Error: Architecture not specified."
    usage
    return 1
fi

if [[ -z "$instance_types" ]]; then
    errecho "Error: Instance type not specified."
    usage
    return 1
fi

local tmp_json_file="temp_ec2.json"
echo -n '[
    {
        "Name": "processor-info.supported-architecture",
        "Values": [' >"$tmp_json_file"

local items
IFS=', ' read -ra items <<<"$architecture"
local array_size
array_size=${#items[@]}
for i in $(seq 0 $((array_size - 1))); do
    echo -n '""${items[$i]}""' >>"$tmp_json_file"
    if [[ $i -lt $((array_size - 1)) ]]; then
        echo -n ', ' >>"$tmp_json_file"
    fi
done
echo -n ']],
    {
        "Name": "instance-type",
        "Values": [' >>"$tmp_json_file"
IFS=', ' read -ra items <<<"$instance_types"
local array_size
array_size=${#items[@]}
for i in $(seq 0 $((array_size - 1))); do
```

```

    echo -n '""${items[$i]}""' >>"$tmp_json_file"
    if [[ $i -lt $((array_size - 1)) ]]; then
        echo -n ', ' >>"$tmp_json_file"
    fi
done

echo -n ']]]' >>"$tmp_json_file"

local response
response=$(aws ec2 describe-instance-types --filters file://"${tmp_json_file}" \
    --query 'InstanceTypes[*].[InstanceType]' --output text)

local error_code=$?

rm "$tmp_json_file"

if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    echo "ERROR: AWS reports describe-instance-types operation failed."
    return 1
fi

echo "$response"
return 0
}

#####
# function ec2_run_instances
#
# This function launches one or more Amazon Elastic Compute Cloud (Amazon EC2)
# instances.
#
# Parameters:
#     -i image_id - The ID of the Amazon Machine Image (AMI) to use.
#     -t instance_type - The instance type to use (e.g., t2.micro).
#     -k key_pair_name - The name of the key pair to use.
#     -s security_group_id - The ID of the security group to use.
#     -c count - The number of instances to launch (default: 1).
#     -h - Display help.
#
# Returns:
#     0 - If successful.
#     1 - If it fails.
#####

```

```
function ec2_run_instances() {
  local image_id instance_type key_pair_name security_group_id count response
  local option OPTARG # Required to use getopt command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function ec2_run_instances"
    echo "Launches one or more Amazon Elastic Compute Cloud (Amazon EC2)
instances."
    echo "  -i image_id - The ID of the Amazon Machine Image (AMI) to use."
    echo "  -t instance_type - The instance type to use (e.g., t2.micro)."
    echo "  -k key_pair_name - The name of the key pair to use."
    echo "  -s security_group_id - The ID of the security group to use."
    echo "  -c count - The number of instances to launch (default: 1)."
    echo "  -h - Display help."
    echo ""
  }

  # Retrieve the calling parameters.
  while getopt "i:t:k:s:c:h" option; do
    case "${option}" in
      i) image_id="${OPTARG}" ;;
      t) instance_type="${OPTARG}" ;;
      k) key_pair_name="${OPTARG}" ;;
      s) security_group_id="${OPTARG}" ;;
      c) count="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1

  if [[ -z "$image_id" ]]; then
    errecho "ERROR: You must provide an Amazon Machine Image (AMI) ID with the -i
parameter."
    usage
    return 1
  fi
}
```

```
fi

if [[ -z "$instance_type" ]]; then
    errecho "ERROR: You must provide an instance type with the -t parameter."
    usage
    return 1
fi

if [[ -z "$key_pair_name" ]]; then
    errecho "ERROR: You must provide a key pair name with the -k parameter."
    usage
    return 1
fi

if [[ -z "$security_group_id" ]]; then
    errecho "ERROR: You must provide a security group ID with the -s parameter."
    usage
    return 1
fi

if [[ -z "$count" ]]; then
    count=1
fi

response=$(aws ec2 run-instances \
    --image-id "$image_id" \
    --instance-type "$instance_type" \
    --key-name "$key_pair_name" \
    --security-group-ids "$security_group_id" \
    --count "$count" \
    --query 'Instances[*].[InstanceId]' \
    --output text) || {
    aws_cli_error_log ${?}
    errecho "ERROR: AWS reports run-instances operation failed.$response"
    return 1
}

echo "$response"

return 0
}

#####
# function ec2_describe_instances
```

```

#
# This function describes one or more Amazon Elastic Compute Cloud (Amazon EC2)
# instances.
#
# Parameters:
#     -i instance_id - The ID of the instance to describe (optional).
#     -q query - The query to filter the response (optional).
#     -h - Display help.
#
# Returns:
#     0 - If successful.
#     1 - If it fails.
#####
function ec2_describe_instances() {
    local instance_id query response
    local option OPTARG # Required to use getopt command in a function.

    # bashsupport disable=BP5008
    function usage() {
        echo "function ec2_describe_instances"
        echo "Describes one or more Amazon Elastic Compute Cloud (Amazon EC2)
instances."
        echo "  -i instance_id - The ID of the instance to describe (optional)."
        echo "  -q query - The query to filter the response (optional)."
        echo "  -h - Display help."
        echo ""
    }

    # Retrieve the calling parameters.
    while getopt "i:q:h" option; do
        case "${option}" in
            i) instance_id="${OPTARG}" ;;
            q) query="${OPTARG}" ;;
            h)
                usage
                return 0
                ;;
            \?)
                echo "Invalid parameter"
                usage
                return 1
                ;;
        esac
    done
}

```

```

export OPTIND=1

local aws_cli_args=()

if [[ -n "$instance_id" ]]; then
    # shellcheck disable=SC2206
    aws_cli_args+=("--instance-ids" $instance_id)
fi

local query_arg=""
if [[ -n "$query" ]]; then
    query_arg="--query '$query'"
else
    query_arg="--query Reservations[*].Instances[*].
[InstanceId,ImageId,InstanceType,KeyName,VpcId,PublicIpAddress,State.Name]"
fi

# shellcheck disable=SC2086
response=$(aws ec2 describe-instances \
    "${aws_cli_args[@]}" \
    $query_arg \
    --output text) || {
    aws_cli_error_log ${?}
    errecho "ERROR: AWS reports describe-instances operation failed.$response"
    return 1
}

echo "$response"

return 0
}

#####
# function ec2_stop_instances
#
# This function stops one or more Amazon Elastic Compute Cloud (Amazon EC2)
# instances.
#
# Parameters:
#     -i instance_id - The ID(s) of the instance(s) to stop (comma-separated).
#     -h - Display help.
#
# Returns:
#     0 - If successful.

```



```

# 1 - If it fails.
#####
function ec2_stop_instances() {
    local instance_ids
    local option OPTARG # Required to use getopt command in a function.

    # bashsupport disable=BP5008
    function usage() {
        echo "function ec2_stop_instances"
        echo "Stops one or more Amazon Elastic Compute Cloud (Amazon EC2) instances."
        echo " -i instance_id - The ID(s) of the instance(s) to stop (comma-
separated)."
        echo " -h - Display help."
        echo ""
    }

    # Retrieve the calling parameters.
    while getopt "i:h" option; do
        case "${option}" in
            i) instance_ids="${OPTARG}" ;;
            h)
                usage
                return 0
                ;;
            \?)
                echo "Invalid parameter"
                usage
                return 1
                ;;
        esac
    done
    export OPTIND=1

    if [[ -z "$instance_ids" ]]; then
        errecho "ERROR: You must provide one or more instance IDs with the -i
parameter."
        usage
        return 1
    fi

    response=$(aws ec2 stop-instances \
        --instance-ids "${instance_ids}") || {
        aws_cli_error_log ${?}
        errecho "ERROR: AWS reports stop-instances operation failed with $response."
    }
}

```

```
    return 1
}

return 0
}

#####
# function ec2_start_instances
#
# This function starts one or more Amazon Elastic Compute Cloud (Amazon EC2)
# instances.
#
# Parameters:
#     -i instance_id - The ID(s) of the instance(s) to start (comma-separated).
#     -h - Display help.
#
# Returns:
#     0 - If successful.
#     1 - If it fails.
#####
function ec2_start_instances() {
    local instance_ids
    local option OPTARG # Required to use getopt command in a function.

    # bashsupport disable=BP5008
    function usage() {
        echo "function ec2_start_instances"
        echo "Starts one or more Amazon Elastic Compute Cloud (Amazon EC2)
instances."
        echo "  -i instance_id - The ID(s) of the instance(s) to start (comma-
separated)."
        echo "  -h - Display help."
        echo ""
    }

    # Retrieve the calling parameters.
    while getopt "i:h" option; do
        case "${option}" in
            i) instance_ids="${OPTARG}" ;;
            h)
                usage
                return 0
                ;;
            \?)

```

```

        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
done
export OPTIND=1

if [[ -z "$instance_ids" ]]; then
    errecho "ERROR: You must provide one or more instance IDs with the -i
parameter."
    usage
    return 1
fi

response=$(aws ec2 start-instances \
    --instance-ids "${instance_ids}") || {
    aws_cli_error_log ${?}
    errecho "ERROR: AWS reports start-instances operation failed with $response."
    return 1
}

return 0
}

#####
# function ec2_allocate_address
#
# This function allocates an Elastic IP address for use with Amazon Elastic
# Compute Cloud (Amazon EC2) instances in a specific AWS Region.
#
# Parameters:
#     -d domain - The domain for the Elastic IP address (either 'vpc' or
#     'standard').
#
# Returns:
#     The allocated Elastic IP address, or an error message if the operation
#     fails.
# And:
#     0 - If successful.
#     1 - If it fails.
#
#####
function ec2_allocate_address() {

```

```
local domain response

# Function to display usage information
function usage() {
    echo "function ec2_allocate_address"
    echo "Allocates an Elastic IP address for use with Amazon Elastic Compute
Cloud (Amazon EC2) instances in a specific AWS Region."
    echo " -d domain - The domain for the Elastic IP address (either 'vpc' or
'standard')."
    echo ""
}

# Parse the command-line arguments
while getopts "d:h" option; do
    case "${option}" in
        d) domain="${OPTARG}" ;;
        h)
            usage
            return 0
            ;;
        \?)
            echo "Invalid parameter"
            usage
            return 1
            ;;
    esac
done
export OPTIND=1

# Validate the input parameters
if [[ -z "$domain" ]]; then
    errecho "ERROR: You must provide a domain with the -d parameter (either 'vpc'
or 'standard')."
    return 1
fi

if [[ "$domain" != "vpc" && "$domain" != "standard" ]]; then
    errecho "ERROR: Invalid domain value. Must be either 'vpc' or 'standard'."
    return 1
fi

# Allocate the Elastic IP address
response=$(aws ec2 allocate-address \
    --domain "$domain" \
```

```

--query "[PublicIp,AllocationId]" \
--output text) || {
aws_cli_error_log ${?}
errecho "ERROR: AWS reports allocate-address operation failed."
errecho "$response"
return 1
}

echo "$response"
return 0
}

#####
# function ec2_associate_address
#
# This function associates an Elastic IP address with an Amazon Elastic Compute
# Cloud (Amazon EC2) instance.
#
# Parameters:
#   -a allocation_id - The allocation ID of the Elastic IP address to
#   associate.
#   -i instance_id - The ID of the EC2 instance to associate the Elastic IP
#   address with.
#
# Returns:
#   0 - If successful.
#   1 - If it fails.
#
#####
function ec2_associate_address() {
    local allocation_id instance_id response

    # Function to display usage information
    function usage() {
        echo "function ec2_associate_address"
        echo "Associates an Elastic IP address with an Amazon Elastic Compute Cloud
(Amazon EC2) instance."
        echo "  -a allocation_id - The allocation ID of the Elastic IP address to
associate."
        echo "  -i instance_id - The ID of the EC2 instance to associate the Elastic
IP address with."
        echo ""
    }
}

```

```
# Parse the command-line arguments
while getopts "a:i:h" option; do
  case "${option}" in
    a) allocation_id="${OPTARG}" ;;
    i) instance_id="${OPTARG}" ;;
    h)
      usage
      return 0
      ;;
    \?)
      echo "Invalid parameter"
      usage
      return 1
      ;;
  esac
done
export OPTIND=1

# Validate the input parameters
if [[ -z "$allocation_id" ]]; then
  errecho "ERROR: You must provide an allocation ID with the -a parameter."
  return 1
fi

if [[ -z "$instance_id" ]]; then
  errecho "ERROR: You must provide an instance ID with the -i parameter."
  return 1
fi

# Associate the Elastic IP address
response=$(aws ec2 associate-address \
  --allocation-id "$allocation_id" \
  --instance-id "$instance_id" \
  --query "AssociationId" \
  --output text) || {
  aws_cli_error_log ${?}
  errecho "ERROR: AWS reports associate-address operation failed."
  errecho "$response"
  return 1
}

echo "$response"
return 0
}
```

```
#####
# function ec2_disassociate_address
#
# This function disassociates an Elastic IP address from an Amazon Elastic
  Compute Cloud (Amazon EC2) instance.
#
# Parameters:
#   -a association_id - The association ID that represents the association of
  the Elastic IP address with an instance.
#
# And:
#   0 - If successful.
#   1 - If it fails.
#
#####
function ec2_disassociate_address() {
    local association_id response

    # Function to display usage information
    function usage() {
        echo "function ec2_disassociate_address"
        echo "Disassociates an Elastic IP address from an Amazon Elastic Compute
  Cloud (Amazon EC2) instance."
        echo " -a association_id - The association ID that represents the
  association of the Elastic IP address with an instance."
        echo ""
    }

    # Parse the command-line arguments
    while getopts "a:h" option; do
        case "${option}" in
            a) association_id="${OPTARG}" ;;
            h)
                usage
                return 0
                ;;
            \?)
                echo "Invalid parameter"
                usage
                return 1
                ;;
        esac
    done
```

```

export OPTIND=1

# Validate the input parameters
if [[ -z "$association_id" ]]; then
    errecho "ERROR: You must provide an association ID with the -a parameter."
    return 1
fi

response=$(aws ec2 disassociate-address \
    --association-id "$association_id") || {
    aws_cli_error_log ${?}
    errecho "ERROR: AWS reports disassociate-address operation failed."
    errecho "$response"
    return 1
}

return 0
}

#####
# function ec2_release_address
#
# This function releases an Elastic IP address from an Amazon Elastic Compute
# Cloud (Amazon EC2) instance.
#
# Parameters:
#     -a allocation_id - The allocation ID of the Elastic IP address to
#     release.
#
# Returns:
#     0 - If successful.
#     1 - If it fails.
#
#####
function ec2_release_address() {
    local allocation_id response

    # Function to display usage information
    function usage() {
        echo "function ec2_release_address"
        echo "Releases an Elastic IP address from an Amazon Elastic Compute Cloud
        (Amazon EC2) instance."
        echo "    -a allocation_id - The allocation ID of the Elastic IP address to
        release."
    }
}

```



```
    echo ""
}

# Parse the command-line arguments
while getopts "a:h" option; do
    case "${option}" in
        a) allocation_id="${OPTARG}" ;;
        h)
            usage
            return 0
            ;;
        \?)
            echo "Invalid parameter"
            usage
            return 1
            ;;
    esac
done
export OPTIND=1

# Validate the input parameters
if [[ -z "$allocation_id" ]]; then
    errecho "ERROR: You must provide an allocation ID with the -a parameter."
    return 1
fi

response=$(aws ec2 release-address \
    --allocation-id "$allocation_id") || {
    aws_cli_error_log ${?}
    errecho "ERROR: AWS reports release-address operation failed."
    errecho "$response"
    return 1
}

return 0
}

#####
# function ec2_terminate_instances
#
# This function terminates one or more Amazon Elastic Compute Cloud (Amazon EC2)
# instances using the AWS CLI.
#
# Parameters:
```

```

#     -i instance_ids - A space-separated list of instance IDs.
#     -h - Display help.
#
# Returns:
#     0 - If successful.
#     1 - If it fails.
#####
function ec2_terminate_instances() {
    local instance_ids response
    local option OPTARG # Required to use getopt command in a function.

    # bashsupport disable=BP5008
    function usage() {
        echo "function ec2_terminate_instances"
        echo "Terminates one or more Amazon Elastic Compute Cloud (Amazon EC2)
instances."
        echo "  -i instance_ids - A space-separated list of instance IDs."
        echo "  -h - Display help."
        echo ""
    }

    # Retrieve the calling parameters.
    while getopt "i:h" option; do
        case "${option}" in
            i) instance_ids="${OPTARG}" ;;
            h)
                usage
                return 0
                ;;
            \?)
                echo "Invalid parameter"
                usage
                return 1
                ;;
        esac
    done
    export OPTIND=1

    # Check if instance ID is provided
    if [[ -z "${instance_ids}" ]]; then
        echo "Error: Missing required instance IDs parameter."
        usage
        return 1
    fi
}

```

```

# shellcheck disable=SC2086
response=$(aws ec2 terminate-instances \
  "--instance-ids" $instance_ids \
  --query 'TerminatingInstances[*].[InstanceId,CurrentState.Name]' \
  --output text) || {
  aws_cli_error_log ${?}
  errecho "ERROR: AWS reports terminate-instances operation failed.$response"
  return 1
}

return 0
}

#####
# function ec2_delete_security_group
#
# This function deletes an Amazon Elastic Compute Cloud (Amazon EC2) security
# group.
#
# Parameters:
#     -i security_group_id - The ID of the security group to delete.
#
# And:
#     0 - If successful.
#     1 - If it fails.
#####
function ec2_delete_security_group() {
  local security_group_id response
  local option OPTARG # Required to use getopt command in a function.

# bashsupport disable=BP5008
function usage() {
  echo "function ec2_delete_security_group"
  echo "Deletes an Amazon Elastic Compute Cloud (Amazon EC2) security group."
  echo "  -i security_group_id - The ID of the security group to delete."
  echo ""
}

# Retrieve the calling parameters.
while getopt "i:h" option; do
  case "${option}" in
    i) security_group_id="${OPTARG}" ;;
    h)

```

```

        usage
        return 0
        ;;
    \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
done
export OPTIND=1

if [[ -z "$security_group_id" ]]; then
    errecho "ERROR: You must provide a security group ID with the -i parameter."
    usage
    return 1
fi

response=$(aws ec2 delete-security-group --group-id "$security_group_id" --
output text) || {
    aws_cli_error_log ${?}
    errecho "ERROR: AWS reports delete-security-group operation failed.$response"
    return 1
}

return 0
}

#####
# function ec2_delete_keypair
#
# This function deletes an Amazon EC2 ED25519 or 2048-bit RSA key pair.
#
# Parameters:
#     -n key_pair_name - A key pair name.
#
# And:
#     0 - If successful.
#     1 - If it fails.
#####
function ec2_delete_keypair() {
    local key_pair_name response

    local option OPTARG # Required to use getopt command in a function.

```

```
# bashsupport disable=BP5008
function usage() {
    echo "function ec2_delete_keypair"
    echo "Deletes an Amazon EC2 ED25519 or 2048-bit RSA key pair."
    echo "  -n key_pair_name - A key pair name."
    echo ""
}

# Retrieve the calling parameters.
while getopts "n:h" option; do
    case "${option}" in
        n) key_pair_name="${OPTARG}" ;;
        h)
            usage
            return 0
            ;;
        \?)
            echo "Invalid parameter"
            usage
            return 1
            ;;
    esac
done
export OPTIND=1

if [[ -z "$key_pair_name" ]]; then
    errecho "ERROR: You must provide a key pair name with the -n parameter."
    usage
    return 1
fi

response=$(aws ec2 delete-key-pair \
    --key-name "$key_pair_name") || {
    aws_cli_error_log ${?}
    errecho "ERROR: AWS reports delete-key-pair operation failed.$response"
    return 1
}

return 0
}
```

Fungsi utilitas yang digunakan dalam skenario ini.

```
#####  
# function errecho  
#  
# This function outputs everything sent to it to STDERR (standard error output).  
#####  
function errecho() {  
    printf "%s\n" "$*" 1>&2  
}  
  
#####  
# function aws_cli_error_log()  
#  
# This function is used to log the error messages from the AWS CLI.  
#  
# The function expects the following argument:  
#     $1 - The error code returned by the AWS CLI.  
#  
# Returns:  
#     0: - Success.  
#  
#####  
function aws_cli_error_log() {  
    local err_code=$1  
    errecho "Error code : $err_code"  
    if [ "$err_code" == 1 ]; then  
        errecho " One or more S3 transfers failed."  
    elif [ "$err_code" == 2 ]; then  
        errecho " Command line failed to parse."  
    elif [ "$err_code" == 130 ]; then  
        errecho " Process received SIGINT."  
    elif [ "$err_code" == 252 ]; then  
        errecho " Command syntax invalid."  
    elif [ "$err_code" == 253 ]; then  
        errecho " The system environment or configuration was invalid."  
    elif [ "$err_code" == 254 ]; then  
        errecho " The service returned an error."  
    elif [ "$err_code" == 255 ]; then  
        errecho " 255 is a catch-all error."  
    fi  
  
    return 0  
}
```

- Untuk detail API, lihat topik berikut di Referensi Perintah AWS CLI .
 - [AllocateAddress](#)
 - [AssociateAddress](#)
 - [AuthorizeSecurityGroupIngress](#)
 - [CreateKeyPair](#)
 - [CreateSecurityGroup](#)
 - [DeleteKeyPair](#)
 - [DeleteSecurityGroup](#)
 - [DescribeImages](#)
 - [DescribeInstanceTypes](#)
 - [DescribeInstances](#)
 - [DescribeKeyPairs](#)
 - [DescribeSecurityGroups](#)
 - [DisassociateAddress](#)
 - [ReleaseAddress](#)
 - [RunInstances](#)
 - [StartInstances](#)
 - [StopInstances](#)
 - [TerminateInstances](#)
 - [UnmonitorInstances](#)

Java

SDK untuk Java 2.x

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
* environment, including your credentials.
*
* For more information, see the following documentation topic:
*
* https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-started.html
*
* This Java example performs the following tasks:
*
* 1. Creates an RSA key pair and saves the private key data as a .pem file.
* 2. Lists key pairs.
* 3. Creates a security group for the default VPC.
* 4. Displays security group information.
* 5. Gets a list of Amazon Linux 2 AMIs and selects one.
* 6. Gets more information about the image.
* 7. Gets a list of instance types that are compatible with the selected AMI's
* architecture.
* 8. Creates an instance with the key pair, security group, AMI, and an
* instance type.
* 9. Displays information about the instance.
* 10. Stops the instance and waits for it to stop.
* 11. Starts the instance and waits for it to start.
* 12. Allocates an Elastic IP address and associates it with the instance.
* 13. Displays SSH connection info for the instance.
* 14. Disassociates and deletes the Elastic IP address.
* 15. Terminates the instance and waits for it to terminate.
* 16. Deletes the security group.
* 17. Deletes the key pair.
*/
public class EC2Scenario {
    public static final String DASHES = new String(new char[80]).replace("\0",
"-");

    public static void main(String[] args) throws InterruptedException {

        final String usage = ""

            Usage:
                <keyName> <fileName> <groupName> <groupDesc> <vpcId>

            Where:
                keyName - A key pair name (for example, TestKeyPair).\s
                fileName - A file name where the key information is written
to.\s
```



```
        groupName - The name of the security group.\s
        groupDesc - The description of the security group.\s
        vpcId - A VPC Id value. You can get this value from the AWS
Management Console.\s
        myIpAddress - The IP address of your development machine.\s

        """;

if (args.length != 6) {
    System.out.println(usage);
    System.exit(1);
}

String keyName = args[0];
String fileName = args[1];
String groupName = args[2];
String groupDesc = args[3];
String vpcId = args[4];
String myIpAddress = args[5];

Region region = Region.US_WEST_2;
Ec2Client ec2 = Ec2Client.builder()
    .region(region)
    .build();

SsmClient ssmClient = SsmClient.builder()
    .region(region)
    .build();

System.out.println(DASHES);
System.out.println("Welcome to the Amazon EC2 example scenario.");
System.out.println(DASHES);

System.out.println(DASHES);
System.out.println("1. Create an RSA key pair and save the private key
material as a .pem file.");
createKeyPair(ec2, keyName, fileName);
System.out.println(DASHES);

System.out.println(DASHES);
System.out.println("2. List key pairs.");
describeKeys(ec2);
System.out.println(DASHES);
```

```
        System.out.println(DASHES);
        System.out.println("3. Create a security group.");
        String groupId = createSecurityGroup(ec2, groupName, groupDesc, vpcId,
myIpAddress);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("4. Display security group info for the newly created
security group.");
        describeSecurityGroups(ec2, groupId);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("5. Get a list of Amazon Linux 2 AMIs and selects one
with amzn2 in the name.");
        String instanceId = getParaValues(ssmClient);
        System.out.println("The instance Id is " + instanceId);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("6. Get more information about an amzn2 image.");
        String amiValue = describeImage(ec2, instanceId);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("7. Get a list of instance types.");
        String instanceType = getInstanceTypes(ec2);
        System.out.println("The instance type is " + instanceType);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("8. Create an instance.");
        String newInstanceId = runInstance(ec2, instanceType, keyName, groupName,
amiValue);
        System.out.println("The instance Id is " + newInstanceId);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("9. Display information about the running instance.
");
        String ipAddress = describeEC2Instances(ec2, newInstanceId);
        System.out.println("You can SSH to the instance using this command:");
        System.out.println("ssh -i " + fileName + "ec2-user@" + ipAddress);
        System.out.println(DASHES);
```

```
System.out.println(DASHES);
System.out.println("10. Stop the instance and use a waiter.");
stopInstance(ec2, newInstanceId);
System.out.println(DASHES);

System.out.println(DASHES);
System.out.println("11. Start the instance and use a waiter.");
startInstance(ec2, newInstanceId);
ipAddress = describeEC2Instances(ec2, newInstanceId);
System.out.println("You can SSH to the instance using this command:");
System.out.println("ssh -i " + fileName + "ec2-user@" + ipAddress);
System.out.println(DASHES);

System.out.println(DASHES);
System.out.println("12. Allocate an Elastic IP address and associate it
with the instance.");
String allocationId = allocateAddress(ec2);
System.out.println("The allocation Id value is " + allocationId);
String associationId = associateAddress(ec2, newInstanceId,
allocationId);
System.out.println("The associate Id value is " + associationId);
System.out.println(DASHES);

System.out.println(DASHES);
System.out.println("13. Describe the instance again.");
ipAddress = describeEC2Instances(ec2, newInstanceId);
System.out.println("You can SSH to the instance using this command:");
System.out.println("ssh -i " + fileName + "ec2-user@" + ipAddress);
System.out.println(DASHES);

System.out.println(DASHES);
System.out.println("14. Disassociate and release the Elastic IP
address.");
disassociateAddress(ec2, associationId);
releaseEC2Address(ec2, allocationId);
System.out.println(DASHES);

System.out.println(DASHES);
System.out.println("15. Terminate the instance and use a waiter.");
terminateEC2(ec2, newInstanceId);
System.out.println(DASHES);

System.out.println(DASHES);
```

```
System.out.println("16. Delete the security group.");
deleteEC2SecGroup(ec2, groupId);
System.out.println(DASHES);

System.out.println(DASHES);
System.out.println("17. Delete the key.");
deleteKeys(ec2, keyName);
System.out.println(DASHES);

System.out.println(DASHES);
System.out.println("You successfully completed the Amazon EC2
scenario.");
System.out.println(DASHES);
ec2.close();
}

public static void deleteEC2SecGroup(Ec2Client ec2, String groupId) {
    try {
        DeleteSecurityGroupRequest request =
DeleteSecurityGroupRequest.builder()
        .groupId(groupId)
        .build();

        ec2.deleteSecurityGroup(request);
        System.out.println("Successfully deleted security group with Id " +
groupId);

    } catch (Ec2Exception e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
}

public static void terminateEC2(Ec2Client ec2, String instanceId) {
    try {
        Ec2Waiter ec2Waiter = Ec2Waiter.builder()
        .overrideConfiguration(b -> b.maxAttempts(100))
        .client(ec2)
        .build();

        TerminateInstancesRequest ti = TerminateInstancesRequest.builder()
        .instanceIds(instanceId)
        .build();
```

```
        System.out.println("Use an Ec2Waiter to wait for the instance to
terminate. This will take a few minutes.");
        ec2.terminateInstances(ti);
        DescribeInstancesRequest instanceRequest =
DescribeInstancesRequest.builder()
            .instanceIds(instanceId)
            .build();

        WaiterResponse<DescribeInstancesResponse> waiterResponse = ec2Waiter
            .waitUntilInstanceTerminated(instanceRequest);
        waiterResponse.matched().response().ifPresent(System.out::println);
        System.out.println("Successfully started instance " + instanceId);
        System.out.println(instanceId + " is terminated!");

    } catch (Ec2Exception e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
}

public static void deleteKeys(Ec2Client ec2, String keyPair) {
    try {
        DeleteKeyPairRequest request = DeleteKeyPairRequest.builder()
            .keyName(keyPair)
            .build();

        ec2.deleteKeyPair(request);
        System.out.println("Successfully deleted key pair named " + keyPair);

    } catch (Ec2Exception e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
}

public static void releaseEC2Address(Ec2Client ec2, String allocId) {
    try {
        ReleaseAddressRequest request = ReleaseAddressRequest.builder()
            .allocationId(allocId)
            .build();

        ec2.releaseAddress(request);
        System.out.println("Successfully released Elastic IP address " +
allocId);
```

```
    } catch (Ec2Exception e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
}

public static void disassociateAddress(Ec2Client ec2, String associationId) {
    try {
        DisassociateAddressRequest addressRequest =
DisassociateAddressRequest.builder()
            .associationId(associationId)
            .build();

        ec2.disassociateAddress(addressRequest);
        System.out.println("You successfully disassociated the address!");

    } catch (Ec2Exception e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
}

public static String associateAddress(Ec2Client ec2, String instanceId,
String allocationId) {
    try {
        AssociateAddressRequest associateRequest =
AssociateAddressRequest.builder()
            .instanceId(instanceId)
            .allocationId(allocationId)
            .build();

        AssociateAddressResponse associateResponse =
ec2.associateAddress(associateRequest);
        return associateResponse.associationId();

    } catch (Ec2Exception e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
    return "";
}

public static String allocateAddress(Ec2Client ec2) {
    try {
```

```
        AllocateAddressRequest allocateRequest =
AllocateAddressRequest.builder()
        .domain(DomainType.VPC)
        .build();

        AllocateAddressResponse allocateResponse =
ec2.allocateAddress(allocateRequest);
        return allocateResponse.allocationId();

    } catch (Ec2Exception e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
    return "";
}

public static void startInstance(Ec2Client ec2, String instanceId) {
    Ec2Waiter ec2Waiter = Ec2Waiter.builder()
        .overrideConfiguration(b -> b.maxAttempts(100))
        .client(ec2)
        .build();

    StartInstancesRequest request = StartInstancesRequest.builder()
        .instanceIds(instanceId)
        .build();

    System.out.println("Use an Ec2Waiter to wait for the instance to run.
This will take a few minutes.");
    ec2.startInstances(request);
    DescribeInstancesRequest instanceRequest =
DescribeInstancesRequest.builder()
        .instanceIds(instanceId)
        .build();

    WaiterResponse<DescribeInstancesResponse> waiterResponse =
ec2Waiter.waitUntilInstanceRunning(instanceRequest);
    waiterResponse.matched().response().ifPresent(System.out::println);
    System.out.println("Successfully started instance " + instanceId);
}

public static void stopInstance(Ec2Client ec2, String instanceId) {
    Ec2Waiter ec2Waiter = Ec2Waiter.builder()
        .overrideConfiguration(b -> b.maxAttempts(100))
        .client(ec2)
```

```

        .build();
        StopInstancesRequest request = StopInstancesRequest.builder()
            .instanceIds(instanceId)
            .build();

        System.out.println("Use an Ec2Waiter to wait for the instance to stop.
This will take a few minutes.");
        ec2.stopInstances(request);
        DescribeInstancesRequest instanceRequest =
DescribeInstancesRequest.builder()
            .instanceIds(instanceId)
            .build();

        WaiterResponse<DescribeInstancesResponse> waiterResponse =
ec2Waiter.waitUntilInstanceStopped(instanceRequest);
        waiterResponse.matched().response().ifPresent(System.out::println);
        System.out.println("Successfully stopped instance " + instanceId);
    }

    public static String describeEC2Instances(Ec2Client ec2, String
newInstanceId) {
        try {
            String pubAddress = "";
            boolean isRunning = false;
            DescribeInstancesRequest request = DescribeInstancesRequest.builder()
                .instanceIds(newInstanceId)
                .build();

            while (!isRunning) {
                DescribeInstancesResponse response =
ec2.describeInstances(request);
                String state =
response.reservations().get(0).instances().get(0).state().name().name();
                if (state.compareTo("RUNNING") == 0) {
                    System.out.println("Image id is " +
response.reservations().get(0).instances().get(0).imageId());
                    System.out.println(
                        "Instance type is " +
response.reservations().get(0).instances().get(0).instanceType());
                    System.out.println(
                        "Instance state is " +
response.reservations().get(0).instances().get(0).state().name());
                    pubAddress =
response.reservations().get(0).instances().get(0).publicIpAddress();

```



```
        System.out.println("Instance address is " + pubAddress);
        isRunning = true;
    }
}
return pubAddress;
} catch (SsmException e) {
    System.err.println(e.getMessage());
    System.exit(1);
}
return "";
}

public static String runInstance(Ec2Client ec2, String instanceType, String
keyName, String groupName,
    String amiId) {
    try {
        RunInstancesRequest runRequest = RunInstancesRequest.builder()
            .instanceType(instanceType)
            .keyName(keyName)
            .securityGroups(groupName)
            .maxCount(1)
            .minCount(1)
            .imageId(amiId)
            .build();

        System.out.println("Going to start an EC2 instance using a waiter");
        RunInstancesResponse response = ec2.runInstances(runRequest);
        String instanceIdVal = response.instances().get(0).instanceId();
        ec2.waiter().waitUntilInstanceRunning(r ->
r.instanceIds(instanceIdVal));
        System.out.println("Successfully started EC2 instance " +
instanceIdVal + " based on AMI " + amiId);
        return instanceIdVal;

    } catch (SsmException e) {
        System.err.println(e.getMessage());
        System.exit(1);
    }
    return "";
}

// Get a list of instance types.
public static String getInstanceTypes(Ec2Client ec2) {
    String instanceType;
```

```
        try {
            DescribeInstanceTypesRequest typesRequest =
DescribeInstanceTypesRequest.builder()
                .maxResults(10)
                .build();

            DescribeInstanceTypesResponse response =
ec2.describeInstanceTypes(typesRequest);
            List<InstanceTypeInfo> instanceTypes = response.instanceTypes();
            for (InstanceTypeInfo type : instanceTypes) {
                System.out.println("The memory information of this type is " +
type.memoryInfo().sizeInMiB());
                System.out.println("Network information is " +
type.networkInfo().toString());
                System.out.println("Instance type is " +
type.instanceType().toString());
                instanceType = type.instanceType().toString();
                if (instanceType.compareTo("t2.2xlarge") == 0){
                    return instanceType;
                }
            }
        } catch (SsmException e) {
            System.err.println(e.getMessage());
            System.exit(1);
        }
        return "";
    }

    // Display the Description field that corresponds to the instance Id value.
    public static String describeImage(Ec2Client ec2, String instanceId) {
        try {
            DescribeImagesRequest imagesRequest = DescribeImagesRequest.builder()
                .imageIds(instanceId)
                .build();

            DescribeImagesResponse response = ec2.describeImages(imagesRequest);
            System.out.println("The description of the first image is " +
response.images().get(0).description());
            System.out.println("The name of the first image is " +
response.images().get(0).name());

            // Return the image Id value.
            return response.images().get(0).imageId();
        }
    }
}
```

```
    } catch (SsmException e) {
        System.err.println(e.getMessage());
        System.exit(1);
    }
    return "";
}

// Get the Id value of an instance with amzn2 in the name.
public static String getParaValues(SsmClient ssmClient) {
    try {
        GetParametersByPathRequest parameterRequest =
GetParametersByPathRequest.builder()
            .path("/aws/service/ami-amazon-linux-latest")
            .build();

        GetParametersByPathIterable responses =
ssmClient.getParametersByPathPaginator(parameterRequest);
        for
(ssoftware.amazon.awssdk.services.ssm.model.GetParametersByPathResponse
response : responses) {
            System.out.println("Test " + response.nextToken());
            List<Parameter> parameterList = response.parameters();
            for (Parameter para : parameterList) {
                System.out.println("The name of the para is: " +
para.name());
                System.out.println("The type of the para is: " +
para.type());
                if (filterName(para.name())) {
                    return para.value();
                }
            }
        }

    } catch (SsmException e) {
        System.err.println(e.getMessage());
        System.exit(1);
    }
    return "";
}

// Return true if the name has amzn2 in it. For example:
// /aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-arm64-gp2
private static boolean filterName(String name) {
```

```
String[] parts = name.split("/");
String myValue = parts[4];
return myValue.contains("amzn2");
}

public static void describeSecurityGroups(Ec2Client ec2, String groupId) {
    try {
        DescribeSecurityGroupsRequest request =
DescribeSecurityGroupsRequest.builder()
            .groupIds(groupId)
            .build();

        // Use a paginator.
        DescribeSecurityGroupsIterable listGroups =
ec2.describeSecurityGroupsPaginator(request);
        listGroups.stream()
            .flatMap(r -> r.securityGroups().stream())
            .forEach(group -> System.out
                .println(" Group id: " +group.groupId() + " group name = " +
group.groupName()));

    } catch (Ec2Exception e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
}

public static String createSecurityGroup(Ec2Client ec2, String groupName,
String groupDesc, String vpcId,
String myIpAddress) {
    try {
        CreateSecurityGroupRequest createRequest =
CreateSecurityGroupRequest.builder()
            .groupName(groupName)
            .description(groupDesc)
            .vpcId(vpcId)
            .build();

        CreateSecurityGroupResponse resp =
ec2.createSecurityGroup(createRequest);
        IpRange ipRange = IpRange.builder()
            .cidrIp(myIpAddress + "/0")
            .build();
```

```
        IpPermission ipPerm = IpPermission.builder()
            .ipProtocol("tcp")
            .toPort(80)
            .fromPort(80)
            .ipRanges(ipRange)
            .build();

        IpPermission ipPerm2 = IpPermission.builder()
            .ipProtocol("tcp")
            .toPort(22)
            .fromPort(22)
            .ipRanges(ipRange)
            .build();

        AuthorizeSecurityGroupIngressRequest authRequest =
AuthorizeSecurityGroupIngressRequest.builder()
            .groupName(groupName)
            .ipPermissions(ipPerm, ipPerm2)
            .build();

        ec2.authorizeSecurityGroupIngress(authRequest);
        System.out.println("Successfully added ingress policy to security
group " + groupName);
        return resp.groupId();

    } catch (Ec2Exception e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
    return "";
}

public static void describeKeys(Ec2Client ec2) {
    try {
        DescribeKeyPairsResponse response = ec2.describeKeyPairs();
        response.keyPairs().forEach(keyPair -> System.out.printf(
            "Found key pair with name %s " +
                "and fingerprint %s",
            keyPair.keyName(),
            keyPair.keyFingerprint()));
    } catch (Ec2Exception e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
}
```

```
    }  
  }  
  
  public static void createKeyPair(Ec2Client ec2, String keyName, String  
fileName) {  
    try {  
      CreateKeyPairRequest request = CreateKeyPairRequest.builder()  
        .keyName(keyName)  
        .build();  
  
      CreateKeyPairResponse response = ec2.createKeyPair(request);  
      String content = response.keyMaterial();  
      BufferedWriter writer = new BufferedWriter(new FileWriter(fileName));  
      writer.write(content);  
      writer.close();  
      System.out.println("Successfully created key pair named " + keyName);  
  
    } catch (Ec2Exception | IOException e) {  
      System.err.println(e.getMessage());  
      System.exit(1);  
    }  
  }  
}
```

- Untuk detail API, lihat topik berikut di Referensi API AWS SDK for Java 2.x .
 - [AllocateAddress](#)
 - [AssociateAddress](#)
 - [AuthorizeSecurityGroupIngress](#)
 - [CreateKeyPair](#)
 - [CreateSecurityGroup](#)
 - [DeleteKeyPair](#)
 - [DeleteSecurityGroup](#)
 - [DescribeImages](#)
 - [DescribeInstanceTypes](#)
 - [DescribeInstances](#)
 - [DescribeKeyPairs](#)
 - [DescribeSecurityGroups](#)

- [DisassociateAddress](#)
- [ReleaseAddress](#)
- [RunInstances](#)
- [StartInstances](#)
- [StopInstances](#)
- [TerminateInstances](#)
- [UnmonitorInstances](#)

JavaScript

SDK untuk JavaScript (v3)

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankan di [Repositori Contoh AWS Kode](#).

Jalankan skenario interaktif di prompt perintah.

```
import { mkdtempSync, writeFileSync, rmSync } from "fs";
import { tmpdir } from "os";
import { join } from "path";
import { get } from "http";

import {
  AllocateAddressCommand,
  AssociateAddressCommand,
  AuthorizeSecurityGroupIngressCommand,
  CreateKeyPairCommand,
  CreateSecurityGroupCommand,
  DeleteKeyPairCommand,
  DeleteSecurityGroupCommand,
  DescribeInstancesCommand,
  DescribeKeyPairsCommand,
  DescribeSecurityGroupsCommand,
  DisassociateAddressCommand,
  EC2Client,
  paginateDescribeImages,
```

```
paginateDescribeInstanceTypes,
ReleaseAddressCommand,
RunInstancesCommand,
StartInstancesCommand,
StopInstancesCommand,
TerminateInstancesCommand,
waitUntilInstanceStatusOk,
waitUntilInstanceStopped,
waitUntilInstanceTerminated,
} from "@aws-sdk/client-ec2";
import { paginateGetParametersByPath, SSMClient } from "@aws-sdk/client-ssm";

import { wrapText } from "@aws-doc-sdk-examples/lib/utils/util-string.js";
import { Prompter } from "@aws-doc-sdk-examples/lib/prompter.js";

const ec2Client = new EC2Client();
const ssmClient = new SSMClient();

const prompter = new Prompter();
const confirmMessage = "Continue?";
const tmpDirectory = mkdtempSync(join(tmpdir(), "ec2-scenario-tmp"));

const createKeyPair = async (keyPairName) => {
  // Create a key pair in Amazon EC2.
  const { KeyMaterial, KeyPairId } = await ec2Client.send(
    // A unique name for the key pair. Up to 255 ASCII characters.
    new CreateKeyPairCommand({ KeyName: keyPairName }),
  );

  // Save the private key in a temporary location.
  writeFileSync(`${tmpDirectory}/${keyPairName}.pem`, KeyMaterial, {
    mode: 0o400,
  });

  return KeyPairId;
};

const describeKeyPair = async (keyPairName) => {
  const command = new DescribeKeyPairsCommand({
    KeyNames: [keyPairName],
  });
  const { KeyPairs } = await ec2Client.send(command);
  return KeyPairs[0];
};
```



```
const createSecurityGroup = async (securityGroupName) => {
  const command = new CreateSecurityGroupCommand({
    GroupName: securityGroupName,
    Description: "A security group for the Amazon EC2 example.",
  });
  const { GroupId } = await ec2Client.send(command);
  return GroupId;
};

const allocateIpAddress = async () => {
  const command = new AllocateAddressCommand({});
  const { PublicIp, AllocationId } = await ec2Client.send(command);
  return { PublicIp, AllocationId };
};

const getLocalIpAddress = () => {
  return new Promise((res, rej) => {
    get("http://checkip.amazonaws.com", (response) => {
      let data = "";
      response.on("data", (chunk) => (data += chunk));
      response.on("end", () => res(data.trim()));
    }).on("error", (err) => {
      rej(err);
    });
  });
};

const authorizeSecurityGroupIngress = async (securityGroupId) => {
  const ipAddress = await getLocalIpAddress();
  const command = new AuthorizeSecurityGroupIngressCommand({
    GroupId: securityGroupId,
    IpPermissions: [
      {
        IpProtocol: "tcp",
        FromPort: 22,
        ToPort: 22,
        IpRanges: [{ CidrIp: `${ipAddress}/32` }],
      },
    ],
  });
  await ec2Client.send(command);
  return ipAddress;
};
```

```
};

const describeSecurityGroup = async (securityGroupName) => {
  const command = new DescribeSecurityGroupsCommand({
    GroupNames: [securityGroupName],
  });
  const { SecurityGroups } = await ec2Client.send(command);

  return SecurityGroups[0];
};

const getAmznLinux2AMIs = async () => {
  const AMIs = [];
  for await (const page of paginateGetParametersByPath(
    {
      client: ssmClient,
    },
    { Path: "/aws/service/ami-amazon-linux-latest" },
  )) {
    page.Parameters.forEach((param) => {
      if (param.Name.includes("amzn2")) {
        AMIs.push(param.Value);
      }
    });
  }

  const imageDetails = [];

  for await (const page of paginateDescribeImages(
    { client: ec2Client },
    { ImageIds: AMIs },
  )) {
    imageDetails.push(...(page.Images || []));
  }

  const choices = imageDetails.map((image, index) => ({
    name: `${image.ImageId} - ${image.Description}`,
    value: index,
  }));

  /**
   * @type {number}
   */
  const selectedIndex = await prompter.select({
```

```
    message: "Select an image.",
    choices,
  });

  return imageDetails[selectedIndex];
};

/**
 * @param {import('@aws-sdk/client-ec2').Image} imageDetails
 */
const getCompatibleInstanceTypes = async (imageDetails) => {
  const paginator = paginateDescribeInstanceTypes(
    { client: ec2Client, pageSize: 25 },
    [
      Filters: [
        {
          Name: "processor-info.supported-architecture",
          Values: [imageDetails.Architecture],
        },
        { Name: "instance-type", Values: ["*.micro", "*.small"] },
      ],
    ],
  );

  const instanceTypes = [];

  for await (const page of paginator) {
    if (page.InstanceTypes.length) {
      instanceTypes.push(...(page.InstanceTypes || []));
    }
  }

  const choices = instanceTypes.map((type, index) => ({
    name: `${type.InstanceType} - Memory:${type.MemoryInfo.SizeInMiB}`,
    value: index,
  }));

  /**
   * @type {number}
   */
  const selectedIndex = await prompter.select({
    message: "Select an instance type.",
    choices,
  });
};
```

```
    return instanceTypes[selectedIndex];
  };

const runInstance = async ({
  keyPairName,
  securityGroupId,
  imageId,
  instanceType,
}) => {
  const command = new RunInstancesCommand({
    KeyName: keyPairName,
    SecurityGroupIds: [securityGroupId],
    ImageId: imageId,
    InstanceType: instanceType,
    MinCount: 1,
    MaxCount: 1,
  });

  const { Instances } = await ec2Client.send(command);
  await waitUntilInstanceStatusOk(
    { client: ec2Client },
    { InstanceIds: [Instances[0].InstanceId] },
  );
  return Instances[0].InstanceId;
};

const describeInstance = async (instanceId) => {
  const command = new DescribeInstancesCommand({
    InstanceIds: [instanceId],
  });

  const { Reservations } = await ec2Client.send(command);
  return Reservations[0].Instances[0];
};

const displaySSHConnectionInfo = ({ publicIp, keyPairName }) => {
  return `ssh -i ${tmpDirectory}/${keyPairName}.pem ec2-user@${publicIp}`;
};

const stopInstance = async (instanceId) => {
  const command = new StopInstancesCommand({ InstanceIds: [instanceId] });
  await ec2Client.send(command);
  await waitUntilInstanceStopped(
    { client: ec2Client },
```

```
    { InstanceIds: [instanceId] },
  );
};

const startInstance = async (instanceId) => {
  const startCommand = new StartInstancesCommand({ InstanceIds: [instanceId] });
  await ec2Client.send(startCommand);
  await waitUntilInstanceStatusOk(
    { client: ec2Client },
    { InstanceIds: [instanceId] },
  );
  return await describeInstance(instanceId);
};

const associateAddress = async ({ allocationId, instanceId }) => {
  const command = new AssociateAddressCommand({
    AllocationId: allocationId,
    InstanceId: instanceId,
  });

  const { AssociationId } = await ec2Client.send(command);
  return AssociationId;
};

const disassociateAddress = async (associationId) => {
  const command = new DisassociateAddressCommand({
    AssociationId: associationId,
  });
  try {
    await ec2Client.send(command);
  } catch (err) {
    console.warn(
      `Failed to disassociated address with association id: ${associationId}`,
      err,
    );
  }
};

const releaseAddress = async (allocationId) => {
  const command = new ReleaseAddressCommand({
    AllocationId: allocationId,
  });

  try {
```

```
    await ec2Client.send(command);
    console.log(`Address with allocation ID ${allocationId} released.\n`);
  } catch (err) {
    console.log(
      `Failed to release address with allocation id: ${allocationId}.`,
      err,
    );
  }
};

const restartInstance = async (instanceId) => {
  console.log("Stopping instance.");
  await stopInstance(instanceId);
  console.log("Instance stopped.");
  console.log("Starting instance.");
  const { PublicIpAddress } = await startInstance(instanceId);
  return PublicIpAddress;
};

const terminateInstance = async (instanceId) => {
  const command = new TerminateInstancesCommand({
    InstanceIds: [instanceId],
  });

  try {
    await ec2Client.send(command);
    await waitUntilInstanceTerminated(
      { client: ec2Client },
      { InstanceIds: [instanceId] },
    );
    console.log(`Instance with ID ${instanceId} terminated.\n`);
  } catch (err) {
    console.warn(`Failed to terminate instance ${instanceId}.`, err);
  }
};

const deleteSecurityGroup = async (securityGroupId) => {
  const command = new DeleteSecurityGroupCommand({
    GroupId: securityGroupId,
  });

  try {
    await ec2Client.send(command);
    console.log(`Security group ${securityGroupId} deleted.\n`);
  }
};
```

```
    } catch (err) {
      console.warn(`Failed to delete security group ${securityGroupId}.`, err);
    }
  };

const deleteKeyPair = async (keyPairName) => {
  const command = new DeleteKeyPairCommand({
    KeyName: keyPairName,
  });

  try {
    await ec2Client.send(command);
    console.log(`Key pair ${keyPairName} deleted.\n`);
  } catch (err) {
    console.warn(`Failed to delete key pair ${keyPairName}.`, err);
  }
};

const deleteTemporaryDirectory = () => {
  try {
    rmSync(tmpDirectory, { recursive: true });
    console.log(`Temporary directory ${tmpDirectory} deleted.\n`);
  } catch (err) {
    console.warn(`Failed to delete temporary directory ${tmpDirectory}.`, err);
  }
};

export const main = async () => {
  const keyPairName = "ec2-scenario-key-pair";
  const securityGroupName = "ec2-scenario-security-group";

  let securityGroupId, ipAllocationId, publicIp, instanceId, associationId;

  console.log(wrapText("Welcome to the Amazon EC2 basic usage scenario.));

  try {
    // Prerequisites
    console.log(
      "Before you launch an instance, you'll need a few things:",
      "\n - A Key Pair",
      "\n - A Security Group",
      "\n - An IP Address",
      "\n - An AMI",
      "\n - A compatible instance type",
    );
  }
};
```

```
    "\n\n I'll go ahead and take care of the first three, but I'll need your
help for the rest.",
  );

  await prompter.confirm({ message: confirmMessage });

  await createKeyPair(keyPairName);
  securityGroupId = await createSecurityGroup(securityGroupName);
  const { PublicIp, AllocationId } = await allocateIpAddress();
  ipAllocationId = AllocationId;
  publicIp = PublicIp;
  const ipAddress = await authorizeSecurityGroupIngress(securityGroupId);

  const { KeyName } = await describeKeyPair(keyPairName);
  const { GroupName } = await describeSecurityGroup(securityGroupName);
  console.log(`# created the key pair ${KeyName}.\n`);
  console.log(
    `# created the security group ${GroupName}`,
    `and allowed SSH access from ${ipAddress} (your IP).\n`,
  );
  console.log(`# allocated ${publicIp} to be used for your EC2 instance.\n`);

  await prompter.confirm({ message: confirmMessage });

  // Creating the instance
  console.log(wrapText("Create the instance."));
  console.log(
    "You get to choose which image you want. Select an amazon-linux-2 image
from the following:",
  );
  );
  const imageDetails = await getAmznLinux2AMIs();
  const instanceTypeDetails = await getCompatibleInstanceTypes(imageDetails);
  console.log("Creating your instance. This can take a few seconds.");
  instanceId = await runInstance({
    keyPairName,
    securityGroupId,
    imageId: imageDetails.ImageId,
    instanceType: instanceTypeDetails.InstanceType,
  });
  const instanceDetails = await describeInstance(instanceId);
  console.log(`# instance ${instanceId}.\n`);
  console.log(instanceDetails);
  console.log(
```



```
\nYou should now be able to SSH into your instance from another
terminal:`,
  \n${displaySSHConnectionInfo({
    publicIp: instanceDetails.PublicIpAddress,
    keyPairName,
  })}`,
);

await prompter.confirm({ message: confirmMessage });

// Understanding the IP address.
console.log(wrapText("Understanding the IP address."));
console.log(
  "When you stop and start an instance, the IP address will change. I'll
restart your",
  "instance for you. Notice how the IP address changes.",
);
const ipAddressAfterRestart = await restartInstance(instanceId);
console.log(
  \n Instance started. The IP address changed from
${instanceDetails.PublicIpAddress} to ${ipAddressAfterRestart}`,
  \n${displaySSHConnectionInfo({
    publicIp: ipAddressAfterRestart,
    keyPairName,
  })}`,
);
await prompter.confirm({ message: confirmMessage });
console.log(
  `If you want to the IP address to be static, you can associate an
allocated`,
  `IP address to your instance. I allocated ${publicIp} for you earlier, and
now I'll associate it to your instance.`,
);
associationId = await associateAddress({
  allocationId: ipAllocationId,
  instanceId,
});
console.log(
  "Done. Now you should be able to SSH using the new IP.\n",
  `${displaySSHConnectionInfo({ publicIp, keyPairName })}`,
);
await prompter.confirm({ message: confirmMessage });
console.log(
```

```
    "I'll restart the server again so you can see the IP address remains the
    same.",
    );
    const ipAddressAfterAssociated = await restartInstance(instanceId);
    console.log(
      `Done. Here's your SSH info. Notice the IP address hasn't changed.` ,
      `\n${displaySSHConnectionInfo({
        publicIp: ipAddressAfterAssociated,
        keyPairName,
      })}` ,
    );
    await prompter.confirm({ message: confirmMessage });
  } catch (err) {
    console.error(err);
  } finally {
    // Clean up.
    console.log(wrapText("Clean up."));
    console.log("Now I'll clean up all of the stuff I created.");
    await prompter.confirm({ message: confirmMessage });
    console.log("Cleaning up. Some of these steps can take a bit of time.");
    await disassociateAddress(associationId);
    await terminateInstance(instanceId);
    await releaseAddress(ipAllocationId);
    await deleteSecurityGroup(securityGroupId);
    deleteTemporaryDirectory();
    await deleteKeyPair(keyPairName);
    console.log(
      "Done cleaning up. Thanks for staying until the end!",
      "If you have any feedback please use the feedback button in the docs",
      "or create an issue on GitHub.",
    );
  }
};
```

- Untuk detail API, lihat topik berikut di Referensi API AWS SDK for JavaScript .
 - [AllocateAddress](#)
 - [AssociateAddress](#)
 - [AuthorizeSecurityGroupIngress](#)
 - [CreateKeyPair](#)
 - [CreateSecurityGroup](#)

- [DeleteKeyPair](#)
- [DeleteSecurityGroup](#)
- [DescribeImages](#)
- [DescribeInstanceTypes](#)
- [DescribeInstances](#)
- [DescribeKeyPairs](#)
- [DescribeSecurityGroups](#)
- [DisassociateAddress](#)
- [ReleaseAddress](#)
- [RunInstances](#)
- [StartInstances](#)
- [StopInstances](#)
- [TerminateInstances](#)
- [UnmonitorInstances](#)

Kotlin

SDK untuk Kotlin

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [AWS Repositori Contoh Kode](#).

```
/**
```

```
Before running this Kotlin code example, set up your development environment, including your credentials.
```

```
For more information, see the following documentation topic:
```

```
https://docs.aws.amazon.com/sdk-for-kotlin/latest/developer-guide/setup.html
```

```
This Kotlin example performs the following tasks:
```

1. Creates an RSA key pair and saves the private key data as a .pem file.

2. Lists key pairs.
 3. Creates a security group for the default VPC.
 4. Displays security group information.
 5. Gets a list of Amazon Linux 2 AMIs and selects one.
 6. Gets more information about the image.
 7. Gets a list of instance types that are compatible with the selected AMI's architecture.
 8. Creates an instance with the key pair, security group, AMI, and an instance type.
 9. Displays information about the instance.
 10. Stops the instance and waits for it to stop.
 11. Starts the instance and waits for it to start.
 12. Allocates an Elastic IP address and associates it with the instance.
 13. Displays SSH connection info for the instance.
 14. Disassociates and deletes the Elastic IP address.
 15. Terminates the instance.
 16. Deletes the security group.
 17. Deletes the key pair.
- */

```
val DASHES = String(CharArray(80)).replace("\u0000", "-")
```

```
suspend fun main(args: Array<String>) {
    val usage = """
        Usage:
            <keyName> <fileName> <groupName> <groupDesc> <vpcId> <myIpAddress>

        Where:
            keyName - A key pair name (for example, TestKeyPair).
            fileName - A file name where the key information is written to.
            groupName - The name of the security group.
            groupDesc - The description of the security group.
            vpcId - A VPC ID. You can get this value from the AWS Management
Console.
            myIpAddress - The IP address of your development machine.

        """

    if (args.size != 6) {
        println(usage)
        exitProcess(0)
    }

    val keyName = args[0]
```

```
val fileName = args[1]
val groupName = args[2]
val groupDesc = args[3]
val vpcId = args[4]
val myIpAddress = args[5]
var newInstanceId: String? = ""

println(DASHES)
println("Welcome to the Amazon EC2 example scenario.")
println(DASHES)

println(DASHES)
println("1. Create an RSA key pair and save the private key material as
a .pem file.")
createKeyPairSc(keyName, fileName)
println(DASHES)

println(DASHES)
println("2. List key pairs.")
describeEC2KeysSc()
println(DASHES)

println(DASHES)
println("3. Create a security group.")
val groupId = createEC2SecurityGroupSc(groupName, groupDesc, vpcId,
myIpAddress)
println(DASHES)

println(DASHES)
println("4. Display security group info for the newly created security
group.")
describeSecurityGroupsSc(groupId.toString())
println(DASHES)

println(DASHES)
println("5. Get a list of Amazon Linux 2 AMIs and select one with amzn2 in
the name.")
val instanceId = getParaValuesSc()
if (instanceId == "") {
    println("The instance Id value isn't valid.")
    exitProcess(0)
}
println("The instance Id is $instanceId.")
println(DASHES)
```

```
println(DASHES)
println("6. Get more information about an amzn2 image and return the AMI
value.")
val amiValue = instanceId?.let { describeImageSc(it) }
if (instanceId == "") {
    println("The instance Id value is invalid.")
    exitProcess(0)
}
println("The AMI value is $amiValue.")
println(DASHES)

println(DASHES)
println("7. Get a list of instance types.")
val instanceType = getInstanceTypesSc()
println(DASHES)

println(DASHES)
println("8. Create an instance.")
if (amiValue != null) {
    newInstanceId = runInstanceSc(instanceType, keyName, groupName, amiValue)
    println("The instance Id is $newInstanceId")
}
println(DASHES)

println(DASHES)
println("9. Display information about the running instance. ")
var ipAddress = describeEC2InstancesSc(newInstanceId)
println("You can SSH to the instance using this command:")
println("ssh -i " + fileName + "ec2-user@" + ipAddress)
println(DASHES)

println(DASHES)
println("10. Stop the instance.")
if (newInstanceId != null) {
    stopInstanceSc(newInstanceId)
}
println(DASHES)

println(DASHES)
println("11. Start the instance.")
if (newInstanceId != null) {
    startInstanceSc(newInstanceId)
}
}
```

```
ipAddress = describeEC2InstancesSc(newInstanceId)
println("You can SSH to the instance using this command:")
println("ssh -i " + fileName + "ec2-user@" + ipAddress)
println(DASHES)

println(DASHES)
println("12. Allocate an Elastic IP address and associate it with the
instance.")
val allocationId = allocateAddressSc()
println("The allocation Id value is $allocationId")
val associationId = associateAddressSc(newInstanceId, allocationId)
println("The associate Id value is $associationId")
println(DASHES)

println(DASHES)
println("13. Describe the instance again.")
ipAddress = describeEC2InstancesSc(newInstanceId)
println("You can SSH to the instance using this command:")
println("ssh -i " + fileName + "ec2-user@" + ipAddress)
println(DASHES)

println(DASHES)
println("14. Disassociate and release the Elastic IP address.")
disassociateAddressSc(associationId)
releaseEC2AddressSc(allocationId)
println(DASHES)

println(DASHES)
println("15. Terminate the instance and use a waiter.")
if (newInstanceId != null) {
    terminateEC2Sc(newInstanceId)
}
println(DASHES)

println(DASHES)
println("16. Delete the security group.")
if (groupId != null) {
    deleteEC2SecGroupSc(groupId)
}
println(DASHES)

println(DASHES)
println("17. Delete the key pair.")
deleteKeysSc(keyName)
```

```
println(DASHES)

println(DASHES)
println("You successfully completed the Amazon EC2 scenario.")
println(DASHES)
}

suspend fun deleteKeysSc(keyPair: String) {
    val request =
        DeleteKeyPairRequest {
            keyName = keyPair
        }
    Ec2Client { region = "us-west-2" }.use { ec2 ->
        ec2.deleteKeyPair(request)
        println("Successfully deleted key pair named $keyPair")
    }
}

suspend fun deleteEC2SecGroupSc(groupIdVal: String) {
    val request =
        DeleteSecurityGroupRequest {
            groupId = groupIdVal
        }
    Ec2Client { region = "us-west-2" }.use { ec2 ->
        ec2.deleteSecurityGroup(request)
        println("Successfully deleted security group with Id $groupIdVal")
    }
}

suspend fun terminateEC2Sc(instanceIdVal: String) {
    val ti =
        TerminateInstancesRequest {
            instanceIds = listOf(instanceIdVal)
        }
    println("Wait for the instance to terminate. This will take a few minutes.")
    Ec2Client { region = "us-west-2" }.use { ec2 ->
        ec2.terminateInstances(ti)
        ec2.waitForInstanceTerminated {
            // suspend call
            instanceIds = listOf(instanceIdVal)
        }
        println("$instanceIdVal is terminated!")
    }
}
```



```
suspend fun releaseEC2AddressSc(allocId: String?) {
    val request =
        ReleaseAddressRequest {
            allocationId = allocId
        }

    Ec2Client { region = "us-west-2" }.use { ec2 ->
        ec2.releaseAddress(request)
        println("Successfully released Elastic IP address $allocId")
    }
}

suspend fun disassociateAddressSc(associationIdVal: String?) {
    val addressRequest =
        DisassociateAddressRequest {
            associationId = associationIdVal
        }

    Ec2Client { region = "us-west-2" }.use { ec2 ->
        ec2.disassociateAddress(addressRequest)
        println("You successfully disassociated the address!")
    }
}

suspend fun associateAddressSc(
    instanceIdVal: String?,
    allocationIdVal: String?,
): String? {
    val associateRequest =
        AssociateAddressRequest {
            instanceId = instanceIdVal
            allocationId = allocationIdVal
        }

    Ec2Client { region = "us-west-2" }.use { ec2 ->
        val associateResponse = ec2.associateAddress(associateRequest)
        return associateResponse.associationId
    }
}

suspend fun allocateAddressSc(): String? {
    val allocateRequest =
        AllocateAddressRequest {
            domain = DomainType.Vpc
        }
}
```

```
    }
    Ec2Client { region = "us-west-2" }.use { ec2 ->
        val allocateResponse = ec2.allocateAddress(allocateRequest)
        return allocateResponse.allocationId
    }
}

suspend fun startInstanceSc(instanceId: String) {
    val request =
        StartInstancesRequest {
            instanceIds = listOf(instanceId)
        }

    Ec2Client { region = "us-west-2" }.use { ec2 ->
        ec2.startInstances(request)
        println("Waiting until instance $instanceId starts. This will take a few
minutes.")
        ec2.waitForInstanceRunning {
            // suspend call
            instanceIds = listOf(instanceId)
        }
        println("Successfully started instance $instanceId")
    }
}

suspend fun stopInstanceSc(instanceId: String) {
    val request =
        StopInstancesRequest {
            instanceIds = listOf(instanceId)
        }

    Ec2Client { region = "us-west-2" }.use { ec2 ->
        ec2.stopInstances(request)
        println("Waiting until instance $instanceId stops. This will take a few
minutes.")
        ec2.waitForInstanceStopped {
            // suspend call
            instanceIds = listOf(instanceId)
        }
        println("Successfully stopped instance $instanceId")
    }
}

suspend fun describeEC2InstancesSc(newInstanceId: String?): String {
```

```

var pubAddress = ""
var isRunning = false
val request =
    DescribeInstancesRequest {
        instanceIds = listOf(newInstanceId.toString())
    }

while (!isRunning) {
    Ec2Client { region = "us-west-2" }.use { ec2 ->
        val response = ec2.describeInstances(request)
        val state =
            response.reservations
                ?.get(0)
                ?.instances
                ?.get(0)
                ?.state
                ?.name
                ?.value
        if (state != null) {
            if (state.compareTo("running") == 0) {
                println("Image id is
${response.reservations!!.get(0).instances?.get(0)?.imageId}")
                println("Instance type is
${response.reservations!!.get(0).instances?.get(0)?.instanceType}")
                println("Instance state is
${response.reservations!!.get(0).instances?.get(0)?.state}")
                pubAddress =
                    response.reservations!!
                        .get(0)
                        .instances
                        ?.get(0)
                        ?.publicIpAddress
                        .toString()
                println("Instance address is $pubAddress")
                isRunning = true
            }
        }
    }
}
return pubAddress
}

suspend fun runInstanceSc(
    instanceTypeVal: String,

```

```

    keyNameVal: String,
    groupNameVal: String,
    amiIdVal: String,
): String {
    val runRequest =
        RunInstancesRequest {
            instanceType = InstanceType.fromValue(instanceTypeVal)
            keyName = keyNameVal
            securityGroups = listOf(groupNameVal)
            maxCount = 1
            minCount = 1
            imageId = amiIdVal
        }

    Ec2Client { region = "us-west-2" }.use { ec2 ->
        val response = ec2.runInstances(runRequest)
        val instanceId = response.instances?.get(0)?.instanceId
        println("Successfully started EC2 Instance $instanceId based on AMI
$amiIdVal")
        return instanceId.toString()
    }
}

// Get a list of instance types.
suspend fun getInstanceTypesSc(): String {
    var instanceType = ""
    val filterObs = ArrayList<Filter>()
    val filter =
        Filter {
            name = "processor-info.supported-architecture"
            values = listOf("arm64")
        }

    filterObs.add(filter)
    val typesRequest =
        DescribeInstanceTypesRequest {
            filters = filterObs
            maxResults = 10
        }
    Ec2Client { region = "us-west-2" }.use { ec2 ->
        val response = ec2.describeInstanceTypes(typesRequest)
        response.instanceTypes?.forEach { type ->
            println("The memory information of this type is
${type.memoryInfo?.sizeInMib}")

```

```
        println("Maximum number of network cards is
${type.networkInfo?.maximumNetworkCards}")
        instanceType = type.instanceType.toString()
    }
    return instanceType
}
}

// Display the Description field that corresponds to the instance Id value.
suspend fun describeImageSc(instanceId: String): String? {
    val imagesRequest =
        DescribeImagesRequest {
            imageIds = listOf(instanceId)
        }

    Ec2Client { region = "us-west-2" }.use { ec2 ->
        val response = ec2.describeImages(imagesRequest)
        println("The description of the first image is
${response.images?.get(0)?.description}")
        println("The name of the first image is
${response.images?.get(0)?.name}")

        // Return the image Id value.
        return response.images?.get(0)?.imageId
    }
}

// Get the Id value of an instance with amzn2 in the name.
suspend fun getParaValuesSc(): String? {
    val parameterRequest =
        GetParametersByPathRequest {
            path = "/aws/service/ami-amazon-linux-latest"
        }

    SsmClient { region = "us-west-2" }.use { ssmClient ->
        val response = ssmClient.getParametersByPath(parameterRequest)
        response.parameters?.forEach { para ->
            println("The name of the para is: ${para.name}")
            println("The type of the para is: ${para.type}")
            println("")
            if (para.name?.let { filterName(it) } == true) {
                return para.value
            }
        }
    }
}
```

```
    }
    return ""
}

fun filterName(name: String): Boolean {
    val parts = name.split("/").toTypedArray()
    val myValue = parts[4]
    return myValue.contains("amzn2")
}

suspend fun describeSecurityGroupsSc(groupId: String) {
    val request =
        DescribeSecurityGroupsRequest {
            groupIds = listOf(groupId)
        }

    Ec2Client { region = "us-west-2" }.use { ec2 ->
        val response = ec2.describeSecurityGroups(request)
        for (group in response.securityGroups!!) {
            println("Found Security Group with id " + group.groupId.toString() +
                " and group VPC " + group.vpcId)
        }
    }
}

suspend fun createEC2SecurityGroupSc(
    groupNameVal: String?,
    groupDescVal: String?,
    vpcIdVal: String?,
    myIpAddress: String?,
): String? {
    val request =
        CreateSecurityGroupRequest {
            groupName = groupNameVal
            description = groupDescVal
            vpcId = vpcIdVal
        }

    Ec2Client { region = "us-west-2" }.use { ec2 ->
        val resp = ec2.createSecurityGroup(request)
        val ipRange =
            IpRange {
                cidrIp = "$myIpAddress/0"
            }
    }
}
```

```
    val ipPerm =
        IpPermission {
            ipProtocol = "tcp"
            toPort = 80
            fromPort = 80
            ipRanges = listOf(ipRange)
        }

    val ipPerm2 =
        IpPermission {
            ipProtocol = "tcp"
            toPort = 22
            fromPort = 22
            ipRanges = listOf(ipRange)
        }

    val authRequest =
        AuthorizeSecurityGroupIngressRequest {
            groupName = groupNameVal
            ipPermissions = listOf(ipPerm, ipPerm2)
        }
    ec2.authorizeSecurityGroupIngress(authRequest)
    println("Successfully added ingress policy to Security Group
$groupNameVal")
    return resp.groupId
}
}

suspend fun describeEC2KeysSc() {
    Ec2Client { region = "us-west-2" }.use { ec2 ->
        val response = ec2.describeKeyPairs(DescribeKeyPairsRequest {})
        response.keyPairs?.forEach { keyPair ->
            println("Found key pair with name ${keyPair.keyName} and fingerprint
${ keyPair.keyFingerprint}")
        }
    }
}

suspend fun createKeyPairSc(
    keyNameVal: String,
    fileNameVal: String,
) {
    val request =
```

```
        CreateKeyPairRequest {
            keyName = keyNameVal
        }

        Ec2Client { region = "us-west-2" }.use { ec2 ->
            val response = ec2.createKeyPair(request)
            val content = response.keyMaterial
            if (content != null) {
                File(fileNameVal).writeText(content)
            }
            println("Successfully created key pair named $keyNameVal")
        }
    }
}
```

- Untuk detail API, lihat topik berikut di referensi API SDK untuk Kotlin AWS .
 - [AllocateAddress](#)
 - [AssociateAddress](#)
 - [AuthorizeSecurityGroupIngress](#)
 - [CreateKeyPair](#)
 - [CreateSecurityGroup](#)
 - [DeleteKeyPair](#)
 - [DeleteSecurityGroup](#)
 - [DescribeImages](#)
 - [DescribeInstanceTypes](#)
 - [DescribeInstances](#)
 - [DescribeKeyPairs](#)
 - [DescribeSecurityGroups](#)
 - [DisassociateAddress](#)
 - [ReleaseAddress](#)
 - [RunInstances](#)
 - [StartInstances](#)
 - [StopInstances](#)
 - [TerminateInstances](#)
 - [UnmonitorInstances](#)

Python

SDK untuk Python (Boto3)

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankan di [Repositori Contoh AWS Kode](#).

Jalankan skenario interaktif di prompt perintah.

```
class Ec2InstanceScenario:
    """Runs an interactive scenario that shows how to get started using EC2
    instances."""

    def __init__(self, inst_wrapper, key_wrapper, sg_wrapper, eip_wrapper,
                 ssm_client):
        """
        :param inst_wrapper: An object that wraps instance actions.
        :param key_wrapper: An object that wraps key pair actions.
        :param sg_wrapper: An object that wraps security group actions.
        :param eip_wrapper: An object that wraps Elastic IP actions.
        :param ssm_client: A Boto3 AWS Systems Manager client.
        """
        self.inst_wrapper = inst_wrapper
        self.key_wrapper = key_wrapper
        self.sg_wrapper = sg_wrapper
        self.eip_wrapper = eip_wrapper
        self.ssm_client = ssm_client

    @demo_func
    def create_and_list_key_pairs(self):
        """
        1. Creates an RSA key pair and saves its private key data as a .pem file
        in secure
           temporary storage. The private key data is deleted after the example
        completes.
        2. Lists the first five key pairs for the current account.
        """
        print(
            "Let's create an RSA key pair that you can be use to securely connect
        to "
```

```

        "your EC2 instance."
    )
    key_name = q.ask("Enter a unique name for your key: ", q.non_empty)
    self.key_wrapper.create(key_name)
    print(
        f"Created a key pair {self.key_wrapper.key_pair.key_name} and saved
the "
        f"private key to {self.key_wrapper.key_file_path}.\n"
    )
    if q.ask("Do you want to list some of your key pairs? (y/n) ",
q.is_yesno):
        self.key_wrapper.list(5)

@demo_func
def create_security_group(self):
    """
    1. Creates a security group for the default VPC.
    2. Adds an inbound rule to allow SSH. The SSH rule allows only
        inbound traffic from the current computer's public IPv4 address.
    3. Displays information about the security group.

    This function uses 'http://checkip.amazonaws.com' to get the current
public IP
address of the computer that is running the example. This method works in
most
cases. However, depending on how your computer connects to the internet,
you
might have to manually add your public IP address to the security group
by using
the AWS Management Console.
    """
    print("Let's create a security group to manage access to your instance.")
    sg_name = q.ask("Enter a unique name for your security group: ",
q.non_empty)
    security_group = self.sg_wrapper.create(
        sg_name, "Security group for example: get started with instances."
    )
    print(
        f"Created security group {security_group.group_name} in your default
"
        f"VPC {security_group.vpc_id}.\n"
    )

    ip_response = urllib.request.urlopen("http://checkip.amazonaws.com")

```

```

current_ip_address = ip_response.read().decode("utf-8").strip()
print("Let's add a rule to allow SSH only from your current IP address.")
print(f"Your public IP address is {current_ip_address}.")
q.ask("Press Enter to add this rule to your security group.")
response = self.sg_wrapper.authorize_ingress(current_ip_address)
if response["Return"]:
    print("Security group rules updated.")
else:
    print("Couldn't update security group rules.")
self.sg_wrapper.describe()

@demo_func
def create_instance(self):
    """
    1. Gets a list of Amazon Linux 2 AMIs from AWS Systems Manager.
    Specifying the
        '/aws/service/ami-amazon-linux-latest' path returns only the latest
    AMIs.
    2. Gets and displays information about the available AMIs and lets you
    select one.
    3. Gets a list of instance types that are compatible with the selected
    AMI and
        lets you select one.
    4. Creates an instance with the previously created key pair and security
    group,
        and the selected AMI and instance type.
    5. Waits for the instance to be running and then displays its
    information.
    """
    ami_paginator = self.ssm_client.get_paginator("get_parameters_by_path")
    ami_options = []
    for page in ami_paginator.paginate(Path="/aws/service/ami-amazon-linux-
latest"):
        ami_options += page["Parameters"]
    amzn2_images = self.inst_wrapper.get_images(
        [opt["Value"] for opt in ami_options if "amzn2" in opt["Name"]]
    )
    print(
        "Let's create an instance from an Amazon Linux 2 AMI. Here are some
options:"
    )
    image_choice = q.choose(
        "Which one do you want to use? ", [opt.description for opt in
amzn2_images]

```

```
)
print("Great choice!\n")

print(
    f"Here are some instance types that support the "
    f"{amzn2_images[image_choice].architecture} architecture of the
image:"
)
inst_types = self.inst_wrapper.get_instance_types(
    amzn2_images[image_choice].architecture
)
inst_type_choice = q.choose(
    "Which one do you want to use? ", [it["InstanceType"] for it in
inst_types]
)
print("Another great choice.\n")

print("Creating your instance and waiting for it to start...")
self.inst_wrapper.create(
    amzn2_images[image_choice],
    inst_types[inst_type_choice]["InstanceType"],
    self.key_wrapper.key_pair,
    [self.sg_wrapper.security_group],
)
print(f"Your instance is ready:\n")
self.inst_wrapper.display()

print("You can use SSH to connect to your instance.")
print(
    "If the connection attempt times out, you might have to manually
update "
    "the SSH ingress rule for your IP address in the AWS Management
Console."
)
self._display_ssh_info()

def _display_ssh_info(self):
    """
    Displays an SSH connection string that can be used to connect to a
running
instance.
    """
    print("To connect, open another command prompt and run the following
command:")
```

```
    if self.eip_wrapper.elastic_ip is None:
        print(
            f"\tssh -i {self.key_wrapper.key_file_path} "
            f"ec2-user@{self.inst_wrapper.instance.public_ip_address}"
        )
    else:
        print(
            f"\tssh -i {self.key_wrapper.key_file_path} "
            f"ec2-user@{self.eip_wrapper.elastic_ip.public_ip}"
        )
    q.ask("Press Enter when you're ready to continue the demo.")

@demo_func
def associate_elastic_ip(self):
    """
    1. Allocates an Elastic IP address and associates it with the instance.
    2. Displays an SSH connection string that uses the Elastic IP address.
    """
    print(
        "You can allocate an Elastic IP address and associate it with your\n"
        "instance\n"
        "to keep a consistent IP address even when your instance restarts."
    )
    elastic_ip = self.eip_wrapper.allocate()
    print(f"Allocated static Elastic IP address: {elastic_ip.public_ip}.")
    self.eip_wrapper.associate(self.inst_wrapper.instance)
    print(f"Associated your Elastic IP with your instance.")
    print(
        "You can now use SSH to connect to your instance by using the Elastic\n"
        "IP."
    )
    self._display_ssh_info()

@demo_func
def stop_and_start_instance(self):
    """
    1. Stops the instance and waits for it to stop.
    2. Starts the instance and waits for it to start.
    3. Displays information about the instance.
    4. Displays an SSH connection string. When an Elastic IP address is
    associated
    with the instance, the IP address stays consistent when the instance
    stops
    and starts.
    """
```

```
"""
print("Let's stop and start your instance to see what changes.")
print("Stopping your instance and waiting until it's stopped...")
self.inst_wrapper.stop()
print("Your instance is stopped. Restarting...")
self.inst_wrapper.start()
print("Your instance is running.")
self.inst_wrapper.display()
if self.eip_wrapper.elastic_ip is None:
    print(
        "Every time your instance is restarted, its public IP address
changes."
    )
else:
    print(
        "Because you have associated an Elastic IP with your instance,
you can \n"
        "connect by using a consistent IP address after the instance
restarts."
    )
self._display_ssh_info()

@demo_func
def cleanup(self):
    """
    1. Disassociate and delete the previously created Elastic IP.
    2. Terminate the previously created instance.
    3. Delete the previously created security group.
    4. Delete the previously created key pair.
    """
    print("Let's clean everything up. This example created these resources:")
    print(f"\tElastic IP: {self.eip_wrapper.elastic_ip.allocation_id}")
    print(f"\tInstance: {self.inst_wrapper.instance.id}")
    print(f"\tSecurity group: {self.sg_wrapper.security_group.id}")
    print(f"\tKey pair: {self.key_wrapper.key_pair.name}")
    if q.ask("Ready to delete these resources? (y/n) ", q.is_yesno):
        self.eip_wrapper.disassociate()
        print("Disassociated the Elastic IP from the instance.")
        self.eip_wrapper.release()
        print("Released the Elastic IP.")
        print("Terminating the instance and waiting for it to terminate...")
        self.inst_wrapper.terminate()
        print("Instance terminated.")
        self.sg_wrapper.delete()
```

```
        print("Deleted security group.")
        self.key_wrapper.delete()
        print("Deleted key pair.")

    def run_scenario(self):
        logging.basicConfig(level=logging.INFO, format="%(levelname)s:
%(message)s")

        print("-" * 88)
        print(
            "Welcome to the Amazon Elastic Compute Cloud (Amazon EC2) get started
with instances demo."
        )
        print("-" * 88)

        self.create_and_list_key_pairs()
        self.create_security_group()
        self.create_instance()
        self.stop_and_start_instance()
        self.associate_elastic_ip()
        self.stop_and_start_instance()
        self.cleanup()

        print("\nThanks for watching!")
        print("-" * 88)

if __name__ == "__main__":
    try:
        scenario = Ec2InstanceScenario(
            InstanceWrapper.from_resource(),
            KeyPairWrapper.from_resource(),
            SecurityGroupWrapper.from_resource(),
            ElasticIpWrapper.from_resource(),
            boto3.client("ssm"),
        )
        scenario.run_scenario()
    except Exception:
        logging.exception("Something went wrong with the demo.")
```

Tentukan kelas yang membungkus aksi pasangan kunci.

```
class KeyPairWrapper:
    """Encapsulates Amazon Elastic Compute Cloud (Amazon EC2) key pair
    actions."""

    def __init__(self, ec2_resource, key_file_dir, key_pair=None):
        """
        :param ec2_resource: A Boto3 Amazon EC2 resource. This high-level
        resource
                               is used to create additional high-level objects
                               that wrap low-level Amazon EC2 service actions.
        :param key_file_dir: The folder where the private key information is
        stored.
                               This should be a secure folder.
        :param key_pair: A Boto3 KeyPair object. This is a high-level object that
        wraps key pair actions.
        """
        self.ec2_resource = ec2_resource
        self.key_pair = key_pair
        self.key_file_path = None
        self.key_file_dir = key_file_dir

    @classmethod
    def from_resource(cls):
        ec2_resource = boto3.resource("ec2")
        return cls(ec2_resource, tempfile.TemporaryDirectory())

    def create(self, key_name):
        """
        Creates a key pair that can be used to securely connect to an EC2
        instance.
        The returned key pair contains private key information that cannot be
        retrieved
        again. The private key data is stored as a .pem file.

        :param key_name: The name of the key pair to create.
        :return: A Boto3 KeyPair object that represents the newly created key
        pair.
        """
        try:
            self.key_pair = self.ec2_resource.create_key_pair(KeyName=key_name)
            self.key_file_path = os.path.join(
                self.key_file_dir.name, f"{self.key_pair.name}.pem"
```



```
        )
        with open(self.key_file_path, "w") as key_file:
            key_file.write(self.key_pair.key_material)
    except ClientError as err:
        logger.error(
            "Couldn't create key %s. Here's why: %s: %s",
            key_name,
            err.response["Error"]["Code"],
            err.response["Error"]["Message"],
        )
        raise
    else:
        return self.key_pair

def list(self, limit):
    """
    Displays a list of key pairs for the current account.

    :param limit: The maximum number of key pairs to list.
    """
    try:
        for kp in self.ec2_resource.key_pairs.limit(limit):
            print(f"Found {kp.key_type} key {kp.name} with fingerprint:")
            print(f"\t{kp.key_fingerprint}")
    except ClientError as err:
        logger.error(
            "Couldn't list key pairs. Here's why: %s: %s",
            err.response["Error"]["Code"],
            err.response["Error"]["Message"],
        )
        raise

def delete(self):
    """
    Deletes a key pair.
    """
    if self.key_pair is None:
        logger.info("No key pair to delete.")
        return

    key_name = self.key_pair.name
    try:
```

```

        self.key_pair.delete()
        self.key_pair = None
    except ClientError as err:
        logger.error(
            "Couldn't delete key %s. Here's why: %s : %s",
            key_name,
            err.response["Error"]["Code"],
            err.response["Error"]["Message"],
        )
        raise

```

Menentukan kelas yang menggabungkan tindakan grup keamanan.

```

class SecurityGroupWrapper:
    """Encapsulates Amazon Elastic Compute Cloud (Amazon EC2) security group
    actions."""

    def __init__(self, ec2_resource, security_group=None):
        """
        :param ec2_resource: A Boto3 Amazon EC2 resource. This high-level
        resource
                                is used to create additional high-level objects
                                that wrap low-level Amazon EC2 service actions.
        :param security_group: A Boto3 SecurityGroup object. This is a high-level
        object
                                that wraps security group actions.
        """
        self.ec2_resource = ec2_resource
        self.security_group = security_group

    @classmethod
    def from_resource(cls):
        ec2_resource = boto3.resource("ec2")
        return cls(ec2_resource)

    def create(self, group_name, group_description):
        """

```

```
Creates a security group in the default virtual private cloud (VPC) of
the
current account.

:param group_name: The name of the security group to create.
:param group_description: The description of the security group to
create.
:return: A Boto3 SecurityGroup object that represents the newly created
security group.
"""
try:
    self.security_group = self.ec2_resource.create_security_group(
        GroupName=group_name, Description=group_description
    )
except ClientError as err:
    logger.error(
        "Couldn't create security group %s. Here's why: %s: %s",
        group_name,
        err.response["Error"]["Code"],
        err.response["Error"]["Message"],
    )
    raise
else:
    return self.security_group

def authorize_ingress(self, ssh_ingress_ip):
    """
    Adds a rule to the security group to allow access to SSH.

    :param ssh_ingress_ip: The IP address that is granted inbound access to
connect
                           to port 22 over TCP, used for SSH.
    :return: The response to the authorization request. The 'Return' field of
the
           response indicates whether the request succeeded or failed.
    """
    if self.security_group is None:
        logger.info("No security group to update.")
        return

    try:
        ip_permissions = [
            {
```

```
        # SSH ingress open to only the specified IP address.
        "IpProtocol": "tcp",
        "FromPort": 22,
        "ToPort": 22,
        "IpRanges": [{"CidrIp": f"{ssh_ingress_ip}/32"}],
    }
]
response = self.security_group.authorize_ingress(
    IpPermissions=ip_permissions
)
except ClientError as err:
    logger.error(
        "Couldn't authorize inbound rules for %s. Here's why: %s: %s",
        self.security_group.id,
        err.response["Error"]["Code"],
        err.response["Error"]["Message"],
    )
    raise
else:
    return response

def describe(self):
    """
    Displays information about the security group.
    """
    if self.security_group is None:
        logger.info("No security group to describe.")
        return

    try:
        print(f"Security group: {self.security_group.group_name}")
        print(f"\tID: {self.security_group.id}")
        print(f"\tVPC: {self.security_group.vpc_id}")
        if self.security_group.ip_permissions:
            print(f"Inbound permissions:")
            pp(self.security_group.ip_permissions)
    except ClientError as err:
        logger.error(
            "Couldn't get data for security group %s. Here's why: %s: %s",
            self.security_group.id,
            err.response["Error"]["Code"],
            err.response["Error"]["Message"],
        )
)
```

```
        raise

def delete(self):
    """
    Deletes the security group.
    """
    if self.security_group is None:
        logger.info("No security group to delete.")
        return

    group_id = self.security_group.id
    try:
        self.security_group.delete()
    except ClientError as err:
        logger.error(
            "Couldn't delete security group %s. Here's why: %s: %s",
            group_id,
            err.response["Error"]["Code"],
            err.response["Error"]["Message"],
        )
        raise
```

Menentukan kelas yang menggabungkan tindakan instans.

```
class InstanceWrapper:
    """Encapsulates Amazon Elastic Compute Cloud (Amazon EC2) instance
    actions."""

    def __init__(self, ec2_resource, instance=None):
        """
        :param ec2_resource: A Boto3 Amazon EC2 resource. This high-level
        resource
                               is used to create additional high-level objects
                               that wrap low-level Amazon EC2 service actions.
        :param instance: A Boto3 Instance object. This is a high-level object
        that
                               wraps instance actions.
        """
```

```

        self.ec2_resource = ec2_resource
        self.instance = instance

    @classmethod
    def from_resource(cls):
        ec2_resource = boto3.resource("ec2")
        return cls(ec2_resource)

    def create(self, image, instance_type, key_pair, security_groups=None):
        """
        Creates a new EC2 instance. The instance starts immediately after
        it is created.

        The instance is created in the default VPC of the current account.

        :param image: A Boto3 Image object that represents an Amazon Machine
        Image (AMI)
            that defines attributes of the instance that is created.
        The AMI
            defines things like the kind of operating system and the
        type of
            storage used by the instance.
        :param instance_type: The type of instance to create, such as 't2.micro'.
            The instance type defines things like the number of
        CPUs and
            the amount of memory.
        :param key_pair: A Boto3 KeyPair or KeyPairInfo object that represents
        the key
            pair that is used to secure connections to the instance.
        :param security_groups: A list of Boto3 SecurityGroup objects that
        represents the
            security groups that are used to grant access to
        the
            instance. When no security groups are specified,
        the
            default security group of the VPC is used.
        :return: A Boto3 Instance object that represents the newly created
        instance.
        """
        try:
            instance_params = {
                "ImageId": image.id,
                "InstanceType": instance_type,

```

```
        "KeyName": key_pair.name,
    }
    if security_groups is not None:
        instance_params["SecurityGroupIds"] = [sg.id for sg in
security_groups]
    self.instance = self.ec2_resource.create_instances(
        **instance_params, MinCount=1, MaxCount=1
    )[0]
    self.instance.wait_until_running()
except ClientError as err:
    logging.error(
        "Couldn't create instance with image %s, instance type %s, and
key %s. "
        "Here's why: %s: %s",
        image.id,
        instance_type,
        key_pair.name,
        err.response["Error"]["Code"],
        err.response["Error"]["Message"],
    )
    raise
else:
    return self.instance

def display(self, indent=1):
    """
    Displays information about an instance.

    :param indent: The visual indent to apply to the output.
    """
    if self.instance is None:
        logger.info("No instance to display.")
        return

    try:
        self.instance.load()
        ind = "\t" * indent
        print(f"{ind}ID: {self.instance.id}")
        print(f"{ind}Image ID: {self.instance.image_id}")
        print(f"{ind}Instance type: {self.instance.instance_type}")
        print(f"{ind}Key name: {self.instance.key_name}")
        print(f"{ind}VPC ID: {self.instance.vpc_id}")
        print(f"{ind}Public IP: {self.instance.public_ip_address}")
```

```
        print(f"{ind}State: {self.instance.state['Name']}")
    except ClientError as err:
        logger.error(
            "Couldn't display your instance. Here's why: %s: %s",
            err.response["Error"]["Code"],
            err.response["Error"]["Message"],
        )
        raise

def terminate(self):
    """
    Terminates an instance and waits for it to be in a terminated state.
    """
    if self.instance is None:
        logger.info("No instance to terminate.")
        return

    instance_id = self.instance.id
    try:
        self.instance.terminate()
        self.instance.wait_until_terminated()
        self.instance = None
    except ClientError as err:
        logging.error(
            "Couldn't terminate instance %s. Here's why: %s: %s",
            instance_id,
            err.response["Error"]["Code"],
            err.response["Error"]["Message"],
        )
        raise

def start(self):
    """
    Starts an instance and waits for it to be in a running state.

    :return: The response to the start request.
    """
    if self.instance is None:
        logger.info("No instance to start.")
        return

    try:
```



```
        response = self.instance.start()
        self.instance.wait_until_running()
    except ClientError as err:
        logger.error(
            "Couldn't start instance %s. Here's why: %s: %s",
            self.instance.id,
            err.response["Error"]["Code"],
            err.response["Error"]["Message"],
        )
        raise
    else:
        return response

def stop(self):
    """
    Stops an instance and waits for it to be in a stopped state.

    :return: The response to the stop request.
    """
    if self.instance is None:
        logger.info("No instance to stop.")
        return

    try:
        response = self.instance.stop()
        self.instance.wait_until_stopped()
    except ClientError as err:
        logger.error(
            "Couldn't stop instance %s. Here's why: %s: %s",
            self.instance.id,
            err.response["Error"]["Code"],
            err.response["Error"]["Message"],
        )
        raise
    else:
        return response

def get_images(self, image_ids):
    """
    Gets information about Amazon Machine Images (AMIs) from a list of AMI
    IDs.
```

```
:param image_ids: The list of AMIs to look up.
:return: A list of Boto3 Image objects that represent the requested AMIs.
"""
try:
    images = list(self.ec2_resource.images.filter(ImageIds=image_ids))
except ClientError as err:
    logger.error(
        "Couldn't get images. Here's why: %s: %s",
        err.response["Error"]["Code"],
        err.response["Error"]["Message"],
    )
    raise
else:
    return images

def get_instance_types(self, architecture):
    """
    Gets instance types that support the specified architecture and are
    designated
    as either 'micro' or 'small'. When an instance is created, the instance
    type
    you specify must support the architecture of the AMI you use.

    :param architecture: The kind of architecture the instance types must
    support,
                        such as 'x86_64'.
    :return: A list of instance types that support the specified architecture
    and are either 'micro' or 'small'.
    """
    try:
        inst_types = []
        it_paginator = self.ec2_resource.meta.client.get_paginator(
            "describe_instance_types"
        )
        for page in it_paginator.paginate(
            Filters=[
                {
                    "Name": "processor-info.supported-architecture",
                    "Values": [architecture],
                },
                {"Name": "instance-type", "Values": ["*.micro", "*.small"]},
            ]
        ):

```

```

        inst_types += page["InstanceTypes"]
    except ClientError as err:
        logger.error(
            "Couldn't get instance types. Here's why: %s: %s",
            err.response["Error"]["Code"],
            err.response["Error"]["Message"],
        )
        raise
    else:
        return inst_types

```

Menentukan kelas yang menggabungkan tindakan IP Elastis.

```

class ElasticIpWrapper:
    """Encapsulates Amazon Elastic Compute Cloud (Amazon EC2) Elastic IP address
    actions."""

    def __init__(self, ec2_resource, elastic_ip=None):
        """
        :param ec2_resource: A Boto3 Amazon EC2 resource. This high-level
        resource
                                is used to create additional high-level objects
                                that wrap low-level Amazon EC2 service actions.
        :param elastic_ip: A Boto3 VpcAddress object. This is a high-level object
        that
                                wraps Elastic IP actions.
        """
        self.ec2_resource = ec2_resource
        self.elastic_ip = elastic_ip

    @classmethod
    def from_resource(cls):
        ec2_resource = boto3.resource("ec2")
        return cls(ec2_resource)

    def allocate(self):
        """
        Allocates an Elastic IP address that can be associated with an Amazon EC2

```

```

        instance. By using an Elastic IP address, you can keep the public IP
address
        constant even when you restart the associated instance.

        :return: The newly created Elastic IP object. By default, the address is
not
                associated with any instance.
        """
        try:
            response =
self.ec2_resource.meta.client.allocate_address(Domain="vpc")
            self.elastic_ip =
self.ec2_resource.VpcAddress(response["AllocationId"])
        except ClientError as err:
            logger.error(
                "Couldn't allocate Elastic IP. Here's why: %s: %s",
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
        else:
            return self.elastic_ip

    def associate(self, instance):
        """
        Associates an Elastic IP address with an instance. When this association
is
        created, the Elastic IP's public IP address is immediately used as the
public
        IP address of the associated instance.

        :param instance: A Boto3 Instance object. This is a high-level object
that wraps
                Amazon EC2 instance actions.
        :return: A response that contains the ID of the association.
        """
        if self.elastic_ip is None:
            logger.info("No Elastic IP to associate.")
            return

        try:
            response = self.elastic_ip.associate(InstanceId=instance.id)
        except ClientError as err:

```

```
        logger.error(
            "Couldn't associate Elastic IP %s with instance %s. Here's why:
%s: %s",
            self.elastic_ip.allocation_id,
            instance.id,
            err.response["Error"]["Code"],
            err.response["Error"]["Message"],
        )
        raise
    return response

def disassociate(self):
    """
    Removes an association between an Elastic IP address and an instance.
    When the
    association is removed, the instance is assigned a new public IP address.
    """
    if self.elastic_ip is None:
        logger.info("No Elastic IP to disassociate.")
        return

    try:
        self.elastic_ip.association.delete()
    except ClientError as err:
        logger.error(
            "Couldn't disassociate Elastic IP %s from its instance. Here's
why: %s: %s",
            self.elastic_ip.allocation_id,
            err.response["Error"]["Code"],
            err.response["Error"]["Message"],
        )
        raise

def release(self):
    """
    Releases an Elastic IP address. After the Elastic IP address is released,
    it can no longer be used.
    """
    if self.elastic_ip is None:
        logger.info("No Elastic IP to release.")
        return
```

```
try:
    self.elastic_ip.release()
except ClientError as err:
    logger.error(
        "Couldn't release Elastic IP address %s. Here's why: %s: %s",
        self.elastic_ip.allocation_id,
        err.response["Error"]["Code"],
        err.response["Error"]["Message"],
    )
    raise
```

- Untuk detail API, lihat topik berikut di Referensi API AWS SDK for Python (Boto3).
 - [AllocateAddress](#)
 - [AssociateAddress](#)
 - [AuthorizeSecurityGroupIngress](#)
 - [CreateKeyPair](#)
 - [CreateSecurityGroup](#)
 - [DeleteKeyPair](#)
 - [DeleteSecurityGroup](#)
 - [DescribeImages](#)
 - [DescribeInstanceTypes](#)
 - [DescribeInstances](#)
 - [DescribeKeyPairs](#)
 - [DescribeSecurityGroups](#)
 - [DisassociateAddress](#)
 - [ReleaseAddress](#)
 - [RunInstances](#)
 - [StartInstances](#)
 - [StopInstances](#)
 - [TerminateInstances](#)
 - [UnmonitorInstances](#)

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Membuat sumber daya Amazon EC2 menggunakan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Pantau permintaan API Amazon EC2 menggunakan Amazon CloudWatch

Anda dapat memantau permintaan API Amazon EC2 menggunakan Amazon CloudWatch, yang mengumpulkan data mentah dan memprosesnya menjadi metrik yang dapat dibaca, mendekati waktu nyata. Metrik ini menyediakan cara sederhana untuk melacak penggunaan dan hasil operasi Amazon EC2 API dari waktu ke waktu. Informasi ini memberi Anda perspektif yang lebih baik tentang kinerja aplikasi web Anda, dan memungkinkan Anda mengidentifikasi dan mendiagnosis berbagai masalah. Anda juga dapat menyetel alarm yang mengawasi ambang batas tertentu, dan mengirim pemberitahuan atau mengambil tindakan tertentu saat ambang batas tersebut terpenuhi.

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

Important

Metrik API Amazon EC2 adalah fitur keikutsertaan. Anda harus meminta akses ke fitur ini. Untuk informasi selengkapnya, lihat [the section called “Aktifkan metrik API Amazon EC2”](#).

Daftar Isi

- [Aktifkan metrik API Amazon EC2](#)
- [Metrik dan dimensi API Amazon EC2](#)
- [Retensi data metrik](#)
- [Permintaan pemantauan yang dibuat atas nama Anda](#)
- [Penagihan](#)
- [Bekerja dengan Amazon CloudWatch](#)

Aktifkan metrik API Amazon EC2

Gunakan prosedur berikut untuk meminta akses ke fitur ini untuk Anda Akun AWS.

Untuk meminta akses ke fitur ini

1. [AWS Support Pusat](#) Terbuka.
2. Pilih Buat kasus.

3. Pilih Akun dan penagihan.
4. Untuk Layanan, pilih Info Umum dan Memulai.
5. Untuk Kategori, pilih Menggunakan AWS & Layanan.
6. Pilih Langkah selanjutnya: Informasi tambahan.
7. Untuk Subjek, masukkan **Request access to Amazon EC2 API metrics**.
8. Untuk Deskripsi, masukkan **Please grant my account access to Amazon EC2 API metrics. Related page: <https://docs.aws.amazon.com/AWSEC2/latest/APIReference/monitor.html>**. Juga termasuk Wilayah di mana Anda memerlukan akses.
9. Pilih Langkah selanjutnya: Selesaikan sekarang atau hubungi kami.
10. Pada tab Hubungi kami, pilih bahasa kontak pilihan Anda dan metode kontak.
11. Pilih Kirim.

Metrik dan dimensi API Amazon EC2

Metrik

Metrik API Amazon EC2 terkandung dalam namespace. `AWS/EC2/API` Tabel berikut mencantumkan metrik yang tersedia untuk permintaan Amazon EC2 API.

Metrik	Deskripsi
<code>ClientErrors</code>	<p>Jumlah permintaan API gagal yang disebabkan oleh kesalahan klien.</p> <p>Kesalahan ini biasanya disebabkan oleh sesuatu yang dilakukan klien, seperti menentukan parameter yang salah atau tidak valid dalam permintaan, atau menggunakan tindakan atau sumber daya atas nama pengguna yang tidak memiliki izin untuk menggunakan tindakan atau sumber daya.</p> <p>Unit: Jumlah</p>
<code>RequestLimitExceeded</code>	<p>Berapa kali tingkat permintaan maksimum yang diizinkan oleh Amazon EC2 API telah terlampaui untuk akun Anda.</p>

Metrik	Deskripsi
	<p>Permintaan API Amazon EC2 dibatasi untuk membantu menjaga kinerja layanan. Jika permintaan Anda telah dibatasi, Anda mendapatkan kesalahan. <code>Client.RequestLimitExceeded</code></p> <p>Unit: Jumlah</p>
<code>ServerErrors</code>	<p>Jumlah permintaan API gagal yang disebabkan oleh kesalahan server internal.</p> <p>Kesalahan ini biasanya disebabkan oleh kesalahan AWS sisi server, pengecualian, atau kegagalan.</p> <p>Unit: Jumlah</p>
<code>SuccessfulCalls</code>	<p>Jumlah permintaan API yang berhasil.</p> <p>Unit: Jumlah</p>

Dimensi

Data metrik Amazon EC2 dapat difilter di semua tindakan API EC2. Untuk informasi selengkapnya tentang dimensi, lihat [CloudWatch Konsep Amazon](#).

Retensi data metrik

Metrik API Amazon EC2 dikirim ke CloudWatch interval 1 menit. CloudWatch mempertahankan data metrik sebagai berikut:

- Titik data dengan periode 60 detik (1 menit) tersedia selama 15 hari.
- Titik data dengan jangka waktu 300 detik (5 menit) tersedia selama 63 hari.
- Poin data dengan jangka waktu 3600 detik (1 jam) tersedia selama 455 hari (15 bulan).

Permintaan pemantauan yang dibuat atas nama Anda

Permintaan API yang dibuat oleh AWS layanan atas nama Anda, seperti permintaan yang dibuat oleh peran terkait layanan, tidak diperhitungkan dalam batas pembatasan API Anda dan permintaan tersebut tidak mengirimkan metrik ke Amazon untuk akun Anda. CloudWatch Permintaan ini tidak dapat dipantau menggunakan CloudWatch.

Permintaan API yang dibuat atas nama Anda oleh penyedia layanan pihak ketiga dihitung terhadap batas pembatasan API Anda dan mereka mengirim metrik ke Amazon CloudWatch untuk akun Anda. Permintaan ini dapat dipantau menggunakan CloudWatch.

Penagihan

CloudWatch Harga dan biaya standar berlaku. Tidak ada biaya tambahan yang dikenakan untuk menggunakan metrik Amazon EC2 API. Untuk informasi selengkapnya, lihat [CloudWatch Harga Amazon](#).

Bekerja dengan Amazon CloudWatch

Daftar Isi

- [Melihat CloudWatch metrik](#)
- [Membuat CloudWatch alarm](#)

Melihat CloudWatch metrik

Gunakan prosedur berikut untuk melihat metrik Amazon EC2 API.

Prasyarat

Anda harus mengaktifkan akses ke metrik API Amazon EC2 untuk akun Anda. Untuk informasi selengkapnya, lihat [the section called “Aktifkan metrik API Amazon EC2”](#).

Untuk melihat metrik API Amazon EC2 menggunakan konsol

1. Buka CloudWatch konsol di <https://console.aws.amazon.com/cloudwatch/>.
2. Di panel navigasi, pilih Metrik, Semua metrik.
3. Pada tab Browse, pilih namespace metrik EC2/API.

4. Untuk melihat metrik, pilih dimensi metrik.

Untuk melihat metrik API Amazon EC2 menggunakan baris perintah

Gunakan salah satu perintah berikut:

- [daftar-metrik](#) ()AWS CLI

```
aws cloudwatch list-metrics --namespace "AWS/EC2/API"
```

- [Dapatkan-CW](#) () [MetricList](#)AWS Tools for Windows PowerShell

```
Get-CWMetricList -Namespace "AWS/EC2/API"
```

Membuat CloudWatch alarm

Anda dapat membuat CloudWatch alarm yang mengirimkan pesan Amazon SNS saat alarm berubah status. Alarm mengawasi metrik tunggal selama periode waktu yang Anda tentukan. Ini mengirimkan pemberitahuan ke topik SNS berdasarkan nilai metrik relatif terhadap ambang batas tertentu selama sejumlah periode waktu.

Misalnya, Anda dapat membuat alarm yang memantau jumlah permintaan DescribeInstances API yang gagal karena kesalahan sisi server. Alarm berikut mengirimkan pemberitahuan email ketika jumlah kegagalan permintaan DescribeInstances API mencapai ambang 10 kesalahan sisi server selama periode 5 menit.

Prasyarat

Anda harus mengaktifkan akses ke metrik Amazon EC2 API untuk akun Anda. Untuk informasi selengkapnya, lihat [the section called "Aktifkan metrik API Amazon EC2"](#).

Untuk membuat alarm untuk kesalahan server permintaan Amazon EC2 DescribeInstances API

1. Buka CloudWatch konsol di <https://console.aws.amazon.com/cloudwatch/>.
2. Pada panel navigasi, pilih Alarm, Semua alarm.
3. Pilih Buat alarm.
4. Pilih Pilih metrik, dan tentukan yang berikut ini:
 - a. Pilih EC2/API.

- b. Pilih Metrik Per-Tindakan.
 - c. Pilih kotak centang di sebelah DescribeInstances yang ada di baris yang sama dengan nama ServerErrors metrik.
 - d. Pilih Pilih Metrik.
5. Halaman Tentukan metrik dan ketentuan ditampilkan, di sana ditampilkan grafik dan informasi lain tentang metrik dan statistik yang Anda pilih.
- a. Di bawah Metrik, tentukan yang berikut ini:
 - i. Untuk Statistik pilih Jumlah.
 - ii. Untuk Periode, verifikasi bahwa 5 menit dipilih.
 - b. Pada Ketentuan, tentukan hal-hal berikut:
 - i. Untuk Jenis ambang batas, pilih Statis.
 - ii. Untuk Kapan pun ServerErrors ada, pilih Greater/Equal \geq .
 - iii. Untuk dari... , masukkan 10.
 - c. Pilih Selanjutnya.
6. Halaman Konfigurasi tindakan muncul.
- Di bawah Pemberitahuan, tentukan yang berikut ini:
 - i. Untuk pemicu status Alam, pilih In alarm.
 - ii. Untuk Pilih topik SNS, pilih Pilih topik SNS yang ada atau Buat topik baru, dan lengkapi bidang yang diperlukan untuk pemberitahuan.
 - iii. Pilih Selanjutnya.
7. Halaman Tambahkan nama dan deskripsi muncul.
- a. Untuk nama Alarm, masukkan nama untuk alarm Anda. Nama harus menggunakan karakter ASCII saja.
 - b. Untuk deskripsi Alarm, masukkan deskripsi opsional untuk alarm Anda.
 - c. Pilih Selanjutnya.
8. Halaman Pratinjau dan buat muncul. Verifikasi bahwa informasinya benar, lalu pilih Buat alarm.

Untuk informasi selengkapnya, lihat [Menggunakan CloudWatch alarm Amazon](#) di Panduan CloudWatch Pengguna Amazon.

Terjemahan disediakan oleh mesin penerjemah. Jika konten terjemahan yang diberikan bertentangan dengan versi bahasa Inggris aslinya, utamakan versi bahasa Inggris.