



Panduan Pengguna

AWS CodeBuild



Versi API 2016-10-06

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

AWS CodeBuild: Panduan Pengguna

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

Merek dagang dan tampilan dagang Amazon tidak boleh digunakan sehubungan dengan produk atau layanan apa pun yang bukan milik Amazon, dengan cara apa pun yang dapat menyebabkan kebingungan di antara 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 berafiliasi, terkait dengan, atau disponsori oleh Amazon, atau mungkin tidak.

Table of Contents

Apa itu AWS CodeBuild?	1
.....	1
Cara menjalankan CodeBuild	1
Harga untuk CodeBuild	3
Bagaimana saya memulai CodeBuild?	3
Konsep	3
Cara kerja CodeBuild	3
Langkah selanjutnya	5
Mulai	6
Memulai menggunakan konsol	6
Langkah-langkah	7
Langkah 1: Buat kode sumber	7
Langkah 2: Buat file buildspec	10
Langkah 3: Buat dua ember S3	12
Langkah 4: Unggah kode sumber dan file buildspec	13
Langkah 5: Buat proyek build	14
Langkah 6: Jalankan build	16
Langkah 7: Lihat informasi build yang dirangkum	17
Langkah 8: Lihat informasi build terperinci	18
Langkah 9: Dapatkan artefak keluaran build	19
Langkah 10: Hapus ember S3	20
Membungkus	21
Memulai menggunakan AWS CLI	21
Langkah-langkah	22
Langkah 1: Buat kode sumber	22
Langkah 2: Buat file buildspec	25
Langkah 3: Buat dua ember S3	27
Langkah 4: Unggah kode sumber dan file buildspec	28
Langkah 5: Buat proyek build	30
Langkah 6: Jalankan build	34
Langkah 7: Lihat informasi build yang dirangkum	35
Langkah 8: Lihat informasi build terperinci	38
Langkah 9: Dapatkan artefak keluaran build	41
Langkah 10: Hapus ember S3	42

Membungkus	43
Sampel	44
Gunakan sampel berbasis kasus	44
Sampel lintas layanan	45
Membangun sampel rencana	85
Buat laporan pengujian menggunakan AWS CLI sampel	90
Sampel Docker untuk CodeBuild	97
Host build output dalam bucket S3	111
Beberapa sumber input dan sampel artefak keluaran	115
Versi runtime dalam contoh file buildspec	119
Contoh versi sumber	128
Sampel repositori sumber pihak ketiga untuk CodeBuild	131
Gunakan versi semantik untuk memberi nama contoh artefak build	149
Sampel Windows	151
Menjalankan sampel	151
Struktur direktori	153
Berkas	153
Rencanakan pembangunan	172
Referensi Buildspec	174
Nama file Buildspec dan lokasi penyimpanan	174
Sintaks Buildspec	175
Contoh Buildspec	197
Versi Buildspec	200
Batch buildspec	200
GitHubReferensi buildspec pelari aksi	208
Membangun referensi lingkungan	210
Gambar Docker disediakan oleh CodeBuild	211
Bangun mode dan tipe komputasi lingkungan	230
Kerang dan perintah dalam membangun lingkungan	238
Variabel lingkungan di lingkungan build	239
Tugas latar belakang di lingkungan build	244
Bangun secara lokal	245
Prasyarat	245
Mengatur image build	245
JalankanCodeBuildagen	246
Menerima pemberitahuan untuk yang baruCodeBuildversi agen	247

Dukungan VPC	249
Kasus penggunaan	249
Mengizinkan akses Amazon VPC diCodeBuildmemproyeksikan	250
Praktik terbaik untuk VPC	251
Memecahkan masalah pengaturan VPC Anda	252
Batasan VPC	253
Gunakan titik akhir VPC	253
Sebelum Anda membuat titik akhir VPC	253
Membuat titik akhir VPC untukCodeBuild	254
Membuat kebijakan titik akhir VPC untukCodeBuild	254
AWS CloudFormationTemplat VPC	255
Menggunakan server proksi	261
Komponen yang diperlukan untuk menjalankan CodeBuild di server proxy	262
Jalankan CodeBuild di server proxy eksplisit	264
Jalankan CodeBuild di server proxy transparan	269
Menjalankan pengelola paket dan alat lain di server proxy	270
Bekerja dengan membangun proyek dan membangun	273
Bekerja dengan membangun proyek	273
Membuat proyek pembangunan	274
Membuat aturan notifikasi	312
Melihat daftar nama proyek build	315
Lihat detail proyek pembangunan	317
Membangun caching	320
Membangun pemicu	325
Webhook	332
Mengubah pengaturan proyek pembangunan	360
Menghapus proyek pembangunan	382
Bekerja dengan proyek bersama	383
Menandai proyek	388
Batch dibangun	394
GitHubPelari aksi	398
Proyek pembangunan publik	406
Bekerja dengan build	407
Jalankan build	409
Tampilkan detail build	420
Melihat daftar ID build	422

Melihat daftar ID build untuk proyek build	426
Hentikan build	429
Menghentikan pembuatan batch	431
Coba lagi membangun	433
Manajer Sesi	434
Menghapus build	439
Bekerja dengan AWS Lambda komputasi	441
Alat dan runtime mana yang akan disertakan dalam gambar docker lingkungan runtime yang dikuratori yang berjalan? AWS Lambda	441
Wilayah mana yang mendukung AWS Lambda komputasi? CodeBuild	441
Keterbatasan AWS Lambda komputasi	442
AWS Lambdamenghitung sampel	442
Menyebarkan fungsi Lambda menggunakan AWS SAM dengan CodeBuild Lambda Java ...	442
Buat aplikasi React satu halaman dengan CodeBuild Lambda Node.js	446
Perbarui konfigurasi fungsi Lambda dengan CodeBuild Lambda Python	449
Bekerja dengan kapasitas cadangan	454
Bagaimana cara memulai dengan armada kapasitas cadangan?	455
Praktik terbaik	455
Dapatkan saya berbagi armada kapasitas cadangan di beberapa CodeBuild proyek?	456
Wilayah mana yang mendukung armada kapasitas cadangan?	456
Properti armada kapasitas cadangan	456
Sampel kapasitas cadangan	459
Caching dengan sampel kapasitas cadangan	459
Keterbatasan armada kapasitas cadangan	461
Bekerja dengan pelaporan pengujian	462
Buat laporan pengujian	463
Bekerja dengan kelompok laporan	464
Membuat grup laporan	465
Memperbarui grup laporan	471
Tentukan file uji	474
Tentukan perintah uji	475
Laporkan penamaan grup	475
Menandai grup laporan	476
Bekerja dengan grup laporan bersama	482
Bekerja dengan laporan	488
Bekerja dengan izin laporan pengujian	489

Buat peran untuk laporan pengujian	489
Izin untuk operasi pelaporan pengujian	491
Contoh izin pelaporan uji	492
Lihat laporan pengujian	492
Melihat laporan pengujian untuk build	493
Melihat laporan pengujian untuk grup laporan	493
Melihat laporan pengujian di AWS akun Anda	493
Pelaporan pengujian dengan kerangka kerja pengujian	494
Melaporkan dengan Jasmine	494
Melaporkan dengan Jest	496
Melaporkan dengan pytest	498
Pelaporan dengan RSpec	499
Laporan cakupan kode	500
.....	500
Membuat laporan cakupan kode	500
Pencatatan dan pemantauan	502
Mencatat panggilan API AWS CodeBuild dengan AWS CloudTrail	502
AWS CodeBuildinformasi dalam CloudTrail	502
Memahami entri file log AWS CodeBuild	503
Memantau AWS CodeBuild	506
Metrik CloudWatch	506
Metrik pemanfaatan sumber daya CloudWatch	509
Dimensi CloudWatch	510
Alarm CloudWatch	511
Metrik CodeBuild	511
Metrik pemanfaatan sumber daya CodeBuild	514
Alarm CodeBuild	518
Keamanan	519
Perlindungan data	519
Enkripsi data	521
Manajemen kunci	522
Privasi lalu lintas	522
Pengelolaan identitas dan akses	523
Ikhtisar mengenai pengelolaan akses	523
Menggunakan kebijakan berbasis identitas	527
AWS CodeBuild referensi izin	556

Menggunakan tag untuk mengontrol akses ke AWS CodeBuild sumber daya	563
Menampilkan sumber daya di konsol	567
Validasi kepatuhan	567
Ketangguhan	568
Keamanan infrastruktur	569
Akses penyedia sumber	569
GitHub dan token akses Server GitHub Perusahaan	569
Kata sandi aplikasi Bitbucket	573
Pencegahan confused deputy lintas layanan	576
Topik lanjutan	578
Pengaturan lanjutan	578
Menambahkan izin CodeBuild akses ke grup atau pengguna IAM	579
Membuat peran CodeBuild layanan	585
Membuat kunci terkelola pelanggan	593
Instal dan konfigurasi AWS CLI	595
Referensi baris perintah	596
AWS Referensi SDK dan alat	598
Didukung AWS SDK dan alat untuk AWS CodeBuild	598
Tentukan titik akhir	599
Tentukan AWS CodeBuild endpoint (AWS CLI)	599
Tentukan AWS CodeBuild titik akhir (AWS SDK)	600
Gunakan CodePipeline dengan CodeBuild	602
Prasyarat	603
Buat pipeline (konsol)	605
Buat pipeline (AWS CLI)	609
Tindakan menambahkan	614
Tindakan menambahkan Tindakan menambahkan Tindakan menambahkan Tindakan menambahkan Tindakan menambahkan pengujian	618
Gunakan CodeBuild dengan Jenkins	621
Menyiapkan Jenkins	622
Menginstal plugin	622
Menggunakan plugin	622
Gunakan CodeBuild dengan Codecov	624
Mengintegrasikan Codecov ke proyek pembangunan	624
Aplikasi tanpa server	627
Sumber daya terkait	52

Memecahkan masalah	629
Apache Maven membangun artefak referensi dari repositori yang salah	630
Perintah build dijalankan sebagai root secara default	632
Build mungkin gagal ketika nama file memiliki non-A.S. Karakter bahasa Inggris	632
Build mungkin gagal saat mendapatkan parameter dari Amazon EC2 Parameter Store	633
Tidak dapat mengakses filter cabang di CodeBuild konsol	634
Tidak dapat melihat keberhasilan atau kegagalan bangunan	634
Status build tidak dilaporkan ke penyedia sumber	634
Tidak dapat menemukan dan memilih gambar dasar platform Windows Server Core 2019	635
Perintah sebelumnya dalam file buildspec tidak dikenali oleh perintah selanjutnya	635
Kesalahan: "Akses ditolak" saat mencoba mengunduh cache	636
Kesalahan: "BUILD_CONTAINER_UNABLE_TO_PULL_IMAGE" saat menggunakan gambar build khusus	636
Kesalahan: "Build container ditemukan mati sebelum menyelesaikan build. build container mati karena kehabisan memori, atau image Docker tidak didukung. ErrorCode: 500"	638
Kesalahan: "Tidak dapat terhubung ke daemon Docker" saat menjalankan build	638
Kesalahan: "CodeBuild tidak diizinkan untuk melakukan: sts:AssumeRole" saat membuat atau memperbarui proyek build	639
Kesalahan: "Kesalahan memanggil GetBucketAcl: Pemilik bucket telah berubah atau peran layanan tidak lagi memiliki izin untuk memanggil s3:GetBucketAcl"	640
Kesalahan: "Gagal mengunggah artefak: Arn tidak valid" saat menjalankan build	641
Kesalahan: "Klon Git gagal: Tidak dapat mengakses 'your-repository-URL': Masalah sertifikat SSL: Sertifikat yang ditandatangani sendiri"	641
Kesalahan: "Bucket yang Anda coba akses harus ditangani menggunakan titik akhir yang ditentukan" saat menjalankan build	642
Error: "Gambar build ini memerlukan pemilihan setidaknya satu versi runtime."	642
Kesalahan: "QUEUED: INSUFFICIENT_SUBNET" saat build dalam antrian build gagal	643
Kesalahan: "Tidak dapat mengunduh cache: RequestError: Permintaan kirim gagal disebabkan oleh: x509: Gagal memuat akar sistem dan tidak ada akar yang disediakan"	644
Kesalahan: "Tidak dapat mengunduh sertifikat dari S3. AccessDenied"	644
Kesalahan: "Tidak dapat menemukan kredensi"	644
RequestError kesalahan batas waktu saat berjalan CodeBuild di server proxy	646
Cangkang bourne (sh) harus ada dalam gambar build	647
Peringatan: "Melewatkan pemasangan runtime. pemilihan versi runtime tidak didukung oleh image build ini" saat menjalankan build	648
Kesalahan: "Tidak dapat memverifikasi JobWorker identitas"	648

Build gagal untuk memulai	648
Mengakses GitHub metadata dalam build yang di-cache secara lokal	649
AccessDenied: Pemilik bucket untuk grup laporan tidak cocok dengan pemilik bucket S3...	649
Kuota	650
Kuota layanan	650
Batas lainnya	655
Proyek Build	655
Bangunan	656
Hitung armada	656
Laporan	657
Tanda	658
Pemberitahuan pihak ketiga AWS CodeBuild untuk Windows	660
1) gambar Docker dasar—windowsservercore	660
2) gambar Docker berbasis jendela — choco	661
3) gambar Docker berbasis jendela — git --version 2.16.2	662
4) gambar Docker dasar jendela— --version 15.0.26320.2 microsoft-build-tools	662
5) gambar Docker berbasis jendela — nuget.commandline --versi 4.5.1	666
7) gambar Docker berbasis jendela — netfx-4.6.2-devpack	667
8) gambar Docker dasar jendela — visualfsharptools, v 4.0	668
9) gambar Docker dasar jendela— -4.6 netfx-pcl-reference-assemblies	669
10) gambar Docker dasar jendela—visualcppbuildtools v 14.0.25420.1	673
11) gambar Docker dasar jendela- 3-ondemand-package.cab microsoft-windows-netfx	677
12) gambar Docker berbasis jendela — dotnet-sdk	678
Riwayat dokumen	680
Pembaruan sebelumnya	697
AWSGlosarium	710
.....	dccxi

Apakah AWS CodeBuild itu?

AWS CodeBuild adalah layanan build yang dikelola sepenuhnya di cloud. CodeBuild mengkompilasi kode sumber Anda, menjalankan pengujian unit, dan menghasilkan artefak yang siap digunakan. CodeBuild menghilangkan kebutuhan untuk menyediakan, mengelola, dan menskalakan server build Anda sendiri. Ini menyediakan lingkungan bangunan yang dikemas untuk bahasa pemrograman populer dan membangun alat seperti Apache Maven, Gradle, dan banyak lagi. Anda juga dapat menyesuaikan lingkungan build CodeBuild untuk menggunakan alat build Anda sendiri. CodeBuild skala secara otomatis untuk memenuhi permintaan build puncak.

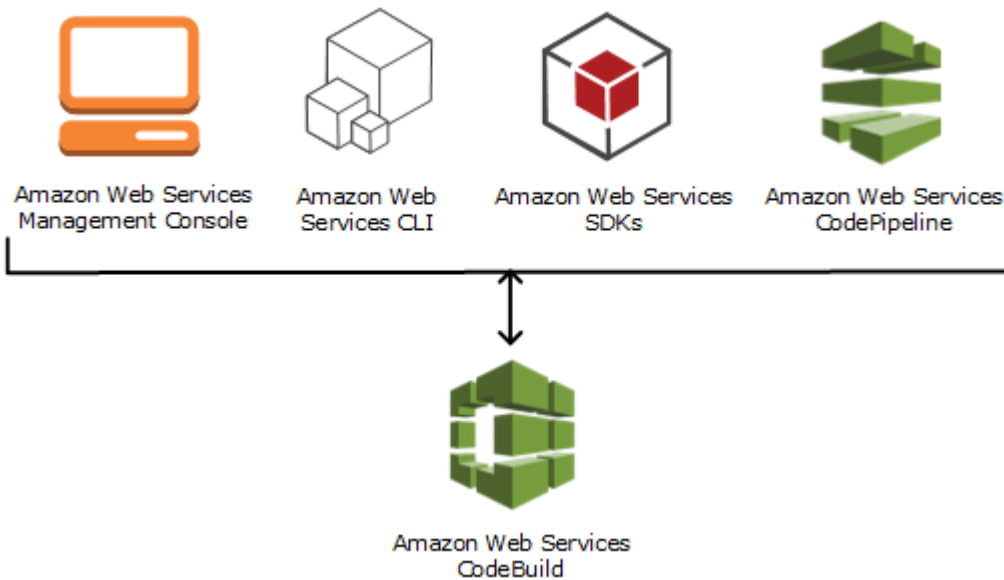
CodeBuild memberikan manfaat ini:

- Dikelola sepenuhnya — CodeBuild menghilangkan kebutuhan untuk mengatur, menambal, memperbarui, dan mengelola server build Anda sendiri.
- Sesuai permintaan — CodeBuild skala sesuai permintaan untuk memenuhi kebutuhan bangunan Anda. Anda hanya membayar untuk jumlah menit build yang Anda konsumsi.
- Out of the box - CodeBuild menyediakan lingkungan build yang telah dikonfigurasi untuk bahasa pemrograman paling populer. Yang perlu Anda lakukan adalah mengarahkan ke skrip build Anda untuk memulai build pertama Anda.

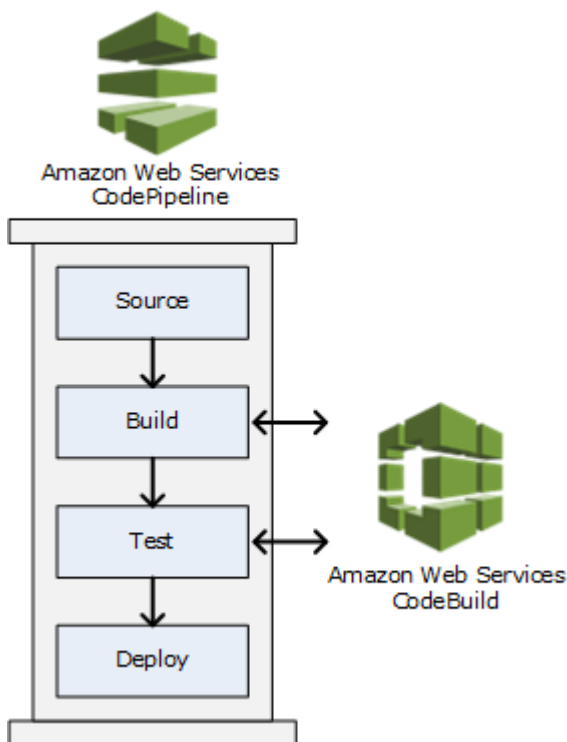
Untuk informasi selengkapnya, lihat [AWS CodeBuild](#).

Cara menjalankan CodeBuild

Anda dapat menggunakan AWS CodePipeline konsol AWS CodeBuild atau untuk menjalankan CodeBuild. Anda juga dapat mengotomatiskan proses CodeBuild dengan menggunakan AWS Command Line Interface (AWS CLI) atau AWS SDK.



Seperti yang ditunjukkan diagram berikut, Anda dapat menambahkan CodeBuild sebagai tindakan build atau pengujian ke tahap build atau pengujian pipeline di AWS CodePipeline. AWS CodePipeline adalah layanan pengiriman berkelanjutan yang dapat Anda gunakan untuk memodelkan, memvisualisasikan, dan mengotomatiskan langkah-langkah yang diperlukan untuk merilis kode Anda. Ini termasuk membangun kode Anda. Pipeline adalah konstruksi alur kerja yang menjelaskan bagaimana perubahan kode melalui proses rilis.



Untuk digunakan CodePipeline untuk membuat pipeline dan kemudian menambahkan tindakan CodeBuild build atau test, lihat [Gunakan CodePipeline dengan CodeBuild](#). Untuk informasi selengkapnya CodePipeline, lihat [Panduan AWS CodePipeline Pengguna](#).

CodeBuild Konsol juga menyediakan cara untuk mencari sumber daya Anda dengan cepat, seperti repositori, membangun proyek, aplikasi penerapan, dan saluran pipa. Pilih Buka sumber daya atau tekan / tombol, lalu masukkan nama sumber daya. Setiap kecocokan akan muncul di daftar. Pencarian peka huruf besar/kecil. Anda hanya melihat sumber daya yang izin untuk menampilkannya Anda memiliki. Untuk informasi selengkapnya, lihat [Menampilkan sumber daya di konsol](#).

Harga untuk CodeBuild

Untuk informasi, lihat [CodeBuild harga](#).

Bagaimana saya memulai CodeBuild?

Kami menyarankan agar Anda menyelesaikan langkah berikut:

1. Pelajari lebih lanjut CodeBuild dengan membaca informasi di [Konsep](#).
2. Bereksperimenlah dengan CodeBuild skenario contoh dengan mengikuti instruksi di [Memulai menggunakan konsol](#).
3. Gunakan CodeBuild dalam skenario Anda sendiri dengan mengikuti instruksi di [Rencanakan pembangunan](#).

Konsep AWS CodeBuild

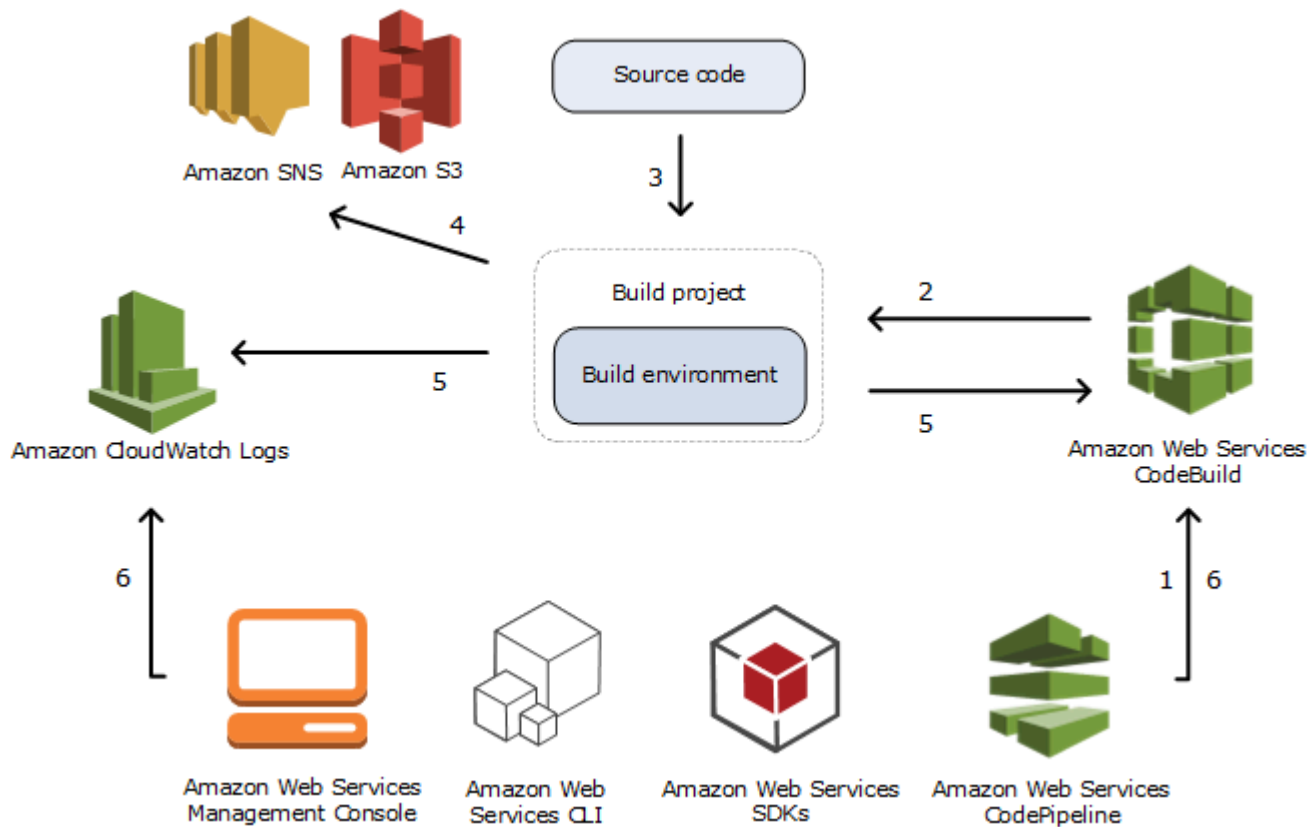
Konsep berikut penting untuk memahami cara kerja CodeBuild.

Topik

- [Cara kerja CodeBuild](#)
- [Langkah selanjutnya](#)

Cara kerja CodeBuild

Diagram berikut menunjukkan apa yang terjadi ketika Anda menjalankan build dengan CodeBuild:



1. Sebagai masukan, Anda harus menyediakan CodeBuild dengan proyek build. SEBUAH proyek build termasuk informasi tentang cara menjalankan build, termasuk di mana untuk mendapatkan kode sumber, yang membangun lingkungan untuk digunakan, yang membangun perintah untuk dijalankan, dan di mana untuk menyimpan output build. SEBUAH lingkungan build mewakili kombinasi sistem operasi, runtime bahasa pemrograman, dan alat yang digunakan CodeBuild untuk menjalankan build. Untuk informasi selengkapnya, lihat :

- [Membuat proyek pembangunan](#)
- [Membangun referensi lingkungan](#)

2. CodeBuild menggunakan proyek build untuk menciptakan lingkungan build.
3. CodeBuild mendownload kode sumber ke lingkungan build dan kemudian menggunakan spesifikasi build (buildspec), seperti yang didefinisikan dalam proyek build atau disertakan langsung dalam kode sumber. SEBUAH buildspec adalah rangkaian perintah build dan pengaturan terkait, dalam format YAML, yang digunakan CodeBuild untuk menjalankan build. Untuk informasi lain, lihat [Referensi Buildspec](#).

4. Jika ada output build, lingkungan build upload outputnya ke bucket S3. Lingkungan build juga dapat melakukan tugas yang Anda tetapkan di buildspec (misalnya, mengirim notifikasi build ke topik Amazon SNS). Sebagai contoh, lihat [Buat contoh notifikasi](#).
5. Saat build berjalan, lingkungan build mengirimkan informasi ke CodeBuild dan Amazon CloudWatch Logs.
6. Saat membangun berjalan, Anda dapat menggunakan AWS CodeBuild konsol, AWS CLI, atau AWS SDK untuk mendapatkan informasi build yang diringkas dari CodeBuild dan informasi build terperinci dari Amazon CloudWatch Logs. Jika Anda menggunakan AWS CodePipeline untuk menjalankan build, Anda bisa mendapatkan informasi build terbatas dari CodePipeline.

Langkah selanjutnya

Sekarang Anda tahu lebih banyak tentang AWS CodeBuild, kami merekomendasikan langkah-langkah selanjutnya:

1. percobaan dengan CodeBuild dalam skenario contoh dengan mengikuti petunjuk di [Memulai menggunakan konsol](#).
2. Gunakan CodeBuild dalam skenario Anda sendiri dengan mengikuti petunjuk di [Rencanakan pembangunan](#).

Memulai dengan CodeBuild

Dalam tutorial berikut, Anda menggunakan AWS CodeBuild untuk membangun kumpulan file input kode sumber sampel ke dalam versi kode sumber yang dapat disebarluaskan.

Kedua tutorial memiliki input dan hasil yang sama, tetapi satu menggunakan AWS CodeBuild konsol dan yang lainnya menggunakan AWS CLI.

Important

Kami tidak merekomendasikan Anda menggunakan AWS akun root untuk menyelesaikan tutorial ini.

Memulai dengan AWS CodeBuild menggunakan konsol

Dalam tutorial ini, Anda gunakan AWS CodeBuild untuk membangun kumpulan file input kode sumber sampel (membangun artefak input atau membangun input) ke dalam versi kode sumber yang dapat diterapkan (membangun artefak keluaran atau keluaran build). Secara khusus, Anda menginstruksikan CodeBuild untuk menggunakan Apache Maven, alat build umum, untuk membangun satu set file kelas Java ke dalam file Java Archive (JAR). Anda tidak perlu terbiasa dengan Apache Maven atau Java untuk menyelesaikan tutorial ini.

Anda dapat bekerja dengan CodeBuild melalui CodeBuild konsol, AWS CodePipeline, AWS CLI, atau AWS SDK. Tutorial ini menunjukkan cara menggunakan CodeBuild konsol. Untuk informasi tentang penggunaan CodePipeline, lihat [Gunakan CodePipeline dengan CodeBuild](#).

Important

Langkah-langkah dalam tutorial ini mengharuskan Anda untuk membuat sumber daya (misalnya, bucket S3) yang mungkin mengakibatkan biaya ke AWS akun Anda. Ini termasuk kemungkinan biaya untuk CodeBuild dan untuk AWS sumber daya dan tindakan yang terkait dengan Amazon S3, AWS KMS, dan CloudWatch Log. Untuk informasi selengkapnya, lihat [AWS CodeBuild harga](#), [harga Amazon S3](#), [AWS Key Management Service harga](#), dan [harga Amazon CloudWatch](#).

Langkah-langkah

- [Langkah 1: Buat kode sumber](#)
- [Langkah 2: Buat file buildspec](#)
- [Langkah 3: Buat dua ember S3](#)
- [Langkah 4: Unggah kode sumber dan file buildspec](#)
- [Langkah 5: Buat proyek build](#)
- [Langkah 6: Jalankan build](#)
- [Langkah 7: Lihat informasi build yang dirangkum](#)
- [Langkah 8: Lihat informasi build terperinci](#)
- [Langkah 9: Dapatkan artefak keluaran build](#)
- [Langkah 10: Hapus ember S3](#)
- [Membungkus](#)

Langkah 1: Buat kode sumber

(Bagian dari: [Memulai dengan AWS CodeBuild menggunakan konsol](#))

Pada langkah ini, Anda membuat kode sumber yang CodeBuild ingin Anda buat ke bucket keluaran. Kode sumber ini terdiri dari dua file kelas Java dan file Apache Maven Project Object Model (POM).

1. Di direktori kosong di komputer atau instance lokal Anda, buat struktur direktori ini.

```
(root directory name)
|-- src
    |-- main
    |   |-- java
    |-- test
        |-- java
```

2. Menggunakan editor teks pilihan Anda, buat file ini, beri nama `MessageUtil.java`, lalu simpan di `src/main/java` direktori.

```
public class MessageUtil {
    private String message;

    public MessageUtil(String message) {
```

```
    this.message = message;
}

public String printMessage() {
    System.out.println(message);
    return message;
}

public String salutationMessage() {
    message = "Hi!" + message;
    System.out.println(message);
    return message;
}
}
```

File kelas ini menciptakan sebagai output string karakter yang dilewatkan ke dalamnya.

MessageUtilKonstruktor menetapkan string karakter. printMessageMetode ini menciptakan output. Output salutationMessage metode Hi! diikuti oleh string karakter.

3. Buat file ini, beri nama `TestMessageUtil.java`, lalu simpan di `/src/test/java` direktori.

```
import org.junit.Test;
import org.junit.Ignore;
import static org.junit.Assert.assertEquals;

public class TestMessageUtil {

    String message = "Robert";
    MessageUtil messageUtil = new MessageUtil(message);

    @Test
    public void testPrintMessage() {
        System.out.println("Inside testPrintMessage()");
        assertEquals(message,messageUtil.printMessage());
    }

    @Test
    public void testSalutationMessage() {
        System.out.println("Inside testSalutationMessage()");
        message = "Hi!" + "Robert";
        assertEquals(message,messageUtil.salutationMessage());
    }
}
```

File kelas ini menetapkan message variabel di MessageUtil kelas untukRobert. Kemudian tes untuk melihat apakah message variabel berhasil diatur dengan memeriksa apakah string Robert dan Hi!Robert muncul dalam output.

4. Buat file ini, beri namapom.xml, lalu simpan di direktori root (tingkat atas).

```
<project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/
maven-v4_0_0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>org.example</groupId>
  <artifactId>messageUtil</artifactId>
  <version>1.0</version>
  <packaging>jar</packaging>
  <name>Message Utility Java Sample App</name>
  <dependencies>
    <dependency>
      <groupId>junit</groupId>
      <artifactId>junit</artifactId>
      <version>4.11</version>
      <scope>test</scope>
    </dependency>
  </dependencies>
  <build>
    <plugins>
      <plugin>
        <groupId>org.apache.maven.plugins</groupId>
        <artifactId>maven-compiler-plugin</artifactId>
        <version>3.8.0</version>
      </plugin>
    </plugins>
  </build>
</project>
```

Apache Maven menggunakan instruksi dalam file ini untuk mengonversi file MessageUtil.java dan menjadi TestMessageUtil.java file bernama messageUtil-1.0.jar dan kemudian menjalankan tes yang ditentukan.

Pada titik ini, struktur direktori Anda akan terlihat seperti ini.

```
(root directory name)
|-- pom.xml
`-- src
    |-- main
    |   |-- java
    |   |-- MessageUtil.java
    |-- test
    |   |-- java
    |   |-- TestMessageUtil.java
```

Langkah selanjutnya

[Langkah 2: Buat file buildspec](#)

Langkah 2: Buat file buildspec

(Langkah sebelumnya: [Langkah 1: Buat kode sumber](#))

Pada langkah ini, Anda membuat file spesifikasi build (spesifikasi build). Buildspec adalah kumpulan perintah build dan pengaturan terkait, dalam format YAML, yang CodeBuild digunakan untuk menjalankan build. Tanpa spesifikasi build, CodeBuild tidak dapat berhasil mengonversi input build Anda menjadi keluaran build atau menemukan artefak keluaran build di lingkungan build untuk diunggah ke bucket keluaran Anda.

Buat file ini, beri nama `buildspec.yml`, lalu simpan di direktori root (tingkat atas).

```
version: 0.2

phases:
  install:
    runtime-versions:
      java: corretto11
  pre_build:
    commands:
      - echo Nothing to do in the pre_build phase...
  build:
    commands:
      - echo Build started on `date`
      - mvn install
  post_build:
    commands:
```

```
- echo Build completed on `date`  
artifacts:  
  files:  
    - target/messageUtil-1.0.jar
```

Important

Karena deklarasi spesifikasi build harus YAMM yang valid, spasi dalam deklarasi spesifikasi build menjadi penting. Jika jumlah spasi dalam deklarasi spesifikasi build Anda tidak cocok dengan yang ini, build mungkin akan segera gagal. Anda dapat menggunakan validator YAMM untuk menguji apakah deklarasi spesifikasi build Anda adalah YAMG yang valid.

Note

Alih-alih menyertakan file spesifikasi build dalam kode sumber, Anda dapat mendeklarasikan perintah build secara terpisah saat membuat project build. Ini sangat membantu jika Anda ingin membangun kode sumber Anda dengan perintah build yang berbeda tanpa memperbarui repositori kode sumber Anda setiap kali. Untuk informasi selengkapnya, lihat [Sintaks Buildspec](#).

Dalam deklarasi spesifikasi build ini:

- `version` mewakili versi standar spesifikasi build yang digunakan. Deklarasi spesifikasi build ini menggunakan versi terbaru, `0.2`.
- `phase` mewakili fase build di mana Anda dapat menginstruksikan CodeBuild untuk menjalankan perintah. Fase build ini tercantum di sini sebagai `install`, `pre_build`, `build`, dan `post_build`. Anda tidak dapat mengubah ejaan nama fase build ini, dan Anda tidak dapat membuat lebih banyak nama fase build.

Dalam contoh ini, selama `build` fase, CodeBuild jalankan `mvn install` perintah. Perintah ini menginstruksikan Apache Maven untuk mengkompilasi, menguji, dan mengemas file kelas Java yang dikompilasi ke dalam artefak keluaran build. Untuk kelengkapan, beberapa `echo` perintah ditempatkan di setiap fase build dalam contoh ini. Ketika Anda melihat informasi build rinci nanti dalam tutorial ini, output dari `echo` perintah ini dapat membantu Anda lebih memahami bagaimana CodeBuild menjalankan perintah dan urutan apa. (Meskipun semua fase build disertakan dalam contoh ini, Anda tidak diharuskan menyertakan fase build jika Anda tidak berencana menjalankan

perintah apa pun selama fase tersebut.) Untuk setiap fase build, CodeBuild jalankan setiap perintah yang ditentukan, satu per satu, dalam urutan yang terdaftar, dari awal hingga akhir.

- `artifacts` mewakili kumpulan artefak keluaran build yang CodeBuild diunggah ke bucket keluaran. `files` mewakili file yang akan disertakan dalam output build. CodeBuild mengunggah `messageUtil-1.0.jar` file tunggal yang ditemukan di direktori `target` relatif di lingkungan build. Nama file `messageUtil-1.0.jar` dan nama direktori `target` didasarkan pada cara Apache Maven membuat dan menyimpan artefak keluaran build untuk contoh ini saja. Dalam build Anda sendiri, nama file dan direktori ini berbeda.

Lihat informasi yang lebih lengkap di [Referensi Buildspec](#).

Pada titik ini, struktur direktori Anda akan terlihat seperti ini.

```
(root directory name)
|-- pom.xml
|-- buildspec.yml
`-- src
    |-- main
    |   |-- java
    |       |-- MessageUtil.java
    |-- test
    |   |-- java
    |       |-- TestMessageUtil.java
```

Langkah selanjutnya

[Langkah 3: Buat dua ember S3](#)

Langkah 3: Buat dua ember S3

(Langkah sebelumnya: [Langkah 2: Buat file buildspec](#))

Meskipun Anda dapat menggunakan satu bucket untuk tutorial ini, dua bucket memudahkan untuk melihat dari mana input build berasal dan ke mana arah output build.

- Salah satu bucket ini (bucket input) menyimpan input build. Dalam tutorial ini, nama bucket input ini adalah `codebuild-region-ID-account-ID-input-bucket`, di mana *Region-ID* adalah *Region* bucket dan *Account-ID* adalah *ID* akun Anda. AWS AWS
- Bucket lainnya (ember keluaran) menyimpan output build. Dalam tutorial ini, nama bucket keluaran ini adalah `codebuild-region-ID-account-ID-output-bucket`.

Jika Anda memilih nama yang berbeda untuk ember ini, pastikan untuk menggunakannya di seluruh tutorial ini.

Kedua ember ini harus berada di AWS Wilayah yang sama dengan bangunan Anda. Misalnya, jika Anda menginstruksikan CodeBuild untuk menjalankan bangunan di Wilayah Timur AS (Ohio), ember ini juga harus berada di Wilayah Timur AS (Ohio).

Untuk informasi selengkapnya, lihat [Membuat Bucket](#) di Panduan Pengguna Layanan Penyimpanan Sederhana Amazon.

Note

Meskipun CodeBuild juga mendukung input build yang disimpan di CodeCommit GitHub,, dan repositori Bitbucket, tutorial ini tidak menunjukkan cara menggunakannya. Untuk informasi selengkapnya, lihat [Rencanakan pembangunan](#).

Langkah selanjutnya

[Langkah 4: Unggah kode sumber dan file buildspec](#)

Langkah 4: Unggah kode sumber dan file buildspec

(Langkah sebelumnya: [Langkah 3: Buat dua ember S3](#))

Pada langkah ini, Anda menambahkan kode sumber dan membangun file spesifikasi ke bucket input.

Menggunakan utilitas zip sistem operasi Anda, buat file bernama `MessageUtil.zip` yang mencakup `MessageUtil.java`, `TestMessageUtil.java`, `pom.xml`, dan `buildspec.yml`.

Struktur direktori `MessageUtil.zip` file harus terlihat seperti ini.

```
MessageUtil.zip
|-- pom.xml
|-- buildspec.yml
`-- src
    |-- main
    |   |-- java
    |   |-- MessageUtil.java
    |-- test
    |   |-- java
```

```
`-- TestMessageUtil.java
```

⚠ Important

Jangan sertakan (*root directory name*) direktori, hanya direktori dan file di (*root directory name*) direktori.

Unggah MessageUtil.zip file ke bucket input bernamacodebuild-*region-ID-account-ID*-input-bucket.

⚠ Important

Untuk CodeCommit, GitHub, dan repositori Bitbucket, menurut konvensi, Anda harus menyimpan file spesifikasi build bernama buildspec.yml di root (tingkat atas) setiap repositori atau menyertakan deklarasi spesifikasi build sebagai bagian dari definisi proyek build. Jangan membuat file ZIP yang berisi kode sumber repositori dan membangun file spesifikasi.

Untuk input build yang disimpan di bucket S3 saja, Anda harus membuat file ZIP yang berisi kode sumber dan, menurut konvensi, file spesifikasi build yang diberi nama buildspec.yml di root (tingkat atas) atau menyertakan deklarasi spesifikasi build sebagai bagian dari definisi proyek build.

Jika Anda ingin menggunakan nama yang berbeda untuk file spesifikasi build, atau Anda ingin mereferensikan spesifikasi build di lokasi selain root, Anda dapat menentukan penggantian spesifikasi build sebagai bagian dari definisi proyek build. Untuk informasi selengkapnya, lihat [Nama file Buildspec dan lokasi penyimpanan](#).

Langkah selanjutnya

[Langkah 5: Buat proyek build](#)

Langkah 5: Buat proyek build

(Langkah sebelumnya: [Langkah 4: Unggah kode sumber dan file buildspec](#))

Pada langkah ini, Anda membuat proyek build yang AWS CodeBuild digunakan untuk menjalankan build. Proyek build mencakup informasi tentang cara menjalankan build, termasuk tempat

mendapatkan kode sumber, lingkungan build mana yang akan digunakan, perintah build mana yang akan dijalankan, dan tempat menyimpan output build. Lingkungan build mewakili kombinasi sistem operasi, runtime bahasa pemrograman, dan alat yang CodeBuild digunakan untuk menjalankan build. Lingkungan build dinyatakan sebagai image Docker. Untuk informasi selengkapnya, lihat [ikhtisar Docker](#) di situs web Docker Docs.

Untuk lingkungan build ini, Anda menginstruksikan CodeBuild untuk menggunakan image Docker yang berisi versi Java Development Kit (JDK) dan Apache Maven.

Untuk membuat proyek build

1. Masuk ke AWS Management Console dan buka AWS CodeBuild konsol di <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Gunakan pemilih AWS wilayah untuk memilih AWS Wilayah yang CodeBuild didukung. Untuk informasi selengkapnya, lihat [AWS CodeBuild titik akhir dan kuota](#) di Referensi Umum Amazon Web Services
3. Jika halaman CodeBuild informasi ditampilkan, pilih Buat proyek build. Jika tidak, pada panel navigasi, perluas Build, pilih Build projects, lalu pilih Create build project.
4. Pada halaman Create build project, dalam konfigurasi Project, untuk nama Project, masukkan nama untuk proyek build ini (dalam contoh ini, `codebuild-demo-project`). Membangun nama proyek harus unik di setiap AWS akun. Jika Anda menggunakan nama yang berbeda, pastikan untuk menggunakannya di seluruh tutorial ini.

Note

Pada halaman Buat proyek build, Anda mungkin melihat pesan galat yang mirip dengan berikut ini: Anda tidak berwenang untuk melakukan operasi ini. . Ini kemungkinan besar karena Anda masuk ke AWS Management Console sebagai pengguna yang tidak memiliki izin untuk membuat proyek build.. Untuk memperbaikinya, keluar dari AWS Management Console, lalu masuk kembali dengan kredensial milik salah satu entitas IAM berikut:

- Pengguna administrator di AWS akun Anda. Untuk informasi selengkapnya, lihat [Membuat pengguna dan grup Akun AWS root pertama Anda](#) di Panduan pengguna.
- Pengguna di AWS akun Anda dengan `AWSCodeBuildAdminAccess`, `AmazonS3ReadOnlyAccess`, dan kebijakan `IAMFullAccess` terkelola yang dilampirkan ke pengguna tersebut atau ke grup IAM

yang menjadi milik pengguna tersebut. Jika Anda tidak memiliki pengguna atau grup di AWS akun Anda dengan izin ini, dan Anda tidak dapat menambahkan izin ini ke pengguna atau grup Anda, hubungi administrator AWS akun Anda untuk mendapatkan bantuan. Untuk informasi selengkapnya, lihat [AWS kebijakan terkelola \(standar\) untuk AWS CodeBuild](#).

Kedua opsi termasuk izin administrator yang memungkinkan Anda membuat proyek build sehingga Anda dapat menyelesaikan tutorial ini. Sebaiknya Anda selalu menggunakan izin minimum yang diperlukan untuk menyelesaikan tugas Anda. Untuk informasi selengkapnya, lihat [AWS CodeBuild referensi izin](#).

5. Di Sumber, untuk penyedia Sumber, pilih Amazon S3.
6. *Untuk Bucket, pilih **codebuild- Region-ID - Account-ID -input-bucket**.*
7. Untuk kunci objek S3, masukkan **MessageUtil.zip**.
8. Di Lingkungan, untuk gambar Lingkungan, biarkan gambar Terkelola dipilih.
9. Untuk sistem operasi, pilih Amazon Linux 2.
10. Untuk Runtime, pilih Standar.
11. Untuk Gambar, pilih `aws/codebuild/amazonlinux2-x86_64-standard:4.0`.
12. Dalam peran Layanan, biarkan peran layanan baru dipilih, dan biarkan nama Peran tidak berubah.
13. Untuk Buildspec, biarkan Gunakan file buildspec yang dipilih.
14. Di Artefak, untuk Jenis, pilih Amazon S3.
15. *Untuk nama Bucket, pilih **codebuild- Region-ID - Account-ID -output-bucket**.*
16. Biarkan Nama dan Jalur kosong.
17. Pilih Buat proyek build.

Langkah selanjutnya

[Langkah 6: Jalankan build](#)

Langkah 6: Jalankan build

(Langkah sebelumnya: [Langkah 5: Buat proyek build](#))

Pada langkah ini, Anda menginstruksikan AWS CodeBuild untuk menjalankan build dengan pengaturan dalam proyek build.

Untuk menjalankan build

1. Buka AWS CodeBuild konsol di <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Di panel navigasi, pilih Membangun proyek.
3. Dalam daftar proyek build, pilih codebuild-demo-project, lalu pilih Start build. Membangun segera dimulai.

Langkah selanjutnya

[Langkah 7: Lihat informasi build yang dirangkum](#)

Langkah 7: Lihat informasi build yang dirangkum

(Langkah sebelumnya: [Langkah 6: Jalankan build](#))

Pada langkah ini, Anda melihat informasi yang dirangkum tentang status build Anda.

Untuk melihat informasi build yang dirangkum

1. Jika <build-ID>halaman codebuild-demo-project: tidak ditampilkan, di bilah navigasi, pilih Build history. Selanjutnya, dalam daftar proyek build, untuk Project, pilih tautan Build run untuk codebuild-demo-project. Seharusnya hanya ada satu tautan yang cocok. (Jika Anda telah menyelesaikan tutorial ini sebelumnya, pilih tautan dengan nilai terbaru di kolom Selesai.)
2. Pada halaman status Build, dalam rincian Fase, fase build berikut harus ditampilkan, dengan Succeeded di kolom Status:
 - DIKIRIMKAN
 - ANTRIAN
 - PENYEDIAAN
 - UNDUH_SUMBER
 - MENGINSTAL
 - PRE_BUILD
 - MEMBANGUN
 - POST_BUILD

- UPLOAD_ARTEFAK
- FINALISASI
- SELESAI

Di Build Status, Succeeded harus ditampilkan.

Jika Anda melihat Dalam Proses sebagai gantinya, pilih tombol segarkan.

3. Di samping setiap fase build, nilai Durasi menunjukkan berapa lama fase build berlangsung. Nilai Waktu akhir menunjukkan kapan fase build itu berakhir.

Langkah selanjutnya

[Langkah 8: Lihat informasi build terperinci](#)

Langkah 8: Lihat informasi build terperinci

(Langkah sebelumnya: [Langkah 7: Lihat informasi build yang dirangkum](#))

Pada langkah ini, Anda melihat informasi mendetail tentang build Anda di CloudWatch Log.

Note

Untuk melindungi informasi sensitif, berikut ini disembunyikan di CodeBuild log:

- AWSID kunci akses. Untuk informasi selengkapnya, lihat [Mengelola Kunci Akses untuk Pengguna IAM](#) di Panduan AWS Identity and Access Management Pengguna.
- String ditentukan menggunakan Parameter Store. Untuk informasi selengkapnya, lihat [Systems Manager Parameter Store](#) dan [Systems Manager Parameter Store Console Walkthrough](#) di Panduan Pengguna Amazon EC2 Systems Manager.
- String ditentukan menggunakan AWS Secrets Manager. Untuk informasi selengkapnya, lihat [Manajemen kunci](#).

Untuk melihat informasi build yang mendetail

1. Dengan halaman detail build yang masih ditampilkan dari langkah sebelumnya, 10.000 baris terakhir dari log build akan ditampilkan di log Build. Untuk melihat seluruh log build di CloudWatch Log, pilih tautan Lihat seluruh log.

2. Di aliran CloudWatch log Log, Anda dapat menelusuri peristiwa log. Secara default, hanya kumpulan peristiwa log terakhir yang ditampilkan. Untuk melihat peristiwa log sebelumnya, gulir ke awal daftar.
3. Dalam tutorial ini, sebagian besar peristiwa log berisi informasi verbose tentang CodeBuild mengunduh dan menginstal file ketergantungan build ke lingkungan build-nya, yang mungkin tidak Anda pedulikan. Anda dapat menggunakan kotak Filter peristiwa untuk mengurangi informasi yang ditampilkan. Misalnya, jika Anda memasukkan "[INFO]" acara Filter, hanya peristiwa yang berisi yang [INFO] ditampilkan. Untuk informasi selengkapnya, lihat [Filter dan sintaks pola](#) di Panduan CloudWatch Pengguna Amazon.

Langkah selanjutnya

[Langkah 9: Dapatkan artefak keluaran build](#)

Langkah 9: Dapatkan artefak keluaran build

(Langkah sebelumnya: [Langkah 8: Lihat informasi build terperinci](#))

Pada langkah ini, Anda mendapatkan `messageUtil-1.0.jar` file yang CodeBuild dibangun dan diunggah ke bucket output.

Anda dapat menggunakan CodeBuild konsol atau konsol Amazon S3 untuk menyelesaikan langkah ini.

Untuk mendapatkan artefak keluaran build (AWS CodeBuild konsol)

1. Dengan CodeBuild konsol masih terbuka dan halaman detail build masih ditampilkan dari langkah sebelumnya, pilih tab Build details dan gulir ke bawah ke bagian Artefacts.

Note

Jika halaman detail build tidak ditampilkan, di bilah navigasi, pilih Riwayat build, lalu pilih tautan Build run.

2. Tautan ke folder Amazon S3 berada di bawah lokasi unggah Artefak. Tautan ini membuka folder di Amazon S3 tempat Anda menemukan file artefak keluaran `messageUtil-1.0.jar` build.

Untuk mendapatkan artefak keluaran build (konsol Amazon S3)

1. Buka konsol Amazon S3 di <https://console.aws.amazon.com/s3/>.
2. Buka codebuild-*region-ID-account-ID*-output-bucket.
3. Buka folder codebuild-demo-project.
4. Buka target folder, tempat Anda menemukan file artefak keluaran messageUtil-1.0.jar build.

Langkah selanjutnya

[Langkah 10: Hapus ember S3](#)

Langkah 10: Hapus ember S3

(Langkah sebelumnya: [Langkah 9: Dapatkan artefak keluaran build](#))

Untuk mencegah tagihan yang sedang berlangsung ke AWS akun Anda, Anda dapat menghapus bucket input dan output yang digunakan dalam tutorial ini. Untuk petunjuk, lihat [Menghapus atau Mengosongkan Bucket di Panduan](#) Pengguna Layanan Penyimpanan Sederhana Amazon.

Jika Anda menggunakan pengguna IAM atau pengguna IAM administrator untuk menghapus bucket ini, pengguna harus memiliki lebih banyak izin akses. Tambahkan pernyataan berikut di antara penanda (**### MULAI MENAMBAHKAN PERNYATAAN DI SINI ### dan ### AKHIR MENAMBAHKAN PERNYATAAN DI SINI ###**) ke kebijakan akses yang ada untuk pengguna.

Elips (...) dalam pernyataan ini digunakan untuk singkatnya. Jangan menghapus pernyataan apa pun dalam kebijakan akses yang ada. Jangan masukkan elips ini ke dalam kebijakan.

```
{
  "Version": "2012-10-17",
  "Id": "...",
  "Statement": [
    ### BEGIN ADDING STATEMENT HERE ###
    {
      "Effect": "Allow",
      "Action": [
        "s3:DeleteBucket",
        "s3:DeleteObject"
      ],
    }
  ],
}
```

```
    "Resource": "*"
  }
  ### END ADDING STATEMENT HERE ###
]
}
```

Langkah selanjutnya

[Membungkus](#)

Membungkus

Dalam tutorial ini, Anda digunakan AWS CodeBuild untuk membangun satu set file kelas Java ke dalam file JAR. Anda kemudian melihat hasil build.

Anda sekarang dapat mencoba menggunakan CodeBuild dalam skenario Anda sendiri. Ikuti petunjuk dalam [Rencanakan pembangunan](#). Jika Anda belum merasa siap, Anda mungkin ingin mencoba membuat beberapa sampel. Lihat informasi yang lebih lengkap di [Sampel](#).

Memulai AWS CodeBuild menggunakan AWS CLI

Dalam tutorial ini, Anda gunakan AWS CodeBuild untuk membangun kumpulan file input kode sumber sampel (disebut artefak input build atau input build) ke dalam versi kode sumber yang dapat diterapkan (disebut artefak keluaran build atau output build). Secara khusus, Anda menginstruksikan CodeBuild untuk menggunakan Apache Maven, alat build umum, untuk membangun satu set file kelas Java ke dalam file Java Archive (JAR). Anda tidak perlu terbiasa dengan Apache Maven atau Java untuk menyelesaikan tutorial ini.

Anda dapat bekerja dengan CodeBuild melalui CodeBuild konsol, AWS CodePipeline, AWS CLI, atau AWS SDK. Tutorial ini menunjukkan bagaimana menggunakan CodeBuild dengan AWS CLI. Untuk informasi tentang penggunaan CodePipeline, lihat [Gunakan CodePipeline dengan CodeBuild](#).

Important

Langkah-langkah dalam tutorial ini mengharuskan Anda untuk membuat sumber daya (misalnya, bucket S3) yang mungkin mengakibatkan biaya ke AWS akun Anda. Ini termasuk kemungkinan biaya untuk CodeBuild dan untuk AWS sumber daya dan tindakan yang terkait dengan Amazon S3, AWS KMS, dan CloudWatch Log. Untuk informasi selengkapnya,

lihat [CodeBuild harga](#), [harga Amazon S3](#), [AWS Key Management Service harga](#), dan [harga Amazon CloudWatch](#) .

Langkah-langkah

- [Langkah 1: Buat kode sumber](#)
- [Langkah 2: Buat file buildspec](#)
- [Langkah 3: Buat dua ember S3](#)
- [Langkah 4: Unggah kode sumber dan file buildspec](#)
- [Langkah 5: Buat proyek build](#)
- [Langkah 6: Jalankan build](#)
- [Langkah 7: Lihat informasi build yang dirangkum](#)
- [Langkah 8: Lihat informasi build terperinci](#)
- [Langkah 9: Dapatkan artefak keluaran build](#)
- [Langkah 10: Hapus ember S3](#)
- [Membungkus](#)

Langkah 1: Buat kode sumber

(Bagian dari: [Memulai AWS CodeBuild menggunakan AWS CLI](#))

Pada langkah ini, Anda membuat kode sumber yang CodeBuild ingin Anda buat ke bucket keluaran. Kode sumber ini terdiri dari dua file kelas Java dan file Apache Maven Project Object Model (POM).

1. Di direktori kosong di komputer atau instance lokal Anda, buat struktur direktori ini.

```
(root directory name)
  |-- src
    |-- main
    |   |-- java
    |-- test
    |   |-- java
```

2. Menggunakan editor teks pilihan Anda, buat file ini, beri nama `MessageUtil.java`, lalu simpan di `src/main/java` direktori.


```
public class MessageUtil {
    private String message;

    public MessageUtil(String message) {
        this.message = message;
    }

    public String printMessage() {
        System.out.println(message);
        return message;
    }

    public String salutationMessage() {
        message = "Hi!" + message;
        System.out.println(message);
        return message;
    }
}
```

File kelas ini menciptakan sebagai output string karakter yang dilewatkan ke dalamnya.

MessageUtilKonstruktor menetapkan string karakter. printMessageMetode ini menciptakan output. Output salutationMessage metode Hi! diikuti oleh string karakter.

3. Buat file ini, beri nama TestMessageUtil.java, lalu simpan di /src/test/java direktori.

```
import org.junit.Test;
import org.junit.Ignore;
import static org.junit.Assert.assertEquals;

public class TestMessageUtil {

    String message = "Robert";
    MessageUtil messageUtil = new MessageUtil(message);

    @Test
    public void testPrintMessage() {
        System.out.println("Inside testPrintMessage()");
        assertEquals(message,messageUtil.printMessage());
    }

    @Test
    public void testSalutationMessage() {
```

```
System.out.println("Inside testSalutationMessage()");
message = "Hi!" + "Robert";
assertEquals(message,messageUtil.salutationMessage());
}
}
```

File kelas ini menetapkan message variabel di MessageUtil kelas untuk Robert. Kemudian tes untuk melihat apakah message variabel berhasil diatur dengan memeriksa apakah string Robert dan Hi!Robert muncul dalam output.

4. Buat file ini, beri namapom.xml, lalu simpan di direktori root (tingkat atas).

```
<project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/
maven-v4_0_0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>org.example</groupId>
  <artifactId>messageUtil</artifactId>
  <version>1.0</version>
  <packaging>jar</packaging>
  <name>Message Utility Java Sample App</name>
  <dependencies>
    <dependency>
      <groupId>junit</groupId>
      <artifactId>junit</artifactId>
      <version>4.11</version>
      <scope>test</scope>
    </dependency>
  </dependencies>
  <build>
    <plugins>
      <plugin>
        <groupId>org.apache.maven.plugins</groupId>
        <artifactId>maven-compiler-plugin</artifactId>
        <version>3.8.0</version>
      </plugin>
    </plugins>
  </build>
</project>
```

Apache Maven menggunakan instruksi dalam file ini untuk mengonversi file `MessageUtil.java` dan menjadi `TestMessageUtil.java` file bernama `messageUtil-1.0.jar` dan kemudian menjalankan tes yang ditentukan.

Pada titik ini, struktur direktori Anda akan terlihat seperti ini.

```
(root directory name)
|-- pom.xml
`-- src
    |-- main
    |   |-- java
    |       |-- MessageUtil.java
    |-- test
    |   |-- java
    |       |-- TestMessageUtil.java
```

Langkah selanjutnya

[Langkah 2: Buat file buildspec](#)

Langkah 2: Buat file buildspec

(Langkah sebelumnya: [Langkah 1: Buat kode sumber](#))

Pada langkah ini, Anda membuat file spesifikasi build (spesifikasi build). Buildspec adalah kumpulan perintah build dan pengaturan terkait, dalam format YAMB, yang CodeBuild digunakan untuk menjalankan build. Tanpa spesifikasi build, CodeBuild tidak dapat berhasil mengonversi input build Anda menjadi keluaran build atau menemukan artefak keluaran build di lingkungan build untuk diunggah ke bucket keluaran Anda.

Buat file ini, beri nama `buildspec.yml`, lalu simpan di direktori root (tingkat atas).

```
version: 0.2

phases:
  install:
    runtime-versions:
      java: corretto11
  pre_build:
```

```
  commands:
    - echo Nothing to do in the pre_build phase...
build:
  commands:
    - echo Build started on `date`
    - mvn install
post_build:
  commands:
    - echo Build completed on `date`
artifacts:
  files:
    - target/messageUtil-1.0.jar
```

Important

Karena deklarasi spesifikasi build harus YAMB yang valid, spasi dalam deklarasi spesifikasi build menjadi penting. Jika jumlah spasi dalam deklarasi spesifikasi build Anda tidak cocok dengan yang ini, build mungkin akan segera gagal. Anda dapat menggunakan validator YAMB untuk menguji apakah deklarasi spesifikasi build Anda adalah YAMB yang valid.

Note

Alih-alih menyertakan file spesifikasi build dalam kode sumber, Anda dapat mendeklarasikan perintah build secara terpisah saat membuat project build. Ini sangat membantu jika Anda ingin membangun kode sumber Anda dengan perintah build yang berbeda tanpa memperbarui repositori kode sumber Anda setiap kali. Untuk informasi selengkapnya, lihat [Sintaks Buildspec](#).

Dalam deklarasi spesifikasi build ini:

- `version` mewakili versi standar spesifikasi build yang digunakan. Deklarasi spesifikasi build ini menggunakan versi terbaru, `0.2`.
- `phase` mewakili fase build di mana Anda dapat menginstruksikan CodeBuild untuk menjalankan perintah. Fase build ini tercantum di sini sebagai `install`, `pre_build`, `build`, dan `post_build`. Anda tidak dapat mengubah ejaan nama fase build ini, dan Anda tidak dapat membuat lebih banyak nama fase build.

Dalam contoh ini, selama `build` fase, CodeBuild jalankan `mvn install` perintah. Perintah ini menginstruksikan Apache Maven untuk mengkompilasi, menguji, dan mengemas file kelas Java yang dikompilasi ke dalam artefak keluaran build. Untuk kelengkapan, beberapa `echo` perintah ditempatkan di setiap fase build dalam contoh ini. Ketika Anda melihat informasi build rinci nanti dalam tutorial ini, output dari `echo` perintah ini dapat membantu Anda lebih memahami bagaimana CodeBuild menjalankan perintah dan urutan apa. (Meskipun semua fase build disertakan dalam contoh ini, Anda tidak diharuskan menyertakan fase build jika Anda tidak berencana menjalankan perintah apa pun selama fase tersebut.) Untuk setiap fase build, CodeBuild jalankan setiap perintah yang ditentukan, satu per satu, dalam urutan yang terdaftar, dari awal hingga akhir.

- `artifacts` mewakili kumpulan artefak keluaran build yang CodeBuild diunggah ke bucket keluaran. `files` mewakili file yang akan disertakan dalam output build. CodeBuild mengunggah `messageUtil-1.0.jar` file tunggal yang ditemukan di direktori `target` relatif di lingkungan build. Nama file `messageUtil-1.0.jar` dan nama direktori `target` didasarkan pada cara Apache Maven membuat dan menyimpan artefak keluaran build untuk contoh ini saja. Dalam build Anda sendiri, nama file dan direktori ini berbeda.

Lihat informasi yang lebih lengkap di [Referensi Buildspec](#).

Pada titik ini, struktur direktori Anda akan terlihat seperti ini.

```
(root directory name)
|-- pom.xml
|-- buildspec.yml
`-- src
    |-- main
    |   |-- java
    |   |-- MessageUtil.java
    |-- test
    |   |-- java
    |   |-- TestMessageUtil.java
```

Langkah selanjutnya

[Langkah 3: Buat dua ember S3](#)

Langkah 3: Buat dua ember S3

(Langkah sebelumnya: [Langkah 2: Buat file buildspec](#))

Meskipun Anda dapat menggunakan satu bucket untuk tutorial ini, dua bucket memudahkan untuk melihat dari mana input build berasal dan ke mana arah output build.

- Salah satu bucket ini (bucket input) menyimpan input build. Dalam tutorial ini, nama bucket input ini adalah `codebuild-region-ID-account-ID-input-bucket`, di mana *Region-ID* adalah *Region* bucket dan *Account-ID* adalah *ID* akun Anda. AWS AWS
- Bucket lainnya (ember keluaran) menyimpan output build. Dalam tutorial ini, nama bucket keluaran ini adalah `codebuild-region-ID-account-ID-output-bucket`.

Jika Anda memilih nama yang berbeda untuk ember ini, pastikan untuk menggunakannya di seluruh tutorial ini.

Kedua ember ini harus berada di AWS Wilayah yang sama dengan bangunan Anda. Misalnya, jika Anda menginstruksikan CodeBuild untuk menjalankan bangunan di Wilayah Timur AS (Ohio), ember ini juga harus berada di Wilayah Timur AS (Ohio).

Untuk informasi selengkapnya, lihat [Membuat Bucket](#) di Panduan Pengguna Layanan Penyimpanan Sederhana Amazon.

Note

Meskipun CodeBuild juga mendukung input build yang disimpan di CodeCommit GitHub,, dan repositori Bitbucket, tutorial ini tidak menunjukkan cara menggunakannya. Untuk informasi selengkapnya, lihat [Rencanakan pembangunan](#).

Langkah selanjutnya

[Langkah 4: Unggah kode sumber dan file buildspec](#)

Langkah 4: Unggah kode sumber dan file buildspec

(Langkah sebelumnya: [Langkah 3: Buat dua ember S3](#))

Pada langkah ini, Anda menambahkan kode sumber dan membangun file spesifikasi ke bucket input.

Menggunakan utilitas zip sistem operasi Anda, buat file bernama `MessageUtil.zip` yang mencakup `MessageUtil.java`, `TestMessageUtil.java`, `pom.xml`, dan `buildspec.yml`.

Struktur direktori MessageUtil.zip file harus terlihat seperti ini.

```
MessageUtil.zip
|-- pom.xml
|-- buildspec.yml
`-- src
    |-- main
    |   |-- java
    |   |   |-- MessageUtil.java
    |-- test
    |   |-- java
    |   |   |-- TestMessageUtil.java
```

Important

Jangan sertakan (*root directory name*) direktori, hanya direktori dan file di (*root directory name*) direktori.

Unggah MessageUtil.zip file ke bucket input bernamacodebuild-*region-ID-account-ID*-input-bucket.

Important

Untuk CodeCommit, GitHub, dan repositori Bitbucket, menurut konvensi, Anda harus menyimpan file spesifikasi build bernama buildspec.yml di root (tingkat atas) setiap repositori atau menyertakan deklarasi spesifikasi build sebagai bagian dari definisi proyek build. Jangan membuat file ZIP yang berisi kode sumber repositori dan membangun file spesifikasi.

Untuk input build yang disimpan di bucket S3 saja, Anda harus membuat file ZIP yang berisi kode sumber dan, menurut konvensi, file spesifikasi build yang diberi nama buildspec.yml di root (tingkat atas) atau menyertakan deklarasi spesifikasi build sebagai bagian dari definisi proyek build.

Jika Anda ingin menggunakan nama yang berbeda untuk file spesifikasi build, atau Anda ingin mereferensikan spesifikasi build di lokasi selain root, Anda dapat menentukan penggantian spesifikasi build sebagai bagian dari definisi proyek build. Untuk informasi selengkapnya, lihat [Nama file Buildspec dan lokasi penyimpanan](#).

Langkah selanjutnya

[Langkah 5: Buat proyek build](#)

Langkah 5: Buat proyek build

(Langkah sebelumnya: [Langkah 4: Unggah kode sumber dan file buildspec](#))

Pada langkah ini, Anda membuat proyek build yang AWS CodeBuild digunakan untuk menjalankan build. Proyek build mencakup informasi tentang cara menjalankan build, termasuk tempat mendapatkan kode sumber, lingkungan build mana yang akan digunakan, perintah build mana yang akan dijalankan, dan tempat menyimpan output build. Lingkungan build mewakili kombinasi sistem operasi, runtime bahasa pemrograman, dan alat yang CodeBuild digunakan untuk menjalankan build. Lingkungan build dinyatakan sebagai image Docker. Untuk informasi selengkapnya, lihat [ikhtisar Docker](#) di situs web Docker Docs.

Untuk lingkungan build ini, Anda menginstruksikan CodeBuild untuk menggunakan image Docker yang berisi versi Java Development Kit (JDK) dan Apache Maven.

Untuk membuat proyek build

1. Gunakan AWS CLI untuk menjalankan create-project perintah:

```
aws codebuild create-project --generate-cli-skeleton
```

Data berformat JSON muncul di output. Salin data ke file bernama `create-project.json` di lokasi di komputer lokal atau contoh di AWS CLI mana diinstal. Jika Anda memilih untuk menggunakan nama file yang berbeda, pastikan untuk menggunakannya di seluruh tutorial ini.

Ubah data yang disalin untuk mengikuti format ini, lalu simpan hasil Anda:

```
{
  "name": "codebuild-demo-project",
  "source": {
    "type": "S3",
    "location": "codebuild-region-ID-account-ID-input-bucket/MessageUtil.zip"
  },
  "artifacts": {
    "type": "S3",
    "location": "codebuild-region-ID-account-ID-output-bucket"
  },
}
```



```
"environment": {
  "type": "LINUX_CONTAINER",
  "image": "aws/codebuild/standard:5.0",
  "computeType": "BUILD_GENERAL1_SMALL"
},
"serviceRole": "serviceIAMRole"
}
```

Ganti *ServiceIamRole* dengan Amazon Resource Name (ARN) CodeBuild dari peran layanan (misalnya, `arn:aws:iam::account-ID:role/role-name` Untuk membuatnya, lihat [Membuat peran CodeBuild layanan](#).

Dalam data ini:

- `namemewakili` pengidentifikasi yang diperlukan untuk proyek build ini (dalam contoh ini, `codebuild-demo-project`). Nama proyek build harus unik di semua proyek build di akun Anda.
- Untuk `source`, `type` adalah nilai wajib yang mewakili jenis repositori kode sumber (dalam contoh ini, S3 untuk bucket Amazon S3).
- Untuk `source`, `location` mewakili jalur ke kode sumber (dalam contoh ini, nama bucket input diikuti dengan nama file ZIP).
- Untuk `artifacts`, `type` adalah nilai wajib yang mewakili tipe repositori artefak keluaran build (dalam contoh ini, S3 untuk bucket Amazon S3).
- Untuk `artifacts`, `location` mewakili nama bucket keluaran yang Anda buat atau identifikasi sebelumnya (dalam contoh ini, `codebuild-region-ID-account-ID-output-bucket`).
- Untuk `environment`, `type` adalah nilai wajib yang mewakili jenis lingkungan build (dalam contoh ini, `LINUX_CONTAINER`).
- Untuk `environment`, `image` adalah nilai wajib yang mewakili nama gambar Docker dan kombinasi tag yang digunakan proyek build ini, seperti yang ditentukan oleh jenis repositori gambar Docker (dalam contoh ini, `aws/codebuild/standard:5.0` untuk gambar Docker di repositori gambar Docker). CodeBuild `aws/codebuild/standard` adalah nama gambar Docker. `5.0` adalah tag dari gambar Docker.

Untuk menemukan lebih banyak gambar Docker yang dapat Anda gunakan dalam skenario Anda, lihat. [Membangun referensi lingkungan](#)

- Untuk `environment`, `computeType` adalah nilai wajib yang mewakili sumber daya komputasi CodeBuild menggunakan (dalam contoh ini, `BUILD_GENERAL1_SMALL`).

Note

Nilai lain yang tersedia dalam data berformat JSON asli, seperti `description`, `auth` (termasuk `type` dan `resource`), `buildspec`, (untuk) `path`, `namespaceType`, `name` (termasuk dan `artifacts`), `packaging`, `timeoutInMinutes`, `encryptionKey`, `name` dan `environmentVariables` (termasuk dan `value`) adalah `tags` opsional `key.value`. Mereka tidak digunakan dalam tutorial ini, jadi mereka tidak ditampilkan di sini. Untuk informasi selengkapnya, lihat [Buat proyek build \(AWS CLI\)](#).

2. Beralih ke direktori yang berisi file yang baru saja Anda simpan, lalu jalankan `create-project` perintah lagi.

```
aws codebuild create-project --cli-input-json file://create-project.json
```

Jika berhasil, data yang mirip dengan ini muncul di output.

```
{
  "project": {
    "name": "codebuild-demo-project",
    "serviceRole": "serviceIAMRole",
    "tags": [],
    "artifacts": {
      "packaging": "NONE",
      "type": "S3",
      "location": "codebuild-region-ID-account-ID-output-bucket",
      "name": "message-util.zip"
    },
    "lastModified": 1472661575.244,
    "timeoutInMinutes": 60,
    "created": 1472661575.244,
    "environment": {
      "computeType": "BUILD_GENERAL1_SMALL",
      "image": "aws/codebuild/standard:5.0",
      "type": "LINUX_CONTAINER",
      "environmentVariables": []
    },
    "source": {
```

```
    "type": "S3",
    "location": "codebuild-region-ID-account-ID-input-bucket/MessageUtil.zip"
  },
  "encryptionKey": "arn:aws:kms:region-ID:account-ID:alias/aws/s3",
  "arn": "arn:aws:codebuild:region-ID:account-ID:project/codebuild-demo-project"
}
}
```

- `project` mewakili informasi tentang proyek pembangunan ini.
- `tags` mewakili setiap tag yang dideklarasikan.
- `packaging` mewakili bagaimana artefak keluaran build disimpan dalam bucket keluaran. `NONE` berarti folder dibuat di bucket output. Artefak keluaran build disimpan di folder itu.
- `lastModified` mewakili waktu, dalam format waktu Unix, ketika informasi tentang proyek pembangunan terakhir diubah.
- `timeoutInMinutes` mewakili jumlah menit setelah itu CodeBuild menghentikan build jika build belum selesai. (Standarnya adalah 60 menit.)
- `created` mewakili waktu, dalam format waktu Unix, ketika proyek build dibuat.
- `environmentVariables` mewakili variabel lingkungan apa pun yang dideklarasikan dan tersedia CodeBuild untuk digunakan selama pembuatan.
- `encryptionKey` mewakili ARN dari kunci terkelola pelanggan yang CodeBuild digunakan untuk mengenkripsi artefak keluaran build.
- `arn` mewakili ARN dari proyek pembangunan.

Note

Setelah Anda menjalankan `create-project` perintah, pesan kesalahan yang mirip dengan berikut ini mungkin akan muncul: `User: User-ARN is not authorized to perform: codebuild:CreateProject` Hal ini kemungkinan besar karena Anda mengkonfigurasi AWS CLI dengan kredensi pengguna yang tidak memiliki izin yang cukup untuk digunakan CodeBuild untuk membuat proyek build. Untuk memperbaikinya, konfigurasi kredensi AWS CLI dengan milik salah satu entitas IAM berikut:

- Pengguna administrator di AWS akun Anda. Untuk informasi selengkapnya, lihat [Membuat pengguna dan grup Akun AWS root pertama Anda](#) di Panduan pengguna.
- Pengguna di AWS akun Anda dengan `AWSCodeBuildAdminAccess`, `AmazonS3ReadOnlyAccess`, dan kebijakan

IAMFullAccess terkelola yang dilampirkan ke pengguna tersebut atau ke grup IAM yang menjadi milik pengguna tersebut. Jika Anda tidak memiliki pengguna atau grup di AWS akun Anda dengan izin ini, dan Anda tidak dapat menambahkan izin ini ke pengguna atau grup Anda, hubungi administrator AWS akun Anda untuk mendapatkan bantuan. Untuk informasi selengkapnya, lihat [AWS kebijakan terkelola \(standar\) untuk AWS CodeBuild](#).

Langkah selanjutnya

[Langkah 6: Jalankan build](#)

Langkah 6: Jalankan build

(Langkah sebelumnya: [Langkah 5: Buat proyek build](#))

Pada langkah ini, Anda menginstruksikan AWS CodeBuild untuk menjalankan build dengan pengaturan dalam proyek build.

Untuk menjalankan build

1. Gunakan AWS CLI untuk menjalankan start-build perintah:

```
aws codebuild start-build --project-name project-name
```

Ganti nama *proyek dengan nama* proyek build Anda dari langkah sebelumnya (misalnya, codebuild-demo-project

2. Jika berhasil, data yang mirip dengan berikut ini muncul di output:

```
{
  "build": {
    "buildComplete": false,
    "initiator": "user-name",
    "artifacts": {
      "location": "arn:aws:s3:::codebuild-region-ID-account-ID-output-bucket/
message-util.zip"
    },
    "projectName": "codebuild-demo-project",
    "timeoutInMinutes": 60,
    "buildStatus": "IN_PROGRESS",
    "environment": {
      "computeType": "BUILD_GENERAL1_SMALL",
```

```
    "image": "aws/codebuild/standard:5.0",
    "type": "LINUX_CONTAINER",
    "environmentVariables": []
  },
  "source": {
    "type": "S3",
    "location": "codebuild-region-ID-account-ID-input-bucket/MessageUtil.zip"
  },
  "currentPhase": "SUBMITTED",
  "startTime": 1472848787.882,
  "id": "codebuild-demo-project:0cfbb6ec-3db9-4e8c-992b-1ab28EXAMPLE",
  "arn": "arn:aws:codebuild:region-ID:account-ID:build/codebuild-demo-
project:0cfbb6ec-3db9-4e8c-992b-1ab28EXAMPLE"
}
}
```

- `build` mewakili informasi tentang build ini.
 - `buildComplete` mewakili apakah build telah selesai (`true`). Atau, `false`.
 - `initiator` mewakili entitas yang memulai pembangunan.
 - `artifacts` mewakili informasi tentang keluaran build, termasuk lokasinya.
 - `projectName` mewakili nama proyek pembangunan.
 - `buildStatus` mewakili status build saat ini saat start-build perintah dijalankan.
 - `currentPhase` mewakili fase build saat ini saat start-build perintah dijalankan.
 - `startTime` mewakili waktu, dalam format waktu Unix, saat proses pembuatan dimulai.
 - `id` mewakili ID build.
 - `arn` mewakili ARN build.

Catat id nilainya. Anda membutuhkannya di langkah berikutnya.

Langkah selanjutnya

[Langkah 7: Lihat informasi build yang dirangkum](#)

Langkah 7: Lihat informasi build yang dirangkum

(Langkah sebelumnya: [Langkah 6: Jalankan build](#))

Pada langkah ini, Anda melihat informasi yang dirangkum tentang status build Anda.

Untuk melihat informasi build yang dirangkum

Gunakan AWS CLI untuk menjalankan perintah `batch-get-builds`.

```
aws codebuild batch-get-builds --ids id
```

Ganti *id* dengan id nilai yang muncul di output dari langkah sebelumnya.

Jika berhasil, data yang mirip dengan ini muncul di output.

```
{
  "buildsNotFound": [],
  "builds": [
    {
      "buildComplete": true,
      "phases": [
        {
          "phaseStatus": "SUCCEEDED",
          "endTime": 1472848788.525,
          "phaseType": "SUBMITTED",
          "durationInSeconds": 0,
          "startTime": 1472848787.882
        },
        ... The full list of build phases has been omitted for brevity ...
        {
          "phaseType": "COMPLETED",
          "startTime": 1472848878.079
        }
      ],
      "logs": {
        "groupName": "/aws/codebuild/codebuild-demo-project",
        "deepLink": "https://console.aws.amazon.com/cloudwatch/home?region=region-ID#logEvent:group=/aws/codebuild/codebuild-demo-project;stream=38ca1c4a-e9ca-4dbc-bef1-d52bfEXAMPLE",
        "streamName": "38ca1c4a-e9ca-4dbc-bef1-d52bfEXAMPLE"
      },
      "artifacts": {
        "md5sum": "MD5-hash",
        "location": "arn:aws:s3:::codebuild-region-ID-account-ID-output-bucket/message-util.zip",
        "sha256sum": "SHA-256-hash"
      },
      "projectName": "codebuild-demo-project",
    }
  ]
}
```

```

    "timeoutInMinutes": 60,
    "initiator": "user-name",
    "buildStatus": "SUCCEEDED",
    "environment": {
      "computeType": "BUILD_GENERAL1_SMALL",
      "image": "aws/codebuild/standard:5.0",
      "type": "LINUX_CONTAINER",
      "environmentVariables": []
    },
    "source": {
      "type": "S3",
      "location": "codebuild-region-ID-account-ID-input-bucket/MessageUtil.zip"
    },
    "currentPhase": "COMPLETED",
    "startTime": 1472848787.882,
    "endTime": 1472848878.079,
    "id": "codebuild-demo-project:38ca1c4a-e9ca-4dbc-bef1-d52bfEXAMPLE",
    "arn": "arn:aws:codebuild:region-ID:account-ID:build/codebuild-demo-
project:38ca1c4a-e9ca-4dbc-bef1-d52bfEXAMPLE"
  }
]
}

```

- `buildsNotFound` mewakili ID build untuk build apa pun yang informasinya tidak tersedia. Dalam contoh ini, itu harus kosong.
- `builds` mewakili informasi tentang setiap build di mana informasi tersedia. Dalam contoh ini, informasi tentang hanya satu build muncul di output.
 - `phases` mewakili serangkaian fase build yang CodeBuild berjalan selama proses build. Informasi tentang setiap fase build dicantumkan secara terpisah sebagai `startTime`, `endTime`, dan `durationInSeconds` (ketika fase build dimulai dan berakhir, dinyatakan dalam format waktu Unix, dan berapa lama berlangsung, dalam hitungan detik), dan `phaseType` seperti (`SUBMITTED`, `PROVISIONING`, `DOWNLOAD_SOURCE`, `INSTALL`, `PRE_BUILD`, `BUILD`, `POST_BUILD`, `UPLOAD_ARTIFACTS`, `FINALIZING`, atau `COMPLETED`) dan `phaseStatus` (seperti `SUCCEEDED`, `FAILED`, `FAULT`, `TIMED_OUT`, `IN_PROGRESS`, atau `STOPPED`). Pertama kali Anda menjalankan `batch-get-builds` perintah, mungkin tidak ada banyak (atau apa pun) fase. Setelah menjalankan `batch-get-builds` perintah berikutnya dengan ID build yang sama, lebih banyak fase build akan muncul di output.
- `logs` mewakili informasi di Amazon CloudWatch Logs tentang log build.

- `md5sum` dan `sha256sum` mewakili hash MD5 dan SHA-256 dari artefak keluaran build. Ini muncul di output hanya jika `packaging` nilai proyek build disetel ke `ZIP`. (Anda tidak menetapkan nilai ini dalam tutorial ini.) Anda dapat menggunakan hash ini bersama dengan alat checksum untuk mengonfirmasi integritas dan keaslian file.

Note

Anda juga dapat menggunakan konsol Amazon S3 untuk melihat hash ini. Pilih kotak di samping artefak keluaran build, pilih `Actions`, lalu pilih `Properties`. Di panel `Properties`, perluas `Metadata`, dan lihat nilai untuk `x-amz-meta-codebuild-content-md5` dan `-content-sha256`. `x-amz-meta-codebuild` (Di konsol Amazon S3, nilai `Etag` artefak keluaran build tidak boleh ditafsirkan sebagai hash MD5 atau SHA-256.)
Jika Anda menggunakan AWS SDK untuk mendapatkan hash ini, nilainya diberi nama `codebuild-content-md5` dan `codebuild-content-sha256`

- `endTime` mewakili waktu, dalam format waktu Unix, ketika proses pembangunan berakhir.

Note

Metadata Amazon S3 memiliki CodeBuild header bernama yang berisi CodeBuild build `x-amz-meta-codebuild-buildarn` yang menerbitkan `buildArn` artefak ke Amazon S3. Ditambahkan untuk memungkinkan pelacakan sumber untuk notifikasi dan referensi dari mana artefak tersebut dihasilkan. `buildArn`

Langkah selanjutnya

[Langkah 8: Lihat informasi build terperinci](#)

Langkah 8: Lihat informasi build terperinci

(Langkah sebelumnya: [Langkah 7: Lihat informasi build yang dirangkum](#))

Pada langkah ini, Anda melihat informasi mendetail tentang build Anda di CloudWatch Log.

Note

Untuk melindungi informasi sensitif, berikut ini disembunyikan di CodeBuild log:

- AWSID kunci akses. Untuk informasi selengkapnya, lihat [Mengelola Kunci Akses untuk Pengguna IAM](#) di Panduan AWS Identity and Access Management Pengguna.
- String ditentukan menggunakan Parameter Store. Untuk informasi selengkapnya, lihat [Systems Manager Parameter Store](#) dan [Systems Manager Parameter Store Console Walkthrough](#) di Panduan Pengguna Amazon EC2 Systems Manager.
- String ditentukan menggunakan AWS Secrets Manager. Untuk informasi selengkapnya, lihat [Manajemen kunci](#).

Untuk melihat informasi build yang mendetail

1. Gunakan browser web Anda untuk pergi ke deepLink lokasi yang muncul di output pada langkah sebelumnya (misalnya, `https://console.aws.amazon.com/cloudwatch/home?region=region-ID#logEvent:group=/aws/codebuild/codebuild-demo-project;stream=38ca1c4a-e9ca-4dbc-bef1-d52bfEXAMPLE`).
2. Di aliran CloudWatch log Log, Anda dapat menelusuri peristiwa log. Secara default, hanya kumpulan peristiwa log terakhir yang ditampilkan. Untuk melihat peristiwa log sebelumnya, gulir ke awal daftar.
3. Dalam tutorial ini, sebagian besar peristiwa log berisi informasi verbose tentang CodeBuild mengunduh dan menginstal file ketergantungan build ke dalam lingkungan build-nya, yang mungkin tidak Anda pedulikan. Anda dapat menggunakan kotak Filter peristiwa untuk mengurangi informasi yang ditampilkan. Misalnya, jika Anda memasukkan "[INFO]" acara Filter, hanya peristiwa yang berisi yang [INFO] ditampilkan. Untuk informasi selengkapnya, lihat [Filter dan sintaks pola](#) di Panduan CloudWatch Pengguna Amazon.

Bagian-bagian dari aliran CloudWatch log Log ini berkaitan dengan tutorial ini.

```
...
[Container] 2016/04/15 17:49:42 Entering phase PRE_BUILD
[Container] 2016/04/15 17:49:42 Running command echo Entering pre_build phase...
[Container] 2016/04/15 17:49:42 Entering pre_build phase...
[Container] 2016/04/15 17:49:42 Phase complete: PRE_BUILD Success: true
[Container] 2016/04/15 17:49:42 Entering phase BUILD
[Container] 2016/04/15 17:49:42 Running command echo Entering build phase...
[Container] 2016/04/15 17:49:42 Entering build phase...
[Container] 2016/04/15 17:49:42 Running command mvn install
[Container] 2016/04/15 17:49:44 [INFO] Scanning for projects...
```

```
[Container] 2016/04/15 17:49:44 [INFO]
[Container] 2016/04/15 17:49:44 [INFO]
-----
[Container] 2016/04/15 17:49:44 [INFO] Building Message Utility Java Sample App 1.0
[Container] 2016/04/15 17:49:44 [INFO]
-----
...
[Container] 2016/04/15 17:49:55
-----
[Container] 2016/04/15 17:49:55 T E S T S
[Container] 2016/04/15 17:49:55
-----
[Container] 2016/04/15 17:49:55 Running TestMessageUtil
[Container] 2016/04/15 17:49:55 Inside testSalutationMessage()
[Container] 2016/04/15 17:49:55 Hi!Robert
[Container] 2016/04/15 17:49:55 Inside testPrintMessage()
[Container] 2016/04/15 17:49:55 Robert
[Container] 2016/04/15 17:49:55 Tests run: 2, Failures: 0, Errors: 0, Skipped: 0, Time
elapsed: 0.018 sec
[Container] 2016/04/15 17:49:55
[Container] 2016/04/15 17:49:55 Results :
[Container] 2016/04/15 17:49:55
[Container] 2016/04/15 17:49:55 Tests run: 2, Failures: 0, Errors: 0, Skipped: 0
...
[Container] 2016/04/15 17:49:56 [INFO]
-----
[Container] 2016/04/15 17:49:56 [INFO] BUILD SUCCESS
[Container] 2016/04/15 17:49:56 [INFO]
-----
[Container] 2016/04/15 17:49:56 [INFO] Total time: 11.845 s
[Container] 2016/04/15 17:49:56 [INFO] Finished at: 2016-04-15T17:49:56+00:00
[Container] 2016/04/15 17:49:56 [INFO] Final Memory: 18M/216M
[Container] 2016/04/15 17:49:56 [INFO]
-----
[Container] 2016/04/15 17:49:56 Phase complete: BUILD Success: true
[Container] 2016/04/15 17:49:56 Entering phase POST_BUILD
[Container] 2016/04/15 17:49:56 Running command echo Entering post_build phase...
[Container] 2016/04/15 17:49:56 Entering post_build phase...
[Container] 2016/04/15 17:49:56 Phase complete: POST_BUILD Success: true
[Container] 2016/04/15 17:49:57 Preparing to copy artifacts
[Container] 2016/04/15 17:49:57 Assembling file list
[Container] 2016/04/15 17:49:57 Expanding target/messageUtil-1.0.jar
[Container] 2016/04/15 17:49:57 Found target/messageUtil-1.0.jar
```

```
[Container] 2016/04/15 17:49:57 Creating zip artifact
```

Dalam contoh ini, CodeBuild berhasil menyelesaikan fase pre-build, build, dan post-build build. Ini menjalankan tes unit dan berhasil membangun `messageUtil-1.0.jar` file.

Langkah selanjutnya

[Langkah 9: Dapatkan artefak keluaran build](#)

Langkah 9: Dapatkan artefak keluaran build

(Langkah sebelumnya: [Langkah 8: Lihat informasi build terperinci](#))

Pada langkah ini, Anda mendapatkan `messageUtil-1.0.jar` file yang CodeBuild dibangun dan diunggah ke bucket output.

Anda dapat menggunakan CodeBuild konsol atau konsol Amazon S3 untuk menyelesaikan langkah ini.

Untuk mendapatkan artefak keluaran build (AWS CodeBuild konsol)

1. Dengan CodeBuild konsol masih terbuka dan halaman detail build masih ditampilkan dari langkah sebelumnya, pilih tab Build details dan gulir ke bawah ke bagian Artefacts.

Note

Jika halaman detail build tidak ditampilkan, di bilah navigasi, pilih Riwayat build, lalu pilih tautan Build run.

2. Tautan ke folder Amazon S3 berada di bawah lokasi unggah Artefak. Tautan ini membuka folder di Amazon S3 tempat Anda menemukan file artefak keluaran `messageUtil-1.0.jar` build.

Untuk mendapatkan artefak keluaran build (konsol Amazon S3)

1. Buka konsol Amazon S3 di <https://console.aws.amazon.com/s3/>.
2. Buka `codebuild-region-ID-account-ID-output-bucket`.
3. Buka folder `codebuild-demo-project`.
4. Buka target folder, tempat Anda menemukan file artefak keluaran `messageUtil-1.0.jar` build.

Langkah selanjutnya

[Langkah 10: Hapus ember S3](#)

Langkah 10: Hapus ember S3

(Langkah sebelumnya: [Langkah 9: Dapatkan artefak keluaran build](#))

Untuk mencegah tagihan yang sedang berlangsung ke AWS akun Anda, Anda dapat menghapus bucket input dan output yang digunakan dalam tutorial ini. Untuk petunjuk, lihat [Menghapus atau Mengosongkan Bucket di Panduan](#) Pengguna Layanan Penyimpanan Sederhana Amazon.

Jika Anda menggunakan pengguna IAM atau pengguna IAM administrator untuk menghapus bucket ini, pengguna harus memiliki lebih banyak izin akses. Tambahkan pernyataan berikut di antara penanda (**### MULAI MENAMBAHKAN PERNYATAAN DI SINI ### dan ### AKHIR MENAMBAHKAN PERNYATAAN DI SINI ###**) ke kebijakan akses yang ada untuk pengguna.

Elips (...) dalam pernyataan ini digunakan untuk singkatnya. Jangan menghapus pernyataan apa pun dalam kebijakan akses yang ada. Jangan masukkan elips ini ke dalam kebijakan.

```
{
  "Version": "2012-10-17",
  "Id": "...",
  "Statement": [
    ### BEGIN ADDING STATEMENT HERE ###
    {
      "Effect": "Allow",
      "Action": [
        "s3:DeleteBucket",
        "s3:DeleteObject"
      ],
      "Resource": "*"
    }
    ### END ADDING STATEMENT HERE ###
  ]
}
```

Langkah selanjutnya

[Membungkus](#)

Membungkus

Dalam tutorial ini, Anda digunakan AWS CodeBuild untuk membangun satu set file kelas Java ke dalam file JAR. Anda kemudian melihat hasil build.

Anda sekarang dapat mencoba menggunakan CodeBuild dalam skenario Anda sendiri. Ikuti petunjuk dalam [Rencanakan pembangunan](#). Jika Anda belum merasa siap, Anda mungkin ingin mencoba membuat beberapa sampel. Lihat informasi yang lebih lengkap di [Sampel](#).

CodeBuild sampel

Kelompok sampel ini dapat digunakan untuk bereksperimen dengan AWS CodeBuild:

Topik

- [Gunakan sampel berbasis kasus untuk CodeBuild](#)
- [Sampel Microsoft Windows untuk CodeBuild](#)

Gunakan sampel berbasis kasus untuk CodeBuild

Anda dapat menggunakan sampel berbasis kasus penggunaan ini untuk bereksperimen dengan: AWS CodeBuild

[Sampel lintas layanan](#)

Daftar sampel lintas layanan untuk bereksperimen AWS CodeBuild.

[Membangun sampel lencana](#)

Menunjukkan cara mengatur CodeBuild dengan lencana build.

[Buat laporan pengujian menggunakan AWS CLI sampel](#)

Menggunakan AWS CLI untuk membuat, menjalankan, dan melihat hasil laporan pengujian.

[Sampel Docker untuk CodeBuild](#)

Menunjukkan cara menggunakan gambar Docker khusus, mempublikasikan gambar Docker ke repositori di Amazon ECR, dan menggunakan gambar Docker di registri pribadi.

[Host build output dalam bucket S3](#)

Menunjukkan cara membuat situs web statis di bucket S3 menggunakan artefak build yang tidak terenkripsi.

[Beberapa sumber input dan sampel artefak keluaran](#)

Menunjukkan cara menggunakan beberapa sumber input dan beberapa artefak keluaran dalam proyek build.

[Versi runtime dalam contoh file buildspec](#)

Menunjukkan cara menentukan runtime dan versinya di file buildspec.

[Contoh versi sumber](#)

Menunjukkan cara menggunakan versi tertentu dari sumber Anda dalam proyek CodeBuild build.

[Sampel repositori sumber pihak ketiga untuk CodeBuild](#)

Menunjukkan cara membuat BitBucket, Server GitHub Perusahaan, dan GitHub menarik permintaan dengan webhook menggunakan CodeBuild

[Gunakan versi semantik untuk memberi nama contoh artefak build](#)

Menunjukkan cara menggunakan versi semantik untuk membuat nama artefak pada waktu pembuatan.

Sampel lintas layanan untuk CodeBuild

Anda dapat menggunakan sampel lintas layanan ini untuk bereksperimen dengan AWS CodeBuild:

[Sampel Amazon ECR](#)

Menggunakan image Docker di repositori Amazon ECR untuk menggunakan Apache Maven untuk menghasilkan satu file JAR.

[Sampel Amazon EFS](#)

Menunjukkan cara mengonfigurasi file buildspec sehingga CodeBuild proyek dipasang dan dibangun di sistem file Amazon EFS.

[AWS CodePipeline sampel](#)

Menunjukkan cara menggunakan AWS CodePipeline untuk membuat build dengan build batch serta beberapa sumber input dan beberapa artefak keluaran.

[AWS Config sampel](#)

Menunjukkan cara mengatur AWS Config. Daftar CodeBuild sumber daya mana yang dilacak dan menjelaskan cara mencari CodeBuild proyek. AWS Config

[Buat contoh notifikasi](#)

Menggunakan Apache Maven untuk menghasilkan satu file JAR. Mengirim pemberitahuan build ke pelanggan topik Amazon SNS.

Sampel Amazon ECR untuk CodeBuild

Contoh ini menggunakan image Docker di repositori image Amazon Elastic Container Registry (Amazon ECR) Registry ECR) untuk membuat proyek Go sampel.

Important

Menjalankan sampel ini dapat mengakibatkan biaya ke AWS akun Anda. Ini termasuk kemungkinan biaya untuk AWS CodeBuild dan untuk AWS sumber daya dan tindakan yang terkait dengan Amazon S3, CloudWatch Log AWS KMS, dan Amazon ECR. Untuk informasi selengkapnya, lihat [CodeBuild harga](#), [harga Amazon S3](#), [harga, AWS Key Management Service harga](#) Amazon, dan [CloudWatch harga Amazon Elastic Container Registry](#).

Menjalankan sampel

Untuk menjalankan sampel ini

1. Untuk membuat dan mendorong gambar Docker ke repositori gambar Anda di Amazon ECR, selesaikan langkah-langkah di bagian “Menjalankan sampel” pada [Publikasikan gambar Docker ke sampel repositori gambar Amazon ECR](#)
2. Buat proyek Go:
 - a. Buat file seperti yang dijelaskan di [Pergi file proyek](#) bagian [Struktur proyek Go](#) dan topik ini, lalu unggah ke bucket input S3 atau repositori AWS CodeCommit GitHub, atau Bitbucket.

Important

Jangan mengunggah(*root directory name*), hanya file di dalamnya(*root directory name*).

Jika Anda menggunakan bucket input S3, pastikan untuk membuat file ZIP yang berisi file, lalu unggah ke bucket input. Jangan tambahkan (*root directory name*) ke file ZIP, hanya file di dalamnya(*root directory name*).

- b. Buat proyek build, jalankan build, dan lihat informasi build terkait.

Jika Anda menggunakan AWS CLI untuk membuat proyek build, input berformat JSON ke `create-project` perintah mungkin terlihat mirip dengan ini. (Ganti placeholder dengan nilai Anda sendiri.)


```
{
  "name": "sample-go-project",
  "source": {
    "type": "S3",
    "location": "codebuild-region-ID-account-ID-input-bucket/GoSample.zip"
  },
  "artifacts": {
    "type": "S3",
    "location": "codebuild-region-ID-account-ID-output-bucket",
    "packaging": "ZIP",
    "name": "GoOutputArtifact.zip"
  },
  "environment": {
    "type": "LINUX_CONTAINER",
    "image": "aws/codebuild/standard:5.0",
    "computeType": "BUILD_GENERAL1_SMALL"
  },
  "serviceRole": "arn:aws:iam::account-ID:role/role-name",
  "encryptionKey": "arn:aws:kms:region-ID:account-ID:key/key-ID"
}
```

- c. Untuk mendapatkan artefak keluaran build, buka bucket keluaran S3 Anda.
 - d. Unduh *GoOutputArtifact*.zip file ke komputer atau instance lokal Anda, lalu ekstrak konten file tersebut. Dalam konten yang diekstraksi, dapatkan hello file.
3. Jika salah satu dari berikut ini benar, Anda harus menambahkan izin ke repositori gambar Anda di Amazon ECR sehingga AWS CodeBuild dapat menarik image Docker-nya ke lingkungan build.
- Proyek Anda menggunakan CodeBuild kredensial untuk menarik gambar Amazon ECR. Ini dilambangkan dengan nilai CODEBUILD dalam imagePullCredentialsType atribut Anda. ProjectEnvironment
 - Proyek Anda menggunakan gambar ECR Amazon lintas akun. Dalam hal ini, proyek Anda harus menggunakan peran layanannya untuk menarik gambar Amazon ECR. Untuk mengaktifkan perilaku ini, setel imagePullCredentialsType atribut Anda ProjectEnvironment keSERVICE_ROLE.
1. Buka konsol Amazon ECR di <https://console.aws.amazon.com/ecr/>.
 2. Dalam daftar nama repositori, pilih nama repositori yang Anda buat atau pilih.
 3. Dari panel navigasi, pilih Izin, pilih Edit, lalu pilih Tambah pernyataan.

4. Untuk nama Pernyataan, masukkan pengenal (misalnya, **CodeBuildAccess**).
5. Untuk Efek, biarkan Izinkan dipilih. Ini menunjukkan bahwa Anda ingin mengizinkan akses ke AWS akun lain.
6. Untuk Principal, lakukan salah satu hal berikut:
 - Jika project Anda menggunakan CodeBuild kredensial untuk menarik image Amazon ECR, di Prinsipal layanan, masukkan. **codebuild.amazonaws.com**
 - Jika project Anda menggunakan image Amazon ECR lintas akun, untuk ID AWS akun, masukkan ID AWS akun yang ingin Anda akses.
7. Lewati daftar Semua entitas IAM.
8. Untuk Tindakan, pilih tindakan khusus tarik: ecr:, ecr: GetDownloadUrlForLayer, dan ecr: BatchGetImage. BatchCheckLayerAvailability
9. Untuk Ketentuan, tambahkan yang berikut ini:

```
{
  "StringEquals":{
    "aws:SourceAccount": "<AWS-account-ID>",
    "aws:SourceArn": "arn:aws:codebuild:<region>:<AWS-account-ID>:project/<project-name>"
  }
}
```

10Pilih Simpan.

Kebijakan ini ditampilkan di Izin. Kepala sekolah adalah apa yang Anda masukkan untuk Principal pada langkah 3 dari prosedur ini:

- Jika project Anda menggunakan CodeBuild kredensial untuk menarik image Amazon ECR, **"codebuild.amazonaws.com"** muncul di bawah Prinsipal layanan.
- Jika project Anda menggunakan image Amazon ECR lintas akun, ID AWS akun yang ingin Anda berikan akses akan muncul di bawah ID AWS Akun.

Kebijakan contoh berikut menggunakan CodeBuild kredensial dan gambar ECR Amazon lintas akun.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
```

```

    "Sid": "CodeBuildAccessPrincipal",
    "Effect": "Allow",
    "Principal": {
      "Service": "codebuild.amazonaws.com"
    },
    "Action": [
      "ecr:GetDownloadUrlForLayer",
      "ecr:BatchGetImage",
      "ecr:BatchCheckLayerAvailability"
    ],
    "Condition": {
      "StringEquals": {
        "aws:SourceArn": "arn:aws:codebuild:<region>:<aws-account-id>:project/<project-name>",
        "aws:SourceAccount": "<aws-account-id>"
      }
    }
  },
  {
    "Sid": "CodeBuildAccessCrossAccount",
    "Effect": "Allow",
    "Principal": {
      "AWS": "arn:aws:iam::<AWS-account-ID>:root"
    },
    "Action": [
      "ecr:GetDownloadUrlForLayer",
      "ecr:BatchGetImage",
      "ecr:BatchCheckLayerAvailability"
    ]
  }
]
}

```

- Jika proyek Anda menggunakan CodeBuild kredensial dan Anda ingin CodeBuild proyek Anda memiliki akses terbuka ke repositori Amazon ECR, Anda dapat menghilangkan Condition kunci dan menambahkan kebijakan sampel berikut.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "CodeBuildAccessPrincipal",
      "Effect": "Allow",

```

```

    "Principal":{
      "Service":"codebuild.amazonaws.com"
    },
    "Action":[
      "ecr:GetDownloadUrlForLayer",
      "ecr:BatchGetImage",
      "ecr:BatchCheckLayerAvailability"
    ]
  },
  {
    "Sid":"CodeBuildAccessCrossAccount",
    "Effect":"Allow",
    "Principal":{
      "AWS":"arn:aws:iam::<AWS-account-ID>:root"
    },
    "Action":[
      "ecr:GetDownloadUrlForLayer",
      "ecr:BatchGetImage",
      "ecr:BatchCheckLayerAvailability"
    ]
  }
]
}

```

4. Buat proyek build, jalankan build, dan lihat informasi build.

Jika Anda menggunakan AWS CLI untuk membuat proyek build, input berformat JSON ke `create-project` perintah mungkin terlihat mirip dengan ini. (Ganti placeholder dengan nilai Anda sendiri.)

```

{
  "name": "amazon-ecr-sample-project",
  "source": {
    "type": "S3",
    "location": "codebuild-region-ID-account-ID-input-bucket/GoSample.zip"
  },
  "artifacts": {
    "type": "S3",
    "location": "codebuild-region-ID-account-ID-output-bucket",
    "packaging": "ZIP",
    "name": "GoOutputArtifact.zip"
  },
  "environment": {

```

```
"type": "LINUX_CONTAINER",
  "image": "account-ID.dkr.ecr.region-ID.amazonaws.com/your-Amazon-ECR-repo-name:tag",
  "computeType": "BUILD_GENERAL1_SMALL"
},
"serviceRole": "arn:aws:iam::account-ID:role/role-name",
"encryptionKey": "arn:aws:kms:region-ID:account-ID:key/key-ID"
}
```

5. Untuk mendapatkan artefak keluaran build, buka bucket keluaran S3 Anda.
6. Unduh *GoOutputArtifact*.zip file ke komputer atau instance lokal Anda, lalu ekstrak konten *GoOutputArtifact*.zip file tersebut. Dalam konten yang diekstraksi, dapatkan hello file.

Struktur proyek Go

Sampel ini mengasumsikan struktur direktori ini.

```
(root directory name)
### buildspec.yml
### hello.go
```

Pergi file proyek

Sampel ini menggunakan file-file ini.

buildspec.yml(dalam(*root directory name*))

```
version: 0.2

phases:
  install:
    runtime-versions:
      golang: 1.13
  build:
    commands:
      - echo Build started on `date`
      - echo Compiling the Go code
      - go build hello.go
  post_build:
    commands:
      - echo Build completed on `date`
```

```
artifacts:
  files:
    - hello
```

hello.go(dalam(*root directory name*))

```
package main
import "fmt"

func main() {
    fmt.Println("hello world")
    fmt.Println("1+1 =", 1+1)
    fmt.Println("7.0/3.0 =", 7.0/3.0)
    fmt.Println(true && false)
    fmt.Println(true || false)
    fmt.Println(!true)
}
```

Sumber daya terkait

- Untuk informasi tentang memulai AWS CodeBuild, lihat [Memulai dengan AWS CodeBuild menggunakan konsol](#).
- Untuk informasi tentang pemecahan masalah di CodeBuild, lihat [Pemecahan masalah AWS CodeBuild](#)
- Untuk informasi tentang kuota di CodeBuild, lihat [Kuota untuk AWS CodeBuild](#).

Sampel Amazon Elastic File System untuk AWS CodeBuild

Anda mungkin ingin membuat AWS CodeBuild build di Amazon Elastic File System, layanan file bersama yang dapat diskalakan untuk instans Amazon EC2. Kapasitas penyimpanan dengan Amazon EFS elastis, sehingga tumbuh atau menyusut saat file ditambahkan dan dihapus. Ini memiliki antarmuka layanan web sederhana yang dapat Anda gunakan untuk membuat dan mengkonfigurasi sistem file. Ini juga mengelola semua infrastruktur penyimpanan file untuk Anda, jadi Anda tidak perlu khawatir tentang menyebarkan, menambal, atau memelihara konfigurasi sistem file. Untuk informasi selengkapnya, lihat [Apa itu Amazon Elastic File System?](#) di Panduan Pengguna Amazon Elastic File System.

Contoh ini menunjukkan kepada Anda cara mengonfigurasi CodeBuild proyek sehingga dipasang dan kemudian membangun aplikasi Java ke sistem file Amazon EFS. Sebelum memulai, Anda harus


memiliki aplikasi Java yang siap dibangun yang diunggah ke bucket input S3 atau, GitHub Enterprise Server AWS CodeCommit GitHub, atau repositori Bitbucket.

Data dalam perjalanan untuk sistem file Anda dienkripsi. Untuk mengenkripsi data dalam perjalanan menggunakan gambar yang berbeda, lihat [Mengenkripsi data](#) dalam perjalanan.

Langkah-langkah tingkat tinggi

Sampel ini mencakup tiga langkah tingkat tinggi yang diperlukan untuk menggunakan Amazon EFS dengan AWS CodeBuild:

1. Buat virtual private cloud (VPC) di akun Anda AWS .
2. Buat sistem file yang menggunakan VPC ini.
3. Buat dan bangun CodeBuild proyek yang menggunakan VPC. CodeBuild Proyek ini menggunakan yang berikut ini untuk mengidentifikasi sistem file:
 - Pengidentifikasi sistem file yang unik. Anda memilih pengenal saat menentukan sistem file dalam proyek build Anda.
 - ID sistem file. ID ditampilkan saat Anda melihat sistem file di konsol Amazon EFS.
 - Titik pemasangan. Ini adalah direktori di wadah Docker Anda yang memasang sistem file.
 - Opsi pemasangan. Ini termasuk rincian tentang cara me-mount sistem file.


 Note

Sistem file yang dibuat di Amazon EFS hanya didukung pada platform Linux.

Buat VPC menggunakan AWS CloudFormation

Buat VPC Anda dengan template. AWS CloudFormation

1. Ikuti petunjuk yang digunakan [AWS CloudFormationTemplat VPC](#) AWS CloudFormation untuk membuat VPC.

 Note

VPC yang dibuat oleh AWS CloudFormation template ini memiliki dua subnet pribadi dan dua subnet publik. Anda hanya boleh menggunakan subnet pribadi ketika Anda

menggunakan AWS CodeBuild untuk me-mount sistem file yang Anda buat di Amazon EFS. Jika Anda menggunakan salah satu subnet publik, build gagal.

2. [Masuk ke AWS Management Console dan buka konsol VPC Amazon di https://console.aws.amazon.com/vpc/.](https://console.aws.amazon.com/vpc/)
3. Pilih VPC yang Anda buat dengan AWS CloudFormation
4. Pada tab Deskripsi, catat nama VPC Anda dan ID-nya. Keduanya diperlukan saat Anda membuat AWS CodeBuild proyek nanti dalam sampel ini.

Buat sistem file Amazon Elastic File System dengan VPC Anda

Buat sistem file Amazon EFS sederhana untuk sampel ini menggunakan VPC yang Anda buat sebelumnya.

1. Masuk ke AWS Management Console dan buka konsol Amazon EFS di <https://console.aws.amazon.com/efs/>.
2. Pilih Buat sistem file.
3. Dari VPC, pilih nama VPC yang Anda catat sebelumnya dalam sampel ini.
4. Biarkan Availability Zones yang terkait dengan subnet Anda dipilih.
5. Pilih Langkah Selanjutnya.
6. Di Tambahkan tag, untuk kunci Nama default, di Nilai, masukkan nama sistem file Amazon EFS Anda.
7. Tetap pilih Bursting dan Tujuan Umum sebagai mode kinerja dan throughput default Anda, lalu pilih Langkah Berikutnya.
8. Untuk Konfigurasi akses klien, pilih Langkah Berikutnya.
9. Pilih Buat Sistem File.
10. (Opsional) Sebaiknya tambahkan kebijakan ke sistem file Amazon EFS Anda yang memberlakukan enkripsi data dalam perjalanan. Di konsol Amazon EFS, pilih Kebijakan sistem file, pilih Edit, pilih kotak berlabel Terapkan enkripsi dalam perjalanan untuk semua klien, lalu pilih Simpan.

Buat CodeBuild proyek untuk digunakan dengan Amazon EFS

Buat AWS CodeBuild proyek yang menggunakan VPC yang Anda buat sebelumnya dalam contoh ini. Saat build dijalankan, ia memasang sistem file Amazon EFS yang dibuat sebelumnya. Selanjutnya, ia menyimpan file.jar yang dibuat oleh aplikasi Java Anda di direktori mount point sistem file Anda.

1. Buka AWS CodeBuild konsol di <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Dari panel navigasi, pilih Build projects, lalu pilih Create build project.
3. Dalam Nama proyek, masukkan nama untuk proyek Anda.
4. Dari penyedia Sumber, pilih repositori yang berisi aplikasi Java yang ingin Anda bangun.
5. Masukkan informasi, seperti URL repositori, yang CodeBuild digunakan untuk menemukan aplikasi Anda. Pilihannya berbeda untuk setiap penyedia sumber. Untuk informasi selengkapnya, lihat [Choose source provider](#).
6. Dari gambar Lingkungan, pilih Gambar terkelola.
7. Dari Sistem operasi, pilih Amazon Linux 2.
8. Dari Runtime, pilih Standar.
9. Dari Gambar, pilih aws/codebuild/amazonlinux2-x86_64-standard:4.0.
10. Dari jenis Lingkungan, pilih Linux.
11. Di bawah Peran layanan, pilih Peran layanan baru. Di Nama peran, masukkan nama untuk peran yang CodeBuild dibuat untuk Anda.
12. Perluas Additional configuration (Konfigurasi tambahan).
13. Pilih Aktifkan bendera ini jika Anda ingin membuat gambar Docker atau ingin build Anda mendapatkan hak istimewa yang lebih tinggi.

Note

Secara default, daemon Docker diaktifkan untuk build non-VPC. Jika Anda ingin menggunakan kontainer Docker untuk build VPC, [lihat Runtime Privilege dan Kemampuan Linux di situs web Docker Docs dan](#) aktifkan mode istimewa. Juga, Windows tidak mendukung mode istimewa.

14. Dari VPC, pilih ID VPC.
15. Dari Subnet, pilih satu atau beberapa subnet pribadi yang terkait dengan VPC Anda. Anda harus menggunakan subnet pribadi dalam build yang memasang sistem file Amazon EFS. Jika Anda menggunakan subnet publik, build gagal.

16. Dari Grup Keamanan, pilih grup keamanan default.

17. Dalam sistem File, masukkan informasi berikut:

- Untuk Identifier, masukkan pengenal sistem file unik. Itu harus kurang dari 129 karakter dan hanya berisi karakter alfanumerik dan garis bawah. CodeBuild menggunakan pengenal ini untuk membuat variabel lingkungan yang mengidentifikasi sistem file elastis. Format variabel lingkungan `CODEBUILD_<file_system_identifier>` dalam huruf kapital. Misalnya, jika Anda masuk `my_efs`, variabel lingkungan adalah `CODEBUILD_MY_EFS`.
- Untuk ID, pilih ID sistem file.
- (Opsional) Masukkan direktori dalam sistem file. CodeBuild memasang direktori ini. Jika Anda membiarkan jalur Direktori kosong, CodeBuild pasang seluruh sistem file. Path relatif terhadap root sistem file.
- Untuk Mount point, masukkan path absolut direktori di container build tempat sistem file dipasang. Jika direktori ini tidak ada, CodeBuild buat selama pembuatan.
- (Opsional) Masukkan opsi pemasangan. Jika Anda membiarkan opsi Mount kosong, CodeBuild gunakan opsi pemasangan defaultnya:

```
nfsvers=4.1
rsize=1048576
wsize=1048576
hard
timeo=600
retrans=2
```

Untuk informasi selengkapnya, lihat [Opsi Pemasangan NFS yang Disarankan](#) di Panduan Pengguna Amazon Elastic File System.

18. Untuk spesifikasi Build, pilih Insert build command, lalu pilih Switch to editor.

19. Masukkan perintah spesifikasi build berikut ke dalam editor. Ganti `<file_system_identifier>` dengan pengenal yang Anda masukkan pada langkah 17. Gunakan huruf kapital (misalnya, `CODEBUILD_MY_EFS`).

```
version: 0.2
phases:
  install:
    runtime-versions:
      java: corretto11
  build:
```

```
commands:
  - mvn compile -Dgpg.skip=true -Dmaven.repo.local=
    $CODEBUILD_<file_system_identifier>
```

20. Gunakan nilai default untuk semua pengaturan lainnya, lalu pilih Buat proyek build. Ketika build Anda selesai, halaman konsol untuk project Anda akan ditampilkan.
21. Pilih Mulai membangun.

CodeBuild dan ringkasan sampel Amazon EFS

Setelah AWS CodeBuild proyek Anda dibangun:

- Anda memiliki file.jar yang dibuat oleh aplikasi Java Anda yang dibangun ke sistem file Amazon EFS Anda di bawah direktori mount point Anda.
- Variabel lingkungan yang mengidentifikasi sistem file Anda dibuat menggunakan pengenalan sistem file yang Anda masukkan saat Anda membuat proyek.

Untuk informasi selengkapnya, lihat [Memasang sistem file](#) di Panduan Pengguna Amazon Elastic File System.

Memecahkan masalah

Berikut ini adalah kesalahan yang mungkin Anda temui saat menyiapkan Amazon EFS dengan CodeBuild.

Topik

- [CLIENT_ERROR: pemasangan '127.0.0.1: /' gagal. izin ditolak](#)
- [CLIENT_ERROR: pemasangan '127.0.0.1: /' gagal. koneksi diatur ulang oleh rekan](#)
- [VPC_CLIENT_ERROR: Kesalahan EC2 yang tidak terduga: UnauthorizedOperation](#)

CLIENT_ERROR: pemasangan '127.0.0.1: /' gagal. izin ditolak

Otorisasi IAM tidak didukung untuk memasang Amazon EFS dengan CodeBuild. Jika Anda menggunakan kebijakan sistem file Amazon EFS khusus, Anda harus memberikan akses baca dan tulis ke semua prinsip IAM. Sebagai contoh:

```
"Principal": {
```

```
"AWS": "*"
}
```

CLIENT_ERROR: pemasangan '127.0.0.1: /' gagal. koneksi diatur ulang oleh rekan

Ada dua kemungkinan penyebab kesalahan ini:

- Subnet CodeBuild VPC berada di zona ketersediaan yang berbeda dari target pemasangan Amazon EFS. Anda dapat mengatasinya dengan menambahkan subnet VPC di zona ketersediaan yang sama dengan target pemasangan Amazon EFS.
- Grup keamanan tidak memiliki izin untuk berkomunikasi dengan Amazon EFS. Anda dapat mengatasinya dengan menambahkan aturan masuk untuk mengizinkan semua lalu lintas dari VPC (tambahkan blok CIDR utama untuk VPC Anda), atau grup keamanan itu sendiri.

VPC_CLIENT_ERROR: Kesalahan EC2 yang tidak terduga: UnauthorizedOperation

Kesalahan ini terjadi ketika semua subnet dalam konfigurasi VPC Anda untuk CodeBuild proyek adalah subnet publik. Anda harus memiliki setidaknya satu subnet pribadi di VPC untuk memastikan konektivitas jaringan.

CodePipeline sampel untuk CodeBuild

Topik

- [AWS CodePipeline integrasi dengan CodeBuild dan batch build](#)
- [AWS CodePipeline integrasi dengan CodeBuild dan beberapa sumber input dan sampel artefak keluaran](#)

AWS CodePipeline integrasi dengan CodeBuild dan batch build

AWS CodeBuild sekarang mendukung pembuatan batch. Contoh ini menunjukkan cara menggunakan AWS CodePipeline untuk membuat proyek build yang menggunakan build batch.

Anda dapat menggunakan file berformat JSON yang mendefinisikan struktur pipeline Anda, dan kemudian menggunakannya dengan file untuk membuat pipeline. AWS CLI Untuk informasi selengkapnya, lihat [Referensi struktur AWS CodePipeline pipa](#) di Panduan AWS CodePipeline Pengguna.

Batch build dengan artefak individual

Gunakan file JSON berikut sebagai contoh struktur pipeline yang membuat batch build dengan artefak terpisah. Untuk mengaktifkan batch build in CodePipeline, atur `BatchEnabled` parameter configuration objek `true`.

```
{
  "pipeline": {
    "roleArn": "arn:aws:iam::account-id:role/my-AWS-CodePipeline-service-role-name",
    "stages": [
      {
        "name": "Source",
        "actions": [
          {
            "inputArtifacts": [],
            "name": "Source1",
            "actionTypeId": {
              "category": "Source",
              "owner": "AWS",
              "version": "1",
              "provider": "S3"
            },
            "outputArtifacts": [
              {
                "name": "source1"
              }
            ],
            "configuration": {
              "S3Bucket": "<my-input-bucket-name>",
              "S3ObjectKey": "my-source-code-file-name.zip"
            },
            "runOrder": 1
          },
          {
            "inputArtifacts": [],
            "name": "Source2",
            "actionTypeId": {
              "category": "Source",
              "owner": "AWS",
              "version": "1",
              "provider": "S3"
            },
            "outputArtifacts": [
```

```
        {
          "name": "source2"
        }
      ],
      "configuration": {
        "S3Bucket": "<my-other-input-bucket-name>",
        "S3ObjectKey": "my-other-source-code-file-name.zip"
      },
      "runOrder": 1
    }
  ]
},
{
  "name": "Build",
  "actions": [
    {
      "inputArtifacts": [
        {
          "name": "source1"
        },
        {
          "name": "source2"
        }
      ],
      "name": "Build",
      "actionTypeId": {
        "category": "Build",
        "owner": "AWS",
        "version": "1",
        "provider": "CodeBuild"
      },
      "outputArtifacts": [
        {
          "name": "build1"
        },
        {
          "name": "build1_artifact1"
        },
        {
          "name": "build1_artifact2"
        },
        {
          "name": "build2_artifact1"
        }
      ],
    }
  ],
}
```

```
        {
          "name": "build2_artifact2"
        }
      ],
      "configuration": {
        "ProjectName": "my-build-project-name",
        "PrimarySource": "source1",
        "BatchEnabled": "true"
      },
      "runOrder": 1
    }
  ]
}
],
"artifactStore": {
  "type": "S3",
  "location": "<AWS-CodePipeline-internal-bucket-name>"
},
"name": "my-pipeline-name",
"version": 1
}
}
```

Berikut ini adalah contoh file CodeBuild buildspec yang akan bekerja dengan konfigurasi pipeline ini.

```
version: 0.2
batch:
  build-list:
    - identifier: build1
      env:
        compute-type: BUILD_GENERAL1_SMALL
    - identifier: build2
      env:
        compute-type: BUILD_GENERAL1_MEDIUM

phases:
  build:
    commands:
      - echo 'file' > output_file

artifacts:
  files:
    - output_file
```

```
secondary-artifacts:
  artifact1:
    files:
      - output_file
  artifact2:
    files:
      - output_file
```

Nama artefak keluaran yang ditentukan dalam file JSON pipeline harus cocok dengan pengenalan build dan artefak yang ditentukan dalam file buildspec Anda. *Sintaksnya adalah `buildIdentifier` untuk artefak utama, dan `buildIdentifier _ ArtifactIdentifier` untuk artefak sekunder.*

Misalnya, untuk nama artefak keluaran `build1`, CodeBuild akan mengunggah artefak utama `build1` ke lokasi `build1`. Untuk nama `outputbuild1_artifact1`, CodeBuild akan mengunggah `artifact1` artefak sekunder `build1` ke lokasi `build1_artifact1`, dan sebagainya. Jika hanya satu lokasi output yang ditentukan, nama harus *`buildIdentifier` saja*.

Setelah Anda membuat file JSON, Anda dapat membuat pipeline Anda. Gunakan AWS CLI untuk menjalankan perintah `create-pipeline` dan meneruskan file ke parameter `--cli-input-json`. Untuk informasi selengkapnya, lihat [Membuat pipeline \(CLI\)](#) di AWS CodePipeline Panduan Pengguna.

Batch build dengan artefak gabungan

Gunakan file JSON berikut sebagai contoh struktur pipeline yang membuat batch build dengan artefak gabungan. Untuk mengaktifkan batch build in CodePipeline, atur `BatchEnabled` parameter configuration objek ke `true`. Untuk menggabungkan artefak build ke lokasi yang sama, atur `CombineArtifacts` parameter configuration objek ke `true`.

```
{
  "pipeline": {
    "roleArn": "arn:aws:iam::account-id:role/my-AWS-CodePipeline-service-role-name",
    "stages": [
      {
        "name": "Source",
        "actions": [
          {
            "inputArtifacts": [],
            "name": "Source1",
            "actionTypeId": {
              "category": "Source",
              "owner": "AWS",
```



```
    "version": "1",
    "provider": "S3"
  },
  "outputArtifacts": [
    {
      "name": "source1"
    }
  ],
  "configuration": {
    "S3Bucket": "<my-input-bucket-name>",
    "S3ObjectKey": "my-source-code-file-name.zip"
  },
  "runOrder": 1
},
{
  "inputArtifacts": [],
  "name": "Source2",
  "actionTypeId": {
    "category": "Source",
    "owner": "AWS",
    "version": "1",
    "provider": "S3"
  },
  "outputArtifacts": [
    {
      "name": "source2"
    }
  ],
  "configuration": {
    "S3Bucket": "<my-other-input-bucket-name>",
    "S3ObjectKey": "my-other-source-code-file-name.zip"
  },
  "runOrder": 1
}
]
},
{
  "name": "Build",
  "actions": [
    {
      "inputArtifacts": [
        {
          "name": "source1"
        }
      ],
```

```
    {
      "name": "source2"
    }
  ],
  "name": "Build",
  "actionTypeId": {
    "category": "Build",
    "owner": "AWS",
    "version": "1",
    "provider": "CodeBuild"
  },
  "outputArtifacts": [
    {
      "name": "output1 "
    }
  ],
  "configuration": {
    "ProjectName": "my-build-project-name",
    "PrimarySource": "source1",
    "BatchEnabled": "true",
    "CombineArtifacts": "true"
  },
  "runOrder": 1
}
]
}
],
"artifactStore": {
  "type": "S3",
  "location": "<AWS-CodePipeline-internal-bucket-name>"
},
"name": "my-pipeline-name",
"version": 1
}
}
```

Berikut ini adalah contoh file CodeBuild buildspec yang akan bekerja dengan konfigurasi pipeline ini.

```
version: 0.2
batch:
  build-list:
    - identifier: build1
  env:
```

```
    compute-type: BUILD_GENERAL1_SMALL
  - identifier: build2
    env:
      compute-type: BUILD_GENERAL1_MEDIUM

phases:
  build:
    commands:
      - echo 'file' > output_file

artifacts:
  files:
    - output_file
```

Jika artefak gabungan diaktifkan untuk pembuatan batch, hanya ada satu output yang diizinkan. CodeBuild akan menggabungkan artefak utama dari semua build menjadi satu file ZIP tunggal.

Setelah Anda membuat file JSON, Anda dapat membuat pipeline Anda. Gunakan AWS CLI untuk menjalankan perintah `create-pipeline` dan meneruskan file ke parameter. `--cli-input-json` Untuk informasi selengkapnya, lihat [Membuat pipeline \(CLI\)](#) di AWS CodePipeline Panduan Pengguna.

AWS CodePipeline integrasi dengan CodeBuild dan beberapa sumber input dan sampel artefak keluaran

Sebuah AWS CodeBuild proyek dapat mengambil lebih dari satu sumber input. Itu juga dapat membuat lebih dari satu artefak keluaran. Contoh ini menunjukkan cara menggunakan AWS CodePipeline untuk membuat proyek build yang menggunakan beberapa sumber input untuk membuat beberapa artefak keluaran. Untuk informasi selengkapnya, lihat [Beberapa sumber input dan sampel artefak keluaran](#).

Anda dapat menggunakan file berformat JSON yang mendefinisikan struktur pipeline Anda, dan kemudian menggunakannya dengan file untuk membuat pipeline. AWS CLI Gunakan file JSON berikut sebagai contoh struktur pipa yang membuat build dengan lebih dari satu sumber input dan lebih dari satu artefak keluaran. Kemudian dalam contoh ini Anda melihat bagaimana file ini menentukan beberapa input dan output. Untuk informasi selengkapnya, lihat [referensi struktur CodePipeline pipa](#) di Panduan AWS CodePipeline Pengguna.

```
{
  "pipeline": {
    "roleArn": "arn:aws:iam::account-id:role/my-AWS-CodePipeline-service-role-name",
    "stages": [
```

```
{
  "name": "Source",
  "actions": [
    {
      "inputArtifacts": [],
      "name": "Source1",
      "actionTypeId": {
        "category": "Source",
        "owner": "AWS",
        "version": "1",
        "provider": "S3"
      },
      "outputArtifacts": [
        {
          "name": "source1"
        }
      ],
      "configuration": {
        "S3Bucket": "my-input-bucket-name",
        "S3ObjectKey": "my-source-code-file-name.zip"
      },
      "runOrder": 1
    },
    {
      "inputArtifacts": [],
      "name": "Source2",
      "actionTypeId": {
        "category": "Source",
        "owner": "AWS",
        "version": "1",
        "provider": "S3"
      },
      "outputArtifacts": [
        {
          "name": "source2"
        }
      ],
      "configuration": {
        "S3Bucket": "my-other-input-bucket-name",
        "S3ObjectKey": "my-other-source-code-file-name.zip"
      },
      "runOrder": 1
    }
  ]
}
```

```
    },
    {
      "name": "Build",
      "actions": [
        {
          "inputArtifacts": [
            {
              "name": "source1"
            },
            {
              "name": "source2"
            }
          ],
          "name": "Build",
          "actionTypeId": {
            "category": "Build",
            "owner": "AWS",
            "version": "1",
            "provider": "AWS CodeBuild"
          },
          "outputArtifacts": [
            {
              "name": "artifact1"
            },
            {
              "name": "artifact2"
            }
          ],
          "configuration": {
            "ProjectName": "my-build-project-name",
            "PrimarySource": "source1"
          },
          "runOrder": 1
        }
      ]
    }
  ],
  "artifactStore": {
    "type": "S3",
    "location": "AWS-CodePipeline-internal-bucket-name"
  },
  "name": "my-pipeline-name",
  "version": 1
}
```

```
}
```

Dalam file JSON ini:

- Salah satu sumber input Anda harus ditunjuk `PrimarySource`. Sumber ini adalah direktori tempat CodeBuild mencari dan menjalankan file `buildspec` Anda. Kata kunci `PrimarySource` digunakan untuk menentukan sumber utama di `configuration` bagian CodeBuild tahap dalam file JSON.
- Setiap sumber input dipasang di direktorinya sendiri. Direktori ini disimpan dalam variabel lingkungan bawaan `$CODEBUILD_SRC_DIR` untuk sumber utama dan `$CODEBUILD_SRC_DIR_yourInputArtifactName` untuk semua sumber lainnya. Untuk pipeline dalam sampel ini, dua direktori sumber input adalah `$CODEBUILD_SRC_DIR` dan `$CODEBUILD_SRC_DIR_source2`. Untuk informasi selengkapnya, lihat [Variabel lingkungan di lingkungan build](#).
- Nama artefak keluaran yang ditentukan dalam file JSON pipeline harus cocok dengan nama artefak sekunder yang ditentukan dalam file `buildspec` Anda. Pipeline ini menggunakan file `buildspec` berikut. Untuk informasi selengkapnya, lihat [Sintaks Buildspec](#).

```
version: 0.2

phases:
  build:
    commands:
      - touch source1_file
      - cd $CODEBUILD_SRC_DIR_source2
      - touch source2_file

artifacts:
  files:
    - '**/*'
  secondary-artifacts:
    artifact1:
      base-directory: $CODEBUILD_SRC_DIR
      files:
        - source1_file
    artifact2:
      base-directory: $CODEBUILD_SRC_DIR_source2
      files:
        - source2_file
```

Setelah Anda membuat file JSON, Anda dapat membuat pipeline Anda. Gunakan AWS CLI untuk menjalankan perintah `create-pipeline` dan meneruskan file ke parameter. `--cli-input-json` Untuk informasi selengkapnya, lihat [Membuat pipeline \(CLI\)](#) di AWS CodePipeline Panduan Pengguna.

Gunakan AWS Config dengan CodeBuild sampel

AWS Config menyediakan inventaris sumber AWS daya Anda dan riwayat perubahan konfigurasi pada sumber daya ini. AWS Config sekarang mendukung AWS CodeBuild sebagai AWS sumber daya, yang berarti layanan dapat melacak CodeBuild proyek Anda. Untuk informasi lebih lanjut tentang AWS Config, lihat [Apa itu AWS Config?](#) di Panduan AWS Config Pengembang.

Anda dapat melihat informasi berikut tentang CodeBuild sumber daya di halaman Inventaris Sumber Daya di AWS Config konsol:

- Garis waktu CodeBuild konfigurasi Anda berubah.
- Detail konfigurasi untuk setiap CodeBuild proyek.
- Hubungan dengan AWS sumber daya lain.
- Daftar perubahan pada CodeBuild proyek Anda.

Prosedur dalam topik ini menunjukkan kepada Anda cara mengatur AWS Config dan mencari dan melihat CodeBuild proyek.

Topik

- [Prasyarat](#)
- [Mengatur AWS Config](#)
- [Cari AWS CodeBuild proyek](#)
- [Melihat detail AWS CodeBuild konfigurasi di AWS Config konsol](#)

Prasyarat

Buat AWS CodeBuild proyek Anda. Untuk petunjuk, lihat [Membuat proyek pembangunan](#).

Mengatur AWS Config

- [Menyiapkan AWS Config \(konsol\)](#)
- [Menyiapkan AWS Config \(AWS CLI\)](#)

Note

Setelah Anda menyelesaikan penyiapan, mungkin diperlukan waktu hingga 10 menit sebelum Anda dapat melihat AWS CodeBuild proyek di AWS Config konsol.

Cari AWS CodeBuild proyek

1. Masuk ke Konsol AWS Manajemen dan buka AWS Config konsol di <https://console.aws.amazon.com/config>.
2. Pada halaman Inventaris sumber daya, pilih AWS CodeBuild Proyek di bawah Jenis sumber daya. Gulir ke bawah dan pilih kotak centang CodeBuild proyek.
3. Pilih Lihat.
4. Setelah daftar CodeBuild proyek ditambahkan, pilih tautan nama CodeBuild proyek di kolom timeline Config.

Melihat detail AWS CodeBuild konfigurasi di AWS Config konsol

Saat Anda mencari sumber daya di halaman inventaris Sumber Daya, Anda dapat memilih AWS Config garis waktu untuk melihat detail tentang CodeBuild proyek Anda. Halaman detail untuk sumber daya menyediakan informasi tentang konfigurasi, hubungan, dan jumlah perubahan yang dilakukan pada sumber daya tersebut.

Blok di bagian atas halaman secara kolektif disebut timeline. Garis waktu menunjukkan tanggal dan waktu rekaman itu dibuat.

Untuk informasi selengkapnya, lihat [Melihat detail konfigurasi di AWS Config konsol](#) di Panduan AWS Config Pengembang.

Buat contoh notifikasi untuk CodeBuild

Amazon CloudWatch Events memiliki dukungan bawaan untuk AWS CodeBuild. CloudWatch Acara adalah aliran peristiwa sistem yang menjelaskan perubahan dalam AWS sumber daya Anda. Dengan CloudWatch Acara, Anda menulis aturan deklaratif untuk mengaitkan peristiwa yang menarik dengan tindakan otomatis yang akan diambil. Contoh ini menggunakan Amazon CloudWatch Events dan Amazon Simple Notification Service (Amazon SNS) untuk mengirim notifikasi build ke pelanggan setiap kali build berhasil, gagal, berpindah dari satu fase build ke fase build lainnya, atau kombinasi dari event ini.

⚠ Important

Menjalankan sampel ini dapat mengakibatkan biaya ke AWS akun Anda. Ini termasuk kemungkinan biaya untuk CodeBuild dan untuk AWS sumber daya dan tindakan yang terkait dengan Amazon CloudWatch dan Amazon SNS. Untuk informasi selengkapnya, lihat [CodeBuild harga](#), [CloudWatch harga Amazon](#), dan harga [Amazon SNS](#).

Menjalankan sampel

Untuk menjalankan sampel ini

1. Jika Anda sudah memiliki topik yang disiapkan dan berlangganan di Amazon SNS yang ingin Anda gunakan untuk sampel ini, lanjutkan ke langkah 4. Jika tidak, jika Anda menggunakan pengguna IAM alih-alih akun AWS root atau pengguna administrator untuk bekerja dengan Amazon SNS, tambahkan pernyataan berikut (*antara ### MULAI MENAMBAHKAN PERNYATAAN DI SINI ### dan ### AKHIR MENAMBAHKAN PERNYATAAN DI SINI ###*) ke pengguna (atau grup IAM yang dikaitkan dengan pengguna). Menggunakan akun AWS root tidak disarankan. Pernyataan ini memungkinkan melihat, membuat, berlangganan, dan menguji pengiriman notifikasi ke topik di Amazon SNS. Elipses (. . .) digunakan untuk singkatnya dan untuk membantu Anda menemukan tempat untuk menambahkan pernyataan. Jangan menghapus pernyataan apa pun, dan jangan ketik elips ini ke dalam kebijakan yang ada.

```
{
  "Statement": [
    ### BEGIN ADDING STATEMENT HERE ###
    {
      "Action": [
        "sns:CreateTopic",
        "sns:GetTopicAttributes",
        "sns:List*",
        "sns:Publish",
        "sns:SetTopicAttributes",
        "sns:Subscribe"
      ],
      "Resource": "*",
      "Effect": "Allow"
    },
    ### END ADDING STATEMENT HERE ###
    ...
  ]
}
```

```
],  
  "Version": "2012-10-17"  
}
```

Note

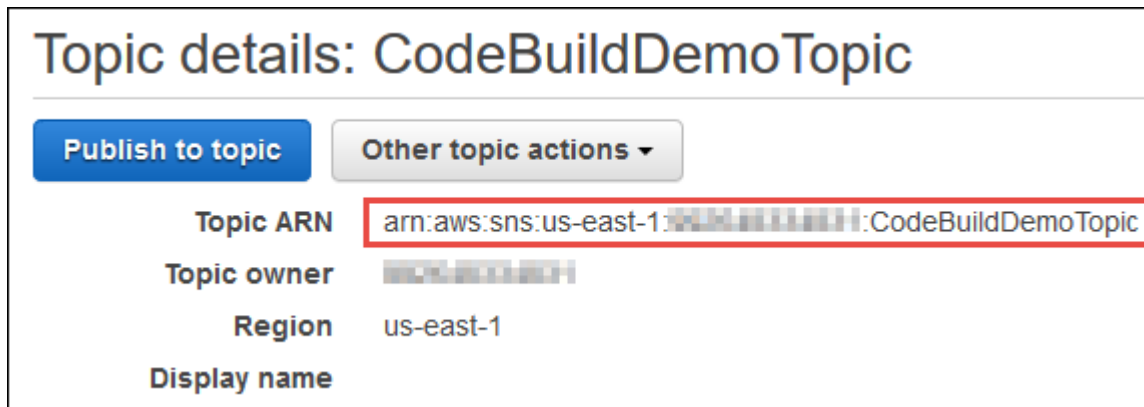
Entitas IAM yang memodifikasi kebijakan ini harus memiliki izin di IAM untuk memodifikasi kebijakan.

Untuk informasi selengkapnya, lihat [Mengedit kebijakan terkelola pelanggan](#) atau bagian “Mengedit atau menghapus kebijakan sebaris untuk grup, pengguna, atau peran” di [Bekerja dengan kebijakan sebaris \(konsol\)](#) di Panduan Pengguna IAM.

2. Buat atau identifikasi topik di Amazon SNS. AWS CodeBuild menggunakan CloudWatch Acara untuk mengirim pemberitahuan build ke topik ini melalui Amazon SNS.

Untuk membuat topik:

1. [Buka konsol Amazon SNS di https://console.aws.amazon.com/sns](https://console.aws.amazon.com/sns).
2. Pilih Buat topik.
3. Di Buat topik baru, untuk nama Topik, masukkan nama untuk topik (misalnya, **CodeBuildDemoTopic**). (Jika Anda memilih nama yang berbeda, gantilah di seluruh sampel ini.)
4. Pilih Buat topik.
5. Pada CodeBuildDemoTopic halaman Detail topik:, salin nilai Topik ARN. Anda memerlukan nilai ini untuk langkah berikutnya.



Topic details: CodeBuildDemoTopic

[Publish to topic](#) [Other topic actions ▾](#)

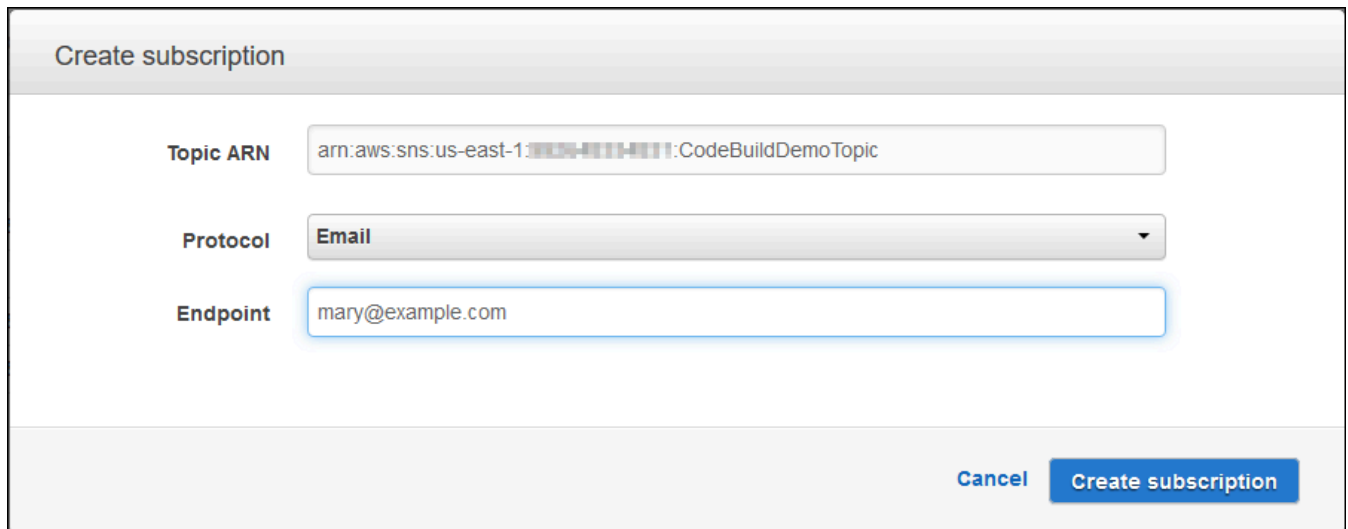
Topic ARN	arn:aws:sns:us-east-1:123456789012:CodeBuildDemoTopic
Topic owner	123456789012
Region	us-east-1
Display name	

Untuk informasi selengkapnya, lihat [Membuat topik](#) di Panduan Pengembang Amazon SNS.

3. Berlangganan satu atau beberapa penerima ke topik untuk menerima pemberitahuan email.

Untuk berlangganan penerima ke suatu topik:

1. Dengan konsol Amazon SNS terbuka dari langkah sebelumnya, di panel navigasi, pilih Langganan, lalu pilih Buat langganan.
2. Di Buat langganan, untuk Topik ARN, tempel topik ARN yang Anda salin dari langkah sebelumnya.
3. Untuk Protokol, pilih Email.
4. Untuk Endpoint, masukkan alamat email lengkap penerima.



The screenshot shows a 'Create subscription' form with the following fields:

- Topic ARN:** arn:aws:sns:us-east-1:123456789012:CodeBuildDemoTopic
- Protocol:** Email
- Endpoint:** mary@example.com

Buttons: Cancel, Create subscription

5. Pilih Buat Langganan.
6. Amazon SNS mengirimkan email konfirmasi berlangganan ke penerima. Untuk mulai menerima notifikasi email, penerima harus memilih tautan Konfirmasi langganan di email konfirmasi langganan. Setelah penerima mengklik tautan, jika berhasil berlangganan, Amazon SNS menampilkan pesan konfirmasi di browser web penerima.

Untuk informasi selengkapnya, lihat [Berlangganan topik](#) di Panduan Pengembang Amazon SNS.

4. Jika Anda menggunakan pengguna alih-alih akun AWS root atau pengguna administrator untuk bekerja dengan CloudWatch Acara, tambahkan pernyataan berikut (antara **### MULAI MENAMBAHKAN PERNYATAAN DI SINI ###** dan **### AKHIR MENAMBAHKAN PERNYATAAN**

DI SINI ###) ke pengguna (atau grup IAM yang dikaitkan dengan pengguna). Menggunakan akun AWS root tidak disarankan. Pernyataan ini digunakan untuk memungkinkan pengguna untuk bekerja dengan CloudWatch Acara. Elipses (. . .) digunakan untuk singkatnya dan untuk membantu Anda menemukan tempat untuk menambahkan pernyataan. Jangan menghapus pernyataan apa pun, dan jangan ketik elips ini ke dalam kebijakan yang ada.

```
{
  "Statement": [
    ### BEGIN ADDING STATEMENT HERE ###
    {
      "Action": [
        "events:*",
        "iam:PassRole"
      ],
      "Resource": "*",
      "Effect": "Allow"
    },
    ### END ADDING STATEMENT HERE ###
    ...
  ],
  "Version": "2012-10-17"
}
```

Note

Entitas IAM yang memodifikasi kebijakan ini harus memiliki izin di IAM untuk memodifikasi kebijakan.

Untuk informasi selengkapnya, lihat [Mengedit kebijakan terkelola pelanggan](#) atau bagian “Mengedit atau menghapus kebijakan sebaris untuk grup, pengguna, atau peran” di [Bekerja dengan kebijakan sebaris \(konsol\)](#) di Panduan Pengguna IAM.

5. Buat aturan di CloudWatch Acara. Untuk melakukan ini, buka CloudWatch konsol, di <https://console.aws.amazon.com/cloudwatch>.
6. Di panel navigasi, di bawah Acara, pilih Aturan, lalu pilih Buat aturan.
7. Pada Langkah 1: Buat halaman aturan, Pola Acara dan Pola acara Build untuk mencocokkan peristiwa berdasarkan layanan harus sudah dipilih.
8. Untuk Nama Layanan, pilih CodeBuild. Untuk Jenis Acara, Semua Acara seharusnya sudah dipilih.

9. Kode berikut harus ditampilkan di Pratinjau Pola Acara:

```
{
  "source": [
    "aws.codebuild"
  ]
}
```

10. Pilih Edit dan ganti kode di Pratinjau Pola Acara dengan salah satu dari dua pola aturan berikut.

Pola aturan pertama ini memicu peristiwa saat build dimulai atau diselesaikan untuk proyek build yang ditentukan di AWS CodeBuild

```
{
  "source": [
    "aws.codebuild"
  ],
  "detail-type": [
    "CodeBuild Build State Change"
  ],
  "detail": {
    "build-status": [
      "IN_PROGRESS",
      "SUCCEEDED",
      "FAILED",
      "STOPPED"
    ],
    "project-name": [
      "my-demo-project-1",
      "my-demo-project-2"
    ]
  }
}
```

Pada aturan sebelumnya, buat perubahan kode berikut sesuai kebutuhan.

- Untuk memicu peristiwa saat build dimulai atau selesai, tinggalkan semua nilai seperti yang ditunjukkan dalam `build-status` array, atau hapus array sama sekali. `build-status`
- Untuk memicu peristiwa hanya ketika build selesai, hapus `IN_PROGRESS` dari `build-status` array.

- Untuk memicu peristiwa hanya ketika build dimulai, hapus semua nilai kecuali `IN_PROGRESS` dari `build-status` array.
- Untuk memicu peristiwa untuk semua proyek build, hapus `project-name` array sama sekali.
- Untuk memicu peristiwa hanya untuk proyek build individual, tentukan nama setiap proyek build dalam `project-name` array.

Pola aturan kedua ini memicu peristiwa setiap kali build berpindah dari satu fase build ke fase build lainnya untuk proyek build yang ditentukan di AWS CodeBuild.

```
{
  "source": [
    "aws.codebuild"
  ],
  "detail-type": [
    "CodeBuild Build Phase Change"
  ],
  "detail": {
    "completed-phase": [
      "SUBMITTED",
      "PROVISIONING",
      "DOWNLOAD_SOURCE",
      "INSTALL",
      "PRE_BUILD",
      "BUILD",
      "POST_BUILD",
      "UPLOAD_ARTIFACTS",
      "FINALIZING"
    ],
    "completed-phase-status": [
      "TIMED_OUT",
      "STOPPED",
      "FAILED",
      "SUCCEEDED",
      "FAULT",
      "CLIENT_ERROR"
    ],
    "project-name": [
      "my-demo-project-1",
      "my-demo-project-2"
    ]
  ]
}
```

```
}
```

Pada aturan sebelumnya, buat perubahan kode berikut sesuai kebutuhan.

- Untuk memicu peristiwa untuk setiap perubahan fase build (yang mungkin mengirim hingga sembilan notifikasi untuk setiap build), tinggalkan semua nilai seperti yang ditunjukkan dalam `completed-phase array`, atau hapus `completed-phase array` sama sekali.
- Untuk memicu peristiwa hanya untuk perubahan fase build individual, hapus nama setiap fase build dalam `completed-phase array` yang tidak ingin Anda picu event.
- Untuk memicu peristiwa untuk setiap perubahan status fase build, tinggalkan semua nilai seperti yang ditunjukkan dalam `completed-phase-status array`, atau hapus `completed-phase-status array` sama sekali.
- Untuk memicu peristiwa hanya untuk perubahan status fase build individual, hapus nama setiap status fase build dalam `completed-phase-status array` yang tidak ingin Anda picu event.
- Untuk memicu peristiwa untuk semua proyek build, hapus `project-name array`.
- Untuk memicu peristiwa untuk proyek build individual, tentukan nama setiap proyek build dalam `project-name array`.

Untuk informasi selengkapnya tentang pola peristiwa, lihat [Pola Peristiwa](#) di Panduan EventBridge Pengguna Amazon.

Untuk informasi selengkapnya tentang pemfilteran dengan pola peristiwa, lihat [Pemfilteran Berbasis Konten dengan Pola Peristiwa di Panduan](#) Pengguna Amazon. EventBridge

Note

Jika Anda ingin memicu peristiwa untuk perubahan status build dan perubahan fase build, Anda harus membuat dua aturan terpisah: satu untuk perubahan status build dan satu lagi untuk perubahan fase build. Jika Anda mencoba menggabungkan kedua aturan menjadi satu aturan, aturan gabungan mungkin menghasilkan hasil yang tidak terduga atau berhenti bekerja sama sekali.

Setelah selesai mengganti kode, pilih Simpan.

11. Untuk Targets (Target), pilih Add target (Tambahkan target).

12. Dalam daftar target, pilih topik SNS.
13. Untuk Topik, pilih topik yang Anda identifikasi atau buat sebelumnya.
14. Perluas Konfigurasi input, lalu pilih Input Transformer.
15. Dalam kotak Input Path, masukkan salah satu jalur input berikut.

Untuk aturan dengan detail-type nilaiCodeBuild Build State Change, masukkan yang berikut ini.

```
{"build-id": "$.detail.build-id", "project-name": "$.detail.project-name", "build-status": "$.detail.build-status"}
```

Untuk aturan dengan detail-type nilaiCodeBuild Build Phase Change, masukkan yang berikut ini.

```
{"build-id": "$.detail.build-id", "project-name": "$.detail.project-name", "completed-phase": "$.detail.completed-phase", "completed-phase-status": "$.detail.completed-phase-status"}
```

Untuk mendapatkan jenis informasi lain, lihat [Buat referensi format masukan pemberitahuan](#).

16. Dalam kotak Template Input, masukkan salah satu template input berikut.

Untuk aturan dengan detail-type nilaiCodeBuild Build State Change, masukkan yang berikut ini.

```
"Build '<build-id>' for build project '<project-name>' has reached the build status of '<build-status>'."
```

Untuk aturan dengan detail-type nilaiCodeBuild Build Phase Change, masukkan yang berikut ini.

```
"Build '<build-id>' for build project '<project-name>' has completed the build phase of '<completed-phase>' with a status of '<completed-phase-status>'."
```

17. Pilih Konfigurasi detail.
18. Pada Langkah 2: Konfigurasi halaman detail aturan, masukkan nama dan deskripsi opsional. Untuk Status, biarkan Diaktifkan dipilih.
19. Pilih Buat aturan.

20. Buat proyek build, jalankan build, dan lihat informasi build.
21. Konfirmasikan bahwa CodeBuild sekarang berhasil mengirim pemberitahuan build. Misalnya, periksa untuk melihat apakah email pemberitahuan build sekarang ada di kotak masuk Anda.

Untuk mengubah perilaku aturan, di CloudWatch konsol, pilih aturan yang ingin diubah, pilih Tindakan, lalu pilih Edit. Buat perubahan pada aturan, pilih Konfigurasi detail, lalu pilih Perbarui aturan.

Untuk berhenti menggunakan aturan untuk mengirim pemberitahuan build, di CloudWatch konsol, pilih aturan yang ingin Anda hentikan, pilih Tindakan, lalu pilih Nonaktifkan.

Untuk menghapus aturan sama sekali, di CloudWatch konsol, pilih aturan yang ingin Anda hapus, pilih Tindakan, lalu pilih Hapus.

Sumber daya terkait

- Untuk informasi tentang memulai AWS CodeBuild, lihat [Memulai dengan AWS CodeBuild menggunakan konsol](#).
- Untuk informasi tentang pemecahan masalah di CodeBuild, lihat [Pemecahan masalah AWS CodeBuild](#).
- Untuk informasi tentang kuota di CodeBuild, lihat [Kuota untuk AWS CodeBuild](#).

Buat referensi format masukan pemberitahuan

CloudWatch memberikan pemberitahuan dalam format JSON.

Pemberitahuan perubahan status build menggunakan format berikut:

```
{
  "version": "0",
  "id": "c030038d-8c4d-6141-9545-00ff7b7153EX",
  "detail-type": "CodeBuild Build State Change",
  "source": "aws.codebuild",
  "account": "123456789012",
  "time": "2017-09-01T16:14:28Z",
  "region": "us-west-2",
  "resources": [
    "arn:aws:codebuild:us-west-2:123456789012:build/my-sample-project:8745a7a9-
c340-456a-9166-edf953571bEX"
  ],
}
```

```
"detail":{
  "build-status": "SUCCEEDED",
  "project-name": "my-sample-project",
  "build-id": "arn:aws:codebuild:us-west-2:123456789012:build/my-sample-
project:8745a7a9-c340-456a-9166-edf953571bEX",
  "additional-information": {
    "artifact": {
      "md5sum": "da9c44c8a9a3cd4b443126e823168fEX",
      "sha256sum":
"6ccc2ae1df9d155ba83c597051611c42d60e09c6329dcb14a312cecc0a8e39EX",
      "location": "arn:aws:s3:::codebuild-123456789012-output-bucket/my-output-
artifact.zip"
    },
    "environment": {
      "image": "aws/codebuild/standard:5.0",
      "privileged-mode": false,
      "compute-type": "BUILD_GENERAL1_SMALL",
      "type": "LINUX_CONTAINER",
      "environment-variables": []
    },
    "timeout-in-minutes": 60,
    "build-complete": true,
    "initiator": "MyCodeBuildDemoUser",
    "build-start-time": "Sep 1, 2017 4:12:29 PM",
    "source": {
      "location": "codebuild-123456789012-input-bucket/my-input-artifact.zip",
      "type": "S3"
    },
    "logs": {
      "group-name": "/aws/codebuild/my-sample-project",
      "stream-name": "8745a7a9-c340-456a-9166-edf953571bEX",
      "deep-link": "https://console.aws.amazon.com/cloudwatch/home?region=us-
west-2#logEvent:group=/aws/codebuild/my-sample-project;stream=8745a7a9-c340-456a-9166-
edf953571bEX"
    },
    "phases": [
      {
        "phase-context": [],
        "start-time": "Sep 1, 2017 4:12:29 PM",
        "end-time": "Sep 1, 2017 4:12:29 PM",
        "duration-in-seconds": 0,
        "phase-type": "SUBMITTED",
        "phase-status": "SUCCEEDED"
      }
    ],
  },
}
```

```
{
  "phase-context": [],
  "start-time": "Sep 1, 2017 4:12:29 PM",
  "end-time": "Sep 1, 2017 4:13:05 PM",
  "duration-in-seconds": 36,
  "phase-type": "PROVISIONING",
  "phase-status": "SUCCEEDED"
},
{
  "phase-context": [],
  "start-time": "Sep 1, 2017 4:13:05 PM",
  "end-time": "Sep 1, 2017 4:13:10 PM",
  "duration-in-seconds": 4,
  "phase-type": "DOWNLOAD_SOURCE",
  "phase-status": "SUCCEEDED"
},
{
  "phase-context": [],
  "start-time": "Sep 1, 2017 4:13:10 PM",
  "end-time": "Sep 1, 2017 4:13:10 PM",
  "duration-in-seconds": 0,
  "phase-type": "INSTALL",
  "phase-status": "SUCCEEDED"
},
{
  "phase-context": [],
  "start-time": "Sep 1, 2017 4:13:10 PM",
  "end-time": "Sep 1, 2017 4:13:10 PM",
  "duration-in-seconds": 0,
  "phase-type": "PRE_BUILD",
  "phase-status": "SUCCEEDED"
},
{
  "phase-context": [],
  "start-time": "Sep 1, 2017 4:13:10 PM",
  "end-time": "Sep 1, 2017 4:14:21 PM",
  "duration-in-seconds": 70,
  "phase-type": "BUILD",
  "phase-status": "SUCCEEDED"
},
{
  "phase-context": [],
  "start-time": "Sep 1, 2017 4:14:21 PM",
  "end-time": "Sep 1, 2017 4:14:21 PM",
```

```

    "duration-in-seconds": 0,
    "phase-type": "POST_BUILD",
    "phase-status": "SUCCEEDED"
  },
  {
    "phase-context": [],
    "start-time": "Sep 1, 2017 4:14:21 PM",
    "end-time": "Sep 1, 2017 4:14:21 PM",
    "duration-in-seconds": 0,
    "phase-type": "UPLOAD_ARTIFACTS",
    "phase-status": "SUCCEEDED"
  },
  {
    "phase-context": [],
    "start-time": "Sep 1, 2017 4:14:21 PM",
    "end-time": "Sep 1, 2017 4:14:26 PM",
    "duration-in-seconds": 4,
    "phase-type": "FINALIZING",
    "phase-status": "SUCCEEDED"
  },
  {
    "start-time": "Sep 1, 2017 4:14:26 PM",
    "phase-type": "COMPLETED"
  }
]
},
"current-phase": "COMPLETED",
"current-phase-context": "[]",
"version": "1"
}
}

```

Pemberitahuan perubahan fase build menggunakan format berikut:

```

{
  "version": "0",
  "id": "43ddc2bd-af76-9ca5-2dc7-b695e15adeEX",
  "detail-type": "CodeBuild Build Phase Change",
  "source": "aws.codebuild",
  "account": "123456789012",
  "time": "2017-09-01T16:14:21Z",
  "region": "us-west-2",
  "resources": [

```

```

    "arn:aws:codebuild:us-west-2:123456789012:build/my-sample-project:8745a7a9-
c340-456a-9166-edf953571bEX"
  ],
  "detail":{
    "completed-phase": "COMPLETED",
    "project-name": "my-sample-project",
    "build-id": "arn:aws:codebuild:us-west-2:123456789012:build/my-sample-
project:8745a7a9-c340-456a-9166-edf953571bEX",
    "completed-phase-context": "[]",
    "additional-information": {
      "artifact": {
        "md5sum": "da9c44c8a9a3cd4b443126e823168fEX",
        "sha256sum":
"6ccc2ae1df9d155ba83c597051611c42d60e09c6329dcb14a312cecc0a8e39EX",
        "location": "arn:aws:s3:::codebuild-123456789012-output-bucket/my-output-
artifact.zip"
      },
      "environment": {
        "image": "aws/codebuild/standard:5.0",
        "privileged-mode": false,
        "compute-type": "BUILD_GENERAL1_SMALL",
        "type": "LINUX_CONTAINER",
        "environment-variables": []
      },
      "timeout-in-minutes": 60,
      "build-complete": true,
      "initiator": "MyCodeBuildDemoUser",
      "build-start-time": "Sep 1, 2017 4:12:29 PM",
      "source": {
        "location": "codebuild-123456789012-input-bucket/my-input-artifact.zip",
        "type": "S3"
      },
      "logs": {
        "group-name": "/aws/codebuild/my-sample-project",
        "stream-name": "8745a7a9-c340-456a-9166-edf953571bEX",
        "deep-link": "https://console.aws.amazon.com/cloudwatch/home?region=us-
west-2#logEvent:group=/aws/codebuild/my-sample-project;stream=8745a7a9-c340-456a-9166-
edf953571bEX"
      },
      "phases": [
        {
          "phase-context": [],
          "start-time": "Sep 1, 2017 4:12:29 PM",
          "end-time": "Sep 1, 2017 4:12:29 PM",

```

```
"duration-in-seconds": 0,
"phase-type": "SUBMITTED",
"phase-status": "SUCCEEDED"
},
{
  "phase-context": [],
  "start-time": "Sep 1, 2017 4:12:29 PM",
  "end-time": "Sep 1, 2017 4:13:05 PM",
  "duration-in-seconds": 36,
  "phase-type": "PROVISIONING",
  "phase-status": "SUCCEEDED"
},
{
  "phase-context": [],
  "start-time": "Sep 1, 2017 4:13:05 PM",
  "end-time": "Sep 1, 2017 4:13:10 PM",
  "duration-in-seconds": 4,
  "phase-type": "DOWNLOAD_SOURCE",
  "phase-status": "SUCCEEDED"
},
{
  "phase-context": [],
  "start-time": "Sep 1, 2017 4:13:10 PM",
  "end-time": "Sep 1, 2017 4:13:10 PM",
  "duration-in-seconds": 0,
  "phase-type": "INSTALL",
  "phase-status": "SUCCEEDED"
},
{
  "phase-context": [],
  "start-time": "Sep 1, 2017 4:13:10 PM",
  "end-time": "Sep 1, 2017 4:13:10 PM",
  "duration-in-seconds": 0,
  "phase-type": "PRE_BUILD",
  "phase-status": "SUCCEEDED"
},
{
  "phase-context": [],
  "start-time": "Sep 1, 2017 4:13:10 PM",
  "end-time": "Sep 1, 2017 4:14:21 PM",
  "duration-in-seconds": 70,
  "phase-type": "BUILD",
  "phase-status": "SUCCEEDED"
},
}
```

```
{
  "phase-context": [],
  "start-time": "Sep 1, 2017 4:14:21 PM",
  "end-time": "Sep 1, 2017 4:14:21 PM",
  "duration-in-seconds": 0,
  "phase-type": "POST_BUILD",
  "phase-status": "SUCCEEDED"
},
{
  "phase-context": [],
  "start-time": "Sep 1, 2017 4:14:21 PM",
  "end-time": "Sep 1, 2017 4:14:21 PM",
  "duration-in-seconds": 0,
  "phase-type": "UPLOAD_ARTIFACTS",
  "phase-status": "SUCCEEDED"
},
{
  "phase-context": [],
  "start-time": "Sep 1, 2017 4:14:21 PM",
  "end-time": "Sep 1, 2017 4:14:26 PM",
  "duration-in-seconds": 4,
  "phase-type": "FINALIZING",
  "phase-status": "SUCCEEDED"
},
{
  "start-time": "Sep 1, 2017 4:14:26 PM",
  "phase-type": "COMPLETED"
}
]
},
"completed-phase-status": "SUCCEEDED",
"completed-phase-duration-seconds": 4,
"version": "1",
"completed-phase-start": "Sep 1, 2017 4:14:21 PM",
"completed-phase-end": "Sep 1, 2017 4:14:26 PM"
}
```

Buat sampel lencana dengan CodeBuild

AWS CodeBuild sekarang mendukung penggunaan lencana build, yang menyediakan gambar (lencana) yang dapat disematkan dan dihasilkan secara dinamis yang menampilkan status build terbaru untuk sebuah proyek. Gambar ini dapat diakses melalui URL yang tersedia untuk umum

yang dihasilkan untuk CodeBuild proyek Anda. Ini memungkinkan siapa saja untuk melihat status CodeBuild proyek. Build badge tidak mengandung informasi keamanan apa pun, sehingga tidak memerlukan otentikasi.


Buat proyek build dengan lencana build diaktifkan (konsol)

1. Buka AWS CodeBuild konsol di <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Jika halaman CodeBuild informasi ditampilkan, pilih Buat proyek build. Jika tidak, pada panel navigasi, perluas Build, pilih Build projects, lalu pilih Create build project.
3. Di Nama proyek, masukkan nama untuk proyek pembangunan ini. Membangun nama proyek harus unik di setiap AWS akun. Anda juga dapat menyertakan deskripsi opsional proyek build untuk membantu pengguna lain memahami tujuan proyek ini.
4. Di Sumber, untuk penyedia Sumber, pilih jenis penyedia kode sumber, lalu lakukan salah satu hal berikut:

Note

CodeBuild tidak mendukung lencana build dengan penyedia sumber Amazon S3. Karena AWS CodePipeline menggunakan Amazon S3 untuk transfer artefak, lencana build tidak didukung untuk proyek build yang merupakan bagian dari pipeline yang dibuat CodePipeline

- Jika Anda memilih CodeCommit, maka untuk Repositori, pilih nama repositori. Pilih Aktifkan lencana build untuk membuat status build project Anda terlihat dan dapat disematkan.
- Jika Anda memilih GitHub, ikuti petunjuk untuk menghubungkan (atau menyambung kembali) dengan GitHub. Pada halaman GitHub Otorisasi aplikasi, untuk akses Organisasi, pilih Minta akses di sebelah setiap repositori yang AWS CodeBuild ingin Anda akses. Setelah Anda memilih Otorisasi aplikasi, kembali ke AWS CodeBuild konsol, untuk Repositori, pilih nama repositori yang berisi kode sumber. Pilih Aktifkan lencana build untuk membuat status build project Anda terlihat dan dapat disematkan.
- Jika Anda memilih Bitbucket, ikuti petunjuk untuk menghubungkan (atau menyambung kembali) dengan Bitbucket. Pada Bitbucket Konfirmasi akses ke halaman akun Anda, untuk akses Organisasi, pilih Berikan akses. Setelah Anda memilih Grant access, kembali ke AWS CodeBuild konsol, untuk Repository, pilih nama repositori yang berisi kode sumber. Pilih Aktifkan lencana build untuk membuat status build project Anda terlihat dan dapat disematkan.

 Important

Memperbarui sumber proyek Anda dapat memengaruhi keakuratan rencana pembuatan proyek.


5. Di Lingkungan:

Untuk citra Lingkungan, lakukan salah satu hal berikut:

- Untuk menggunakan image Docker yang dikelola oleh AWS CodeBuild, pilih Managed image, lalu buat pilihan dari Sistem operasi, Runtime, Image, dan versi Image. Buat pilihan dari jenis Lingkungan jika tersedia.
- Untuk menggunakan gambar Docker lain, pilih Custom image. Untuk jenis Lingkungan, pilih ARM, Linux, GPU Linux, atau Windows. Jika Anda memilih Registri lain, untuk URL registri eksternal, masukkan nama dan tag gambar Docker di Docker Hub, menggunakan format *docker repository/docker image name* Jika Anda memilih Amazon ECR, gunakan repositori Amazon ECR dan gambar Amazon ECR untuk memilih gambar Docker di akun Anda. AWS
- Untuk menggunakan image Docker pribadi, pilih Custom image. Untuk jenis Lingkungan, pilih ARM, Linux, GPU Linux, atau Windows. Untuk registri Gambar, pilih Registri lain, lalu masukkan ARN kredensial untuk gambar Docker pribadi Anda. Kredensialnya harus dibuat oleh Secrets Manager. Untuk informasi lebih lanjut, lihat [Apa itu AWS Secrets Manager?](#) di Panduan Pengguna AWS Secrets Manager .

6. Dalam peran Layanan, lakukan salah satu hal berikut:

- Jika Anda tidak memiliki peran CodeBuild layanan, pilih Peran layanan baru. Di Nama peran, masukkan nama untuk peran baru.
- Jika Anda memiliki peran CodeBuild layanan, pilih Peran layanan yang ada. Di Peran ARN, pilih peran layanan.

 Note

Saat menggunakan konsol untuk membuat atau memperbarui proyek build, Anda dapat membuat peran CodeBuild layanan secara bersamaan. Secara default, peran hanya berfungsi dengan proyek build tersebut. Jika Anda menggunakan konsol untuk mengaitkan peran layanan ini dengan proyek build lain, peran akan diperbarui agar

berfungsi dengan proyek build lainnya. Peran layanan dapat bekerja dengan hingga 10 proyek pembangunan.

7. Di Buildspec, lakukan salah satu hal berikut:

- Pilih Gunakan file buildspec untuk menggunakan file buildspec.yml di direktori root kode sumber.
- Pilih Sisipkan perintah build untuk menggunakan konsol untuk menyisipkan perintah build.

Lihat informasi yang lebih lengkap di [Referensi Buildspec](#).

8. Di Artefak, untuk Jenis, lakukan salah satu hal berikut:

- Jika Anda tidak ingin membuat artefak keluaran build, pilih Tidak ada artefak.
- Untuk menyimpan output build dalam bucket S3, pilih Amazon S3, lalu lakukan hal berikut:
 - Jika Anda ingin menggunakan nama proyek Anda untuk file atau folder ZIP keluaran build, biarkan Nama kosong. Jika tidak, masukkan nama. Secara default, nama artefak adalah nama proyek. Jika Anda ingin menggunakan nama yang berbeda, masukkan di kotak nama artefak. Jika Anda ingin mengeluarkan file ZIP, sertakan ekstensi zip.
 - Untuk nama Bucket, pilih nama bucket keluaran.
 - Jika Anda memilih Sisipkan perintah build sebelumnya dalam prosedur ini, untuk file Output, masukkan lokasi file dari build yang ingin Anda masukkan ke dalam file atau folder ZIP keluaran build. Untuk beberapa lokasi, pisahkan setiap lokasi dengan koma (misalnya, `appspect.yml, target/my-app.jar`). Untuk informasi lebih lanjut, lihat deskripsi files di [Sintaks Buildspec](#).

9. Perluas Konfigurasi tambahan dan pilih opsi yang sesuai.

10. Pilih Buat proyek build. Pada halaman Review, pilih Start build untuk menjalankan build.

Buat proyek build dengan lencana build diaktifkan (CLI)

Untuk informasi tentang membuat proyek build, lihat [Buat proyek build \(AWS CLI\)](#). Untuk menyertakan lencana build dengan AWS CodeBuild project Anda, Anda harus menentukan *BadgeEnabled* dengan nilai `true`

Mengakses lencana AWS CodeBuild build Anda

Anda dapat menggunakan AWS CodeBuild konsol atau AWS CLI untuk mengakses lencana build.

- Di CodeBuild konsol, dalam daftar proyek build, di kolom Name, pilih tautan yang sesuai dengan proyek build. Pada halaman Build project: **project-name**, di Configuration, pilih Salin URL badge. Untuk informasi selengkapnya, lihat [Melihat detail proyek pembangunan \(konsol\)](#).
- Di AWS CLI, jalankan `batch-get-projects` perintah. URL lencana build disertakan dalam bagian detail lingkungan proyek pada output. Untuk informasi selengkapnya, lihat [Lihat detail proyek build \(AWS CLI\)](#).

URL permintaan lencana build dibuat dengan cabang default umum, tetapi Anda dapat menentukan cabang apa pun di repositori sumber yang telah Anda gunakan untuk menjalankan build. Sebagai contoh:

```
https://codebuild.us-east-1.amazon.com/badges?uuid=...&branch=<branch>
```

Anda juga dapat menentukan tag dari repositori sumber Anda dengan mengganti `branch` parameter dengan `tag` parameter di URL lencana. Sebagai contoh:

```
https://codebuild.us-east-1.amazon.com/badges?uuid=...&tag=<tag>
```

Publikasikan lencana CodeBuild build Anda

Anda dapat menampilkan status build terbaru dalam file penurunan harga menggunakan URL lencana build Anda dalam gambar penurunan harga. Ini berguna untuk menampilkan status build terbaru di file `readme.md` di repositori sumber Anda (misalnya, atau). GitHub CodeCommit Sebagai contoh:

```

```

CodeBuild status lencana

- **PASSING** Build terbaru di cabang yang diberikan lulus.
- **GAGAL** Build terbaru di cabang tertentu kehabisan waktu, gagal, salah, atau dihentikan.
- **IN_PROGRESS** Build terbaru di cabang yang diberikan sedang berlangsung.
- **TIDAK DIKETAHUI** Proyek belum menjalankan build untuk cabang yang diberikan atau sama sekali. Selain itu, fitur lencana build mungkin telah dinonaktifkan.

Membuat laporan pengujian dalam CodeBuild menggunakan AWS CLI sampel

Pengujian yang Anda tentukan dalam file buildspec dijalankan selama build. Sampel ini menunjukkan cara menggunakan tes AWS CLI untuk menggabungkan pengujian ke dalam build in CodeBuild. Anda dapat menggunakan JUnit untuk membuat pengujian unit, atau Anda dapat menggunakan alat lain untuk membuat pengujian konfigurasi. Anda kemudian dapat mengevaluasi hasil tes untuk memperbaiki masalah atau mengoptimalkan aplikasi Anda.

Anda dapat menggunakan CodeBuild API atau AWS CodeBuild konsol untuk mengakses hasil pengujian. Contoh ini menunjukkan cara mengonfigurasi laporan agar hasil pengujiannya diekspor ke bucket S3.

Topik

- [Prasyarat](#)
- [Membuat grup laporan](#)
- [Konfigurasi proyek dengan grup laporan](#)
- [Jalankan dan lihat hasil laporan](#)

Prasyarat

- Buat kasus uji Anda. Sampel ini ditulis dengan asumsi bahwa Anda memiliki kasus uji untuk dimasukkan dalam laporan pengujian sampel Anda. Anda menentukan lokasi file pengujian Anda dalam file buildspec.

Format file laporan pengujian berikut didukung:

- Mentimun JSON (.json)
- JUnit XML (.xml)
- NUnit XML (.xml)
- NUnit3 XML (.xml.xml)
- TestNG XML (.xml)
- Studio Visual TRX (.trx)

Buat kasus pengujian Anda dengan kerangka pengujian apa pun yang dapat membuat file laporan dalam salah satu format ini (misalnya, plugin Surefire JUnit, TestNG, atau Mentimun).

- Buat ember S3 dan catat namanya. Untuk informasi selengkapnya, lihat [Bagaimana cara membuat bucket S3?](#) di Panduan Pengguna Amazon S3.
- Buat peran IAM dan catat ARN-nya. Anda memerlukan ARN saat membuat proyek build Anda.
- Jika peran Anda tidak memiliki izin berikut, tambahkan.

```
{
  "Effect": "Allow",
  "Resource": [
    "*"
  ],
  "Action": [
    "codebuild:CreateReportGroup",
    "codebuild:CreateReport",
    "codebuild:UpdateReport",
    "codebuild:BatchPutTestCases"
  ]
}
```

Untuk informasi selengkapnya, lihat [Izin untuk operasi pelaporan pengujian](#).

Membuat grup laporan

1. Buat file bernama `CreateReportGroupInput.json`.
2. Buat folder di bucket S3 tempat hasil pengujian Anda diekspor.
3. Salin yang berikut ini ke dalam `CreateReportGroupInput.json`. Untuk `<bucket-name>`, gunakan nama bucket S3. Untuk `<path-to-folder>`, masukkan path ke folder di bucket S3 Anda.

```
{
  "name": "<report-name>",
  "type": "TEST",
  "exportConfig": {
    "exportConfigType": "S3",
    "s3Destination": {
      "bucket": "<bucket-name>",
      "path": "<path-to-folder>",
      "packaging": "NONE"
    }
  }
}
```

```
}
```

4. Jalankan perintah berikut di direktori yang berisi `CreateReportGroupInput.json`.

```
aws codebuild create-report-group --cli-input-json file://  
CreateReportGroupInput.json
```

Output-nya akan terlihat seperti berikut. Buat catatan ARN untuk `reportGroup` Anda menggunakannya saat membuat proyek yang menggunakan grup laporan ini.

```
{  
  "reportGroup": {  
    "arn": "arn:aws:codebuild:us-west-2:123456789012:report-group/<report-name>",  
    "name": "<report-name>",  
    "type": "TEST",  
    "exportConfig": {  
      "exportConfigType": "S3",  
      "s3Destination": {  
        "bucket": "<s3-bucket-name>",  
        "path": "<folder-path>",  
        "packaging": "NONE",  
        "encryptionKey": "arn:aws:kms:us-west-2:123456789012:alias/aws/s3"  
      }  
    },  
    "created": 1570837165.885,  
    "lastModified": 1570837165.885  
  }  
}
```

Konfigurasi proyek dengan grup laporan

Untuk menjalankan laporan, pertama-tama Anda membuat proyek CodeBuild build yang dikonfigurasi dengan grup laporan Anda. Kasus uji yang ditentukan untuk grup laporan dijalankan saat Anda menjalankan build.

1. Buat file `buildspec` bernama `buildspec.yml`
2. Gunakan YAMG berikut sebagai template untuk `buildspec.yml` file Anda. Pastikan untuk menyertakan perintah yang menjalankan pengujian Anda. Di `reports` bagian ini, tentukan file yang berisi hasil kasus uji Anda. File-file ini menyimpan hasil tes yang dapat Anda akses

CodeBuild. Mereka kedaluwarsa 30 hari setelah dibuat. File-file ini berbeda dari file hasil kasus uji mentah yang Anda ekspor ke bucket S3.

```
version: 0.2
  phases:
    install:
      runtime-versions:
        java: openjdk8
    build:
      commands:
        - echo Running tests
        - <enter commands to run your tests>

  reports:
    <report-name-or-arn>: #test file information
    files:
      - '<test-result-files>'
    base-directory: '<optional-base-directory>'
    discard-paths: false #do not remove file paths from test result files
```

Note

Alih-alih ARN dari grup laporan yang ada, Anda juga dapat menentukan nama untuk grup laporan yang belum dibuat. Jika Anda menentukan nama, bukan ARN, CodeBuild buat grup laporan saat menjalankan build. Namanya berisi nama proyek Anda dan nama yang Anda tentukan dalam file buildspec, dalam format ini: `project-name-report-group-name` Lihat informasi yang lebih lengkap di [Buat laporan pengujian](#) dan [Laporkan penamaan grup](#).

3. Buat file bernama `project.json`. File ini berisi input untuk `create-project` perintah.
4. Salin JSON berikut ke dalam `project.json`. Untuk `source`, masukkan jenis dan lokasi repositori yang berisi file sumber Anda. Untuk `serviceRole`, tentukan ARN dari peran yang Anda gunakan.

```
{
  "name": "test-report-project",
  "description": "sample-test-report-project",
  "source": {
    "type": "CODECOMMIT|CODEPIPELINE|GITHUB|S3|BITBUCKET|GITHUB_ENTERPRISE|NO_SOURCE",
```

```
"location": "<your-source-url>"
},
"artifacts": {
  "type": "NO_ARTIFACTS"
},
"cache": {
  "type": "NO_CACHE"
},
"environment": {
  "type": "LINUX_CONTAINER",
  "image": "aws/codebuild/standard:5.0",
  "computeType": "small"
},
"serviceRole": "arn:aws:iam::<your-aws-account-id>:role/service-role/<your-role-name>"
}
```

5. Jalankan perintah berikut di direktori yang berisi `project.json`. Ini menciptakan proyek bernama `test-project`.

```
aws codebuild create-project --cli-input-json file://project.json
```

Jalankan dan lihat hasil laporan

Di bagian ini, Anda menjalankan build proyek yang Anda buat sebelumnya. Selama proses pembuatan, CodeBuild buat laporan dengan hasil kasus uji. Laporan tersebut terdapat dalam grup laporan yang Anda tentukan.

1. Untuk memulai build, jalankan perintah berikut. `test-report-project` adalah nama proyek build yang dibuat di atas. Catat ID build yang muncul di output.

```
aws codebuild start-build --project-name test-report-project
```

2. Jalankan perintah berikut untuk mendapatkan informasi tentang build Anda, termasuk ARN laporan Anda. Untuk `<build-id>`, tentukan ID build Anda. Catat laporan ARN di `reportArns` properti output.

```
aws codebuild batch-get-builds --ids <build-id>
```


3. Jalankan perintah berikut untuk mendapatkan detail tentang laporan Anda. Untuk *<report-arn>*, tentukan laporan ARN Anda.

```
aws codebuild batch-get-reports --report-arns <report-arn>
```

Output-nya akan terlihat seperti berikut. Output sampel ini menunjukkan berapa banyak pengujian yang berhasil, gagal, dilewati, menghasilkan kesalahan, atau mengembalikan status yang tidak diketahui.

```
{
  "reports": [
    {
      "status": "FAILED",
      "reportGroupArn": "<report-group-arn>",
      "name": "<report-group-name>",
      "created": 1573324770.154,
      "exportConfig": {
        "exportConfigType": "S3",
        "s3Destination": {
          "bucket": "<your-S3-bucket>",
          "path": "<path-to-your-report-results>",
          "packaging": "NONE",
          "encryptionKey": "<encryption-key>"
        }
      },
      "expired": 1575916770.0,
      "truncated": false,
      "executionId": "arn:aws:codebuild:us-west-2:123456789012:build/<name-of-build-project>:2c254862-ddf6-4831-a53f-6839a73829c1",
      "type": "TEST",
      "arn": "<report-arn>",
      "testSummary": {
        "durationInNanoSeconds": 6657770,
        "total": 11,
        "statusCounts": {
          "FAILED": 3,
          "SKIPPED": 7,
          "ERROR": 0,
          "SUCCEEDED": 1,
          "UNKNOWN": 0
        }
      }
    }
  ]
}
```

```
    }
  ],
  "reportsNotFound": []
}
```

4. Jalankan perintah berikut untuk mencantumkan informasi tentang kasus uji untuk laporan Anda. Untuk `<report-arn>`, tentukan ARN laporan Anda. Untuk `--filter` parameter opsional, Anda dapat menentukan satu hasil status (SUCCEEDED, FAILED, SKIPPED, ERROR, atau UNKNOWN).

```
aws codebuild describe-test-cases \
  --report-arn <report-arn> \
  --filter status=SUCCEEDED|FAILED|SKIPPED|ERROR|UNKNOWN
```

Output-nya akan terlihat seperti berikut.

```
{
  "testCases": [
    {
      "status": "FAILED",
      "name": "Test case 1",
      "expired": 1575916770.0,
      "reportArn": "<report-arn>",
      "prefix": "Cucumber tests for agent",
      "message": "A test message",
      "durationInNanoSeconds": 1540540,
      "testRawDataPath": "<path-to-output-report-files>"
    },
    {
      "status": "SUCCEEDED",
      "name": "Test case 2",
      "expired": 1575916770.0,
      "reportArn": "<report-arn>",
      "prefix": "Cucumber tests for agent",
      "message": "A test message",
      "durationInNanoSeconds": 1540540,
      "testRawDataPath": "<path-to-output-report-files>"
    }
  ]
}
```

Sampel Docker untuk CodeBuild

Topik

- [Docker dalam sampel gambar khusus untuk CodeBuild](#)
- [Publikasikan gambar Docker ke sampel repositori gambar Amazon Elastic Container Registry untuk CodeBuild](#)
- [Registri pribadi dengan AWS Secrets Manager sampel untuk CodeBuild](#)

Docker dalam sampel gambar khusus untuk CodeBuild

Sampel ini membangun dan menjalankan image Docker dengan menggunakan AWS CodeBuild dan image build Docker kustom (docker:dinddi Docker Hub).

Untuk mempelajari cara membuat image Docker dengan menggunakan image build yang disediakan oleh CodeBuild dukungan Docker, lihat. [Publikasikan gambar Docker ke sampel repositori gambar Amazon ECR](#)

Important

Menjalankan sampel ini dapat mengakibatkan biaya ke AWS akun Anda. Ini termasuk kemungkinan biaya untuk CodeBuild dan untuk AWS sumber daya dan tindakan yang terkait dengan Amazon S3, AWS KMS, dan CloudWatch Log. Untuk informasi selengkapnya, lihat [CodeBuild harga](#), [harga Amazon S3](#), [AWS Key Management Service harga](#), dan [harga Amazon CloudWatch](#).

Topik

- [Menjalankan sampel](#)
- [Struktur direktori](#)
- [Berkas](#)
- [Sumber daya terkait](#)

Menjalankan sampel

Untuk menjalankan sampel ini

1. Buat file seperti yang dijelaskan di bagian “Struktur direktori” dan “File” dari topik ini, lalu unggah ke bucket input S3 atau AWS CodeCommit, GitHub, atau repositori Bitbucket.

Important

Jangan mengunggah(*root directory name*), hanya file di dalamnya(*root directory name*).

Jika Anda menggunakan bucket input S3, pastikan untuk membuat file ZIP yang berisi file, lalu unggah ke bucket input. Jangan tambahkan (*root directory name*) ke file ZIP, hanya file di dalamnya(*root directory name*).

2. Buat proyek build, jalankan build, dan lihat informasi build terkait.

Jika Anda menggunakan AWS CLI untuk membuat proyek build, input berformat JSON ke `create-project` perintah mungkin terlihat mirip dengan ini. (Ganti placeholder dengan nilai Anda sendiri.)

```
{
  "name": "sample-docker-custom-image-project",
  "source": {
    "type": "S3",
    "location": "codebuild-region-ID-account-ID-input-
bucket/DockerCustomImageSample.zip"
  },
  "artifacts": {
    "type": "NO_ARTIFACTS"
  },
  "environment": {
    "type": "LINUX_CONTAINER",
    "image": "docker:dind",
    "computeType": "BUILD_GENERAL1_SMALL",
    "privilegedMode": false
  },
  "serviceRole": "arn:aws:iam::account-ID:role/role-name",
  "encryptionKey": "arn:aws:kms:region-ID:account-ID:key/key-ID"
}
```

Note

Secara default, daemon Docker diaktifkan untuk build non-VPC. Jika Anda ingin menggunakan kontainer Docker untuk build VPC, [lihat Runtime Privilege dan Kemampuan Linux di situs web Docker Docs dan](#) aktifkan mode istimewa. Juga, Windows tidak mendukung mode istimewa.

- Untuk melihat hasil build, lihat di log build untuk `stringHello, World!`. Untuk informasi selengkapnya, lihat [Tampilkan detail build](#).

Struktur direktori

Sampel ini mengasumsikan struktur direktori ini.

```
(root directory name)
### buildspec.yml
### Dockerfile
```

Berkas

Gambar dasar dari sistem operasi yang digunakan dalam sampel ini adalah Ubuntu. Sampel menggunakan file-file ini.

`buildspec.yml`(dalam(*root directory name*))

```
version: 0.2

phases:
  pre_build:
    commands:
      - docker build -t helloworld .
  build:
    commands:
      - docker images
      - docker run helloworld echo "Hello, World!"
```

`Dockerfile`(dalam(*root directory name*))

```
FROM maven:3.3.9-jdk-8
```

```
RUN echo "Hello World"
```

Sumber daya terkait

- Untuk informasi tentang memulai AWS CodeBuild, lihat [Memulai dengan AWS CodeBuild menggunakan konsol](#).
- Untuk informasi tentang pemecahan masalah di CodeBuild, lihat. [Pemecahan masalah AWS CodeBuild](#)
- Untuk informasi tentang kuota di CodeBuild, lihat [Kuota untuk AWS CodeBuild](#).

Publikasikan gambar Docker ke sampel repositori gambar Amazon Elastic Container Registry untuk CodeBuild

Sampel ini menghasilkan image Docker sebagai output build dan kemudian mendorong image Docker ke repositori image Amazon Elastic Container Registry (Amazon ECR). Anda dapat mengadaptasi sampel ini untuk mendorong image Docker ke Docker Hub. Untuk informasi selengkapnya, lihat [Mengadaptasi sampel untuk mendorong gambar ke Docker Hub](#).

Untuk mempelajari cara membuat image Docker dengan menggunakan image build Docker kustom (`docker:dind` Docker Hub), lihat. [Docker dalam sampel gambar khusus](#)

Sampel ini diuji referensigolang:1.12.

Sampel ini menggunakan fitur build Docker multi-tahap baru, yang menghasilkan image Docker sebagai output build. Kemudian mendorong gambar Docker ke repositori gambar Amazon ECR. Pembuatan gambar Docker multi-tahap membantu mengurangi ukuran gambar Docker akhir. Untuk informasi selengkapnya, lihat [Menggunakan build multi-tahap dengan Docker](#).

Important

Menjalankan sampel ini dapat mengakibatkan biaya ke AWS akun Anda. Ini termasuk kemungkinan biaya untuk AWS CodeBuild dan untuk AWS sumber daya dan tindakan yang terkait dengan Amazon S3, CloudWatch Log AWS KMS, dan Amazon ECR. Untuk informasi selengkapnya, lihat [CodeBuild harga](#), [harga Amazon S3](#), [harga, AWS Key Management Service harga](#) Amazon, dan [CloudWatch harga Amazon Elastic Container Registry](#).

Topik

- [Menjalankan sampel](#)
- [Struktur direktori](#)
- [Berkas](#)
- [Mengadaptasi sampel untuk mendorong gambar ke Docker Hub](#)
- [Sumber daya terkait](#)

Menjalankan sampel

Untuk menjalankan sampel ini

1. Jika Anda sudah memiliki repositori gambar di Amazon ECR yang ingin Anda gunakan, lewati ke langkah 3. Jika tidak, jika Anda menggunakan pengguna alih-alih akun AWS root atau pengguna administrator untuk bekerja dengan Amazon ECR, tambahkan pernyataan ini (antara **### MULAI MENAMBAHKAN PERNYATAAN DI SINI ###** dan **### AKHIRI MENAMBAHKAN PERNYATAAN DI SINI ###**) ke pengguna (atau grup IAM yang dikaitkan dengan pengguna). Menggunakan akun AWS root tidak dianjurkan. Pernyataan ini memungkinkan pembuatan repositori Amazon ECR untuk menyimpan gambar Docker. Elipses (. . .) digunakan untuk singkatnya dan untuk membantu Anda menemukan tempat untuk menambahkan pernyataan. Jangan menghapus pernyataan apa pun, dan jangan ketik elips ini ke dalam kebijakan. Untuk informasi selengkapnya, lihat [Bekerja dengan kebijakan sebaris menggunakan AWS Management Console](#) dalam Panduan pengguna.

```
{
  "Statement": [
    ### BEGIN ADDING STATEMENT HERE ###
    {
      "Action": [
        "ecr:CreateRepository"
      ],
      "Resource": "*",
      "Effect": "Allow"
    },
    ### END ADDING STATEMENT HERE ###
    ...
  ],
  "Version": "2012-10-17"
}
```

Note

Entitas IAM yang memodifikasi kebijakan ini harus memiliki izin di IAM untuk memodifikasi kebijakan.

2. Buat repositori gambar di Amazon ECR. Pastikan untuk membuat repositori di AWS Wilayah yang sama tempat Anda membuat lingkungan build dan menjalankan build. Untuk informasi selengkapnya, lihat [Membuat repositori](#) di Panduan Pengguna Amazon ECR. Nama repositori ini harus cocok dengan nama repositori yang Anda tentukan nanti dalam prosedur ini, yang diwakili oleh variabel lingkungan. `IMAGE_REPO_NAME` Pastikan kebijakan repositori Amazon ECR memberikan akses push image untuk peran IAM layanan Anda CodeBuild .
3. Tambahkan pernyataan ini (antara **### MULAI MENAMBAHKAN PERNYATAAN DI SINI ###** dan **### AKHIR MENAMBAHKAN PERNYATAAN DI SINI ###**) ke kebijakan yang Anda lampirkan ke peran layanan Anda. AWS CodeBuild Pernyataan ini memungkinkan CodeBuild untuk mengunggah gambar Docker ke repositori Amazon ECR. Elipses (. . .) digunakan untuk singkatnya dan untuk membantu Anda menemukan tempat untuk menambahkan pernyataan. Jangan menghapus pernyataan apa pun, dan jangan ketik elips ini ke dalam kebijakan.

```
{
  "Statement": [
    ### BEGIN ADDING STATEMENT HERE ###
    {
      "Action": [
        "ecr:BatchCheckLayerAvailability",
        "ecr:CompleteLayerUpload",
        "ecr:GetAuthorizationToken",
        "ecr:InitiateLayerUpload",
        "ecr:PutImage",
        "ecr:UploadLayerPart"
      ],
      "Resource": "*",
      "Effect": "Allow"
    },
    ### END ADDING STATEMENT HERE ###
    ...
  ],
  "Version": "2012-10-17"
}
```


Note

Entitas IAM yang memodifikasi kebijakan ini harus memiliki izin di IAM untuk memodifikasi kebijakan.

4. Buat file seperti yang dijelaskan di bagian “Struktur direktori” dan “File” dari topik ini, lalu unggah ke bucket input S3 atau AWS CodeCommit, GitHub, atau repositori Bitbucket. Untuk informasi selengkapnya, lihat [Referensi file definisi gambar](#) di Panduan AWS CodePipeline Pengguna.

Important

Jangan mengunggah(*root directory name*), hanya file di dalamnya(*root directory name*).

Jika Anda menggunakan bucket input S3, pastikan untuk membuat file ZIP yang berisi file, lalu unggah ke bucket input. Jangan tambahkan (*root directory name*) ke file ZIP, hanya file di dalamnya(*root directory name*).

5. Buat proyek build, jalankan build, dan lihat informasi build.

Jika Anda menggunakan konsol untuk membuat proyek Anda:

- a. Untuk Sistem operasi, pilih Ubuntu.
- b. Untuk Waktu aktif, pilih Standar.
- c. Untuk Gambar, pilih aws/codebuild/standard:5.0.
- d. Tambahkan variabel lingkungan berikut:
 - *AWS_DEFAULT_REGION* dengan nilai *Region-ID*
 - *AWS_ACCOUNT_ID* dengan nilai *Account-ID*
 - *IMAGE_TAG* dengan nilai Terbaru
 - *IMAGE_REPO_NAME* dengan nilai *Amazon-Ecr-repo-name*

Jika Anda menggunakan AWS CLI untuk membuat proyek build, input berformat JSON ke `create-project` perintah mungkin terlihat mirip dengan ini. (Ganti placeholder dengan nilai Anda sendiri.)

```
{
```

```
"name": "sample-docker-project",
"source": {
  "type": "S3",
  "location": "codebuild-region-ID-account-ID-input-bucket/DockerSample.zip"
},
"artifacts": {
  "type": "NO_ARTIFACTS"
},
"environment": {
  "type": "LINUX_CONTAINER",
  "image": "aws/codebuild/standard:5.0",
  "computeType": "BUILD_GENERAL1_SMALL",
  "environmentVariables": [
    {
      "name": "AWS_DEFAULT_REGION",
      "value": "region-ID"
    },
    {
      "name": "AWS_ACCOUNT_ID",
      "value": "account-ID"
    },
    {
      "name": "IMAGE_REPO_NAME",
      "value": "Amazon-ECR-repo-name"
    },
    {
      "name": "IMAGE_TAG",
      "value": "latest"
    }
  ],
},
"serviceRole": "arn:aws:iam::account-ID:role/role-name",
"encryptionKey": "arn:aws:kms:region-ID:account-ID:key/key-ID"
}
```

6. Konfirmasikan bahwa CodeBuild berhasil mendorong image Docker ke repositori:

1. Buka konsol Amazon ECR di <https://console.aws.amazon.com/ecr/>.
2. Pilih nama repositori. Gambar harus dicantumkan di kolom tag Gambar.

Struktur direktori

Sampel ini mengasumsikan struktur direktori ini.

(root directory name)

```
### buildspec.yml
### Dockerfile
```

Berkas

Sampel ini menggunakan file-file ini.

buildspec.yml(dalam(*root directory name*))

```
version: 0.2

phases:
  pre_build:
    commands:
      - echo Logging in to Amazon ECR...
      - aws ecr get-login-password --region $AWS_DEFAULT_REGION | docker login --
username AWS --password-stdin $AWS_ACCOUNT_ID.dkr.ecr.$AWS_DEFAULT_REGION.amazonaws.com
  build:
    commands:
      - echo Build started on `date`
      - echo Building the Docker image...
      - docker build -t $IMAGE_REPO_NAME:$IMAGE_TAG .
      - docker tag $IMAGE_REPO_NAME:$IMAGE_TAG $AWS_ACCOUNT_ID.dkr.ecr.
$AWS_DEFAULT_REGION.amazonaws.com/$IMAGE_REPO_NAME:$IMAGE_TAG
  post_build:
    commands:
      - echo Build completed on `date`
      - echo Pushing the Docker image...
      - docker push $AWS_ACCOUNT_ID.dkr.ecr.$AWS_DEFAULT_REGION.amazonaws.com/
$IMAGE_REPO_NAME:$IMAGE_TAG
```

Dockerfile(dalam(*root directory name*))

```
FROM golang:1.12-alpine AS build
#Install git
RUN apk add --no-cache git
#Get the hello world package from a GitHub repository
RUN go get github.com/golang/example/hello
WORKDIR /go/src/github.com/golang/example/hello
# Build the project and send the output to /bin/HelloWorld
RUN go build -o /bin/HelloWorld
```

```
FROM golang:1.12-alpine
#Copy the build's output binary from the previous build container
COPY --from=build /bin/HelloWorld /bin/HelloWorld
ENTRYPOINT ["/bin/HelloWorld"]
```

Note

CodeBuild mengganti ENTRYPOINT untuk gambar Docker khusus.

Mengadaptasi sampel untuk mendorong gambar ke Docker Hub

Untuk mendorong gambar Docker ke Docker Hub alih-alih Amazon ECR, edit kode sampel ini.

Note

Jika Anda menggunakan versi Docker lebih awal dari 17.06, hapus opsi. `--no-include-email`

1. Ganti baris kode khusus Amazon ECR ini dalam file: `buildspec.yml`

```
...
pre_build:
  commands:
    - echo Logging in to Amazon ECR...
    - aws ecr get-login-password --region $AWS_DEFAULT_REGION |
docker login --username AWS --password-stdin $AWS_ACCOUNT_ID.dkr.ecr.
$AWS_DEFAULT_REGION.amazonaws.com
build:
  commands:
    - echo Build started on `date`
    - echo Building the Docker image...
    - docker build -t $IMAGE_REPO_NAME:$IMAGE_TAG .
    - docker tag $IMAGE_REPO_NAME:$IMAGE_TAG $AWS_ACCOUNT_ID.dkr.ecr.
$AWS_DEFAULT_REGION.amazonaws.com/$IMAGE_REPO_NAME:$IMAGE_TAG
post_build:
  commands:
    - echo Build completed on `date`
    - echo Pushing the Docker image...
```

```

- docker push $AWS_ACCOUNT_ID.dkr.ecr.$AWS_DEFAULT_REGION.amazonaws.com/
$image_repo_name:$image_tag
...

```

Dengan baris kode khusus Docker Hub ini:

```

...
pre_build:
  commands:
    - echo Logging in to Docker Hub...
    # Type the command to log in to your Docker Hub account here.
build:
  commands:
    - echo Build started on `date`
    - echo Building the Docker image...
    - docker build -t $image_repo_name:$image_tag .
    - docker tag $image_repo_name:$image_tag $image_repo_name:$image_tag
post_build:
  commands:
    - echo Build completed on `date`
    - echo Pushing the Docker image...
    - docker push $image_repo_name:$image_tag
...

```

2. Unggah kode yang diedit ke bucket input S3 atau repositori AWS CodeCommit, GitHub, atau Bitbucket.

Important

Jangan mengunggah(*root directory name*), hanya file di dalamnya(*root directory name*).

Jika Anda menggunakan bucket input S3, pastikan untuk membuat file ZIP yang berisi file, lalu unggah ke bucket input. Jangan tambahkan (*root directory name*) ke file ZIP, hanya file di dalamnya(*root directory name*).

3. Ganti baris kode ini dari input berformat JSON ke perintah: `create-project`

```

...
"environmentVariables": [
  {
    "name": "AWS_DEFAULT_REGION",

```

```
    "value": "region-ID"
  },
  {
    "name": "AWS_ACCOUNT_ID",
    "value": "account-ID"
  },
  {
    "name": "IMAGE_REPO_NAME",
    "value": "Amazon-ECR-repo-name"
  },
  {
    "name": "IMAGE_TAG",
    "value": "latest"
  }
]
...

```

Dengan baris kode ini:

```
...
"environmentVariables": [
  {
    "name": "IMAGE_REPO_NAME",
    "value": "your-Docker-Hub-repo-name"
  },
  {
    "name": "IMAGE_TAG",
    "value": "latest"
  }
]
...

```

4. Buat lingkungan build, jalankan build, dan lihat informasi build terkait.
5. Konfirmasikan bahwa AWS CodeBuild berhasil mendorong image Docker ke repositori. Masuk ke Docker Hub, buka repositori, dan pilih tab Tag. `latest` Tag harus berisi nilai Terakhir Diperbarui yang sangat baru.

Sumber daya terkait

- Untuk informasi tentang memulai AWS CodeBuild, lihat [Memulai dengan AWS CodeBuild menggunakan konsol](#).

- Untuk informasi tentang pemecahan masalah di CodeBuild, lihat. [Pemecahan masalah AWS CodeBuild](#)
- Untuk informasi tentang kuota di CodeBuild, lihat [Kuota untuk AWS CodeBuild](#).

Registri pribadi dengan AWS Secrets Manager sampel untuk CodeBuild

Contoh ini menunjukkan cara menggunakan image Docker yang disimpan dalam registri pribadi sebagai lingkungan AWS CodeBuild runtime Anda. Kredensi untuk registri pribadi disimpan di. AWS Secrets Manager Registri pribadi apa pun berfungsi dengan CodeBuild. Contoh ini menggunakan Docker Hub.

Note

Rahasia terlihat oleh tindakan dan tidak disamarkan saat ditulis ke file.

Persyaratan sampel registri pribadi

Untuk menggunakan registri pribadi dengan AWS CodeBuild, Anda harus memiliki yang berikut:

- Rahasia Secrets Manager yang menyimpan kredensial Docker Hub Anda. Kredensialnya digunakan untuk mengakses repositori pribadi Anda.

Note

Anda akan dikenakan biaya untuk rahasia yang Anda buat.

- Repositori atau akun pribadi.
- Kebijakan IAM peran CodeBuild layanan yang memberikan akses ke rahasia Secrets Manager Anda.

Ikuti langkah-langkah ini untuk membuat sumber daya ini dan kemudian membuat proyek CodeBuild build menggunakan gambar Docker yang disimpan di registri pribadi Anda.

Buat CodeBuild proyek dengan registri pribadi

1. Untuk informasi tentang cara membuat repositori pribadi gratis, lihat [Repositori](#) di Docker Hub. Anda juga dapat menjalankan perintah berikut di terminal untuk menarik gambar, mendapatkan ID-nya, dan mendorongnya ke repositori baru.

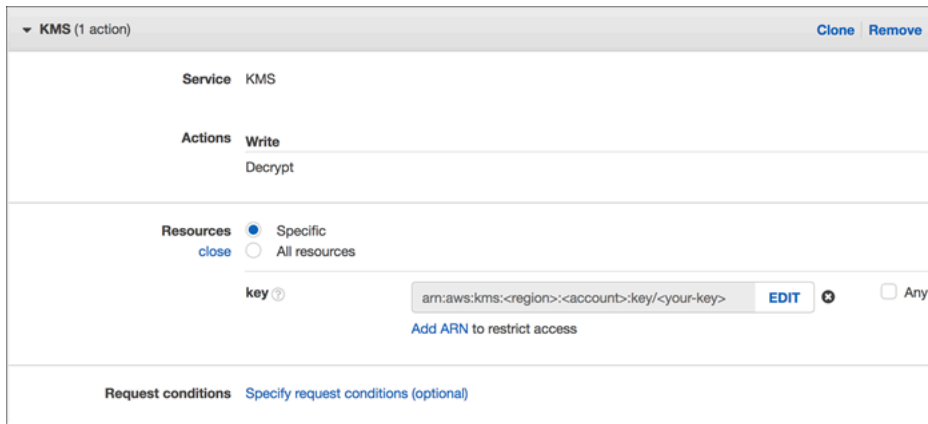
```
docker pull amazonlinux
docker images amazonlinux --format {{.ID}}
docker tag image-id your-username/repository-name:tag
docker login
docker push your-username/repository-name
```

2. Ikuti langkah-langkah di [Buat AWS Secrets Manager rahasia](#) di Panduan AWS Secrets Manager Pengguna.
 - a. Pada langkah 3, di Pilih tipe rahasia, pilih Jenis rahasia lainnya.
 - b. Pada pasangan kunci/nilai, buat satu pasangan nilai kunci untuk nama pengguna Docker Hub Anda dan satu pasangan nilai kunci untuk kata sandi Docker Hub Anda.
 - c. Lanjutkan mengikuti langkah-langkah di [Buat AWS Secrets Manager rahasia](#).
 - d. Pada langkah 5, pada halaman Konfigurasi rotasi otomatis, matikan karena kunci sesuai dengan kredensial Docker Hub Anda.
 - e. Selesai mengikuti langkah-langkah di [Buat AWS Secrets Manager rahasia](#).

Lihat informasi yang lebih lengkap di [Apakah AWS Secrets Manager?](#)

3. Saat Anda membuat AWS CodeBuild proyek di konsol, CodeBuild lampirkan izin yang diperlukan untuk Anda. Jika Anda menggunakan AWS KMS kunci selain `DefaultEncryptionKey`, Anda harus menambahkannya ke peran layanan. Untuk informasi selengkapnya, lihat [Memodifikasi peran \(konsol\)](#) di Panduan Pengguna IAM.

Agar peran layanan Anda berfungsi dengan Secrets Manager, setidaknya harus memiliki `secretsmanager:GetSecretValue` izin.



4. Untuk menggunakan konsol untuk membuat proyek dengan lingkungan yang disimpan dalam registri pribadi, lakukan hal berikut saat Anda membuat proyek. Untuk informasi, lihat [Buat proyek build \(konsol\)](#).

Note

Jika registri pribadi Anda ada di VPC Anda, itu harus memiliki akses internet publik. CodeBuild tidak dapat menarik gambar dari alamat IP pribadi di VPC.

- Dalam gambar Lingkungan, pilih Gambar kustom.
- Untuk jenis Lingkungan, pilih Linux atau Windows.
- Untuk registri Gambar, pilih Registri lain.
- Di URL registri Eksternal, masukkan lokasi gambar dan kredensi Registri - opsional masukkan ARN atau nama kredensial Secrets Manager Anda.

Note

Jika kredensial Anda tidak ada di Wilayah Anda saat ini, maka Anda harus menggunakan ARN. Anda tidak dapat menggunakan nama kredensial jika kredensialnya ada di Wilayah yang berbeda.

Buat situs web statis dengan keluaran build yang dihosting di bucket S3

Anda dapat menonaktifkan enkripsi artefak dalam build. Anda mungkin ingin melakukan ini sehingga Anda dapat mempublikasikan artefak ke lokasi yang dikonfigurasi untuk meng-host situs web. (Anda

tidak dapat mempublikasikan artefak terenkripsi.) Contoh ini menunjukkan bagaimana Anda dapat menggunakan webhook untuk memicu build dan mempublikasikan artefaknya ke bucket S3 yang dikonfigurasi untuk menjadi situs web.

1. Ikuti petunjuk dalam [Menyiapkan situs web statis](#) untuk mengonfigurasi bucket S3 agar berfungsi seperti situs web.
2. Buka AWS CodeBuild konsol di <https://console.aws.amazon.com/codesuite/codebuild/home>.
3. Jika halaman CodeBuild informasi ditampilkan, pilih Buat proyek build. Jika tidak, pada panel navigasi, perluas Build, pilih Build projects, lalu pilih Create build project.
4. Di Nama proyek, masukkan nama untuk proyek pembangunan ini. Membangun nama proyek harus unik di setiap AWS akun. Anda juga dapat menyertakan deskripsi opsional proyek build untuk membantu pengguna lain memahami tujuan proyek ini.
5. Di Sumber, untuk penyedia Sumber, pilih GitHub. Ikuti petunjuk untuk menghubungkan (atau menyambung kembali) dengan GitHub, dan kemudian pilih Otorisasi.

Untuk Webhook, pilih Rebuild setiap kali perubahan kode didorong ke repositori ini. Anda dapat memilih kotak centang ini hanya jika Anda memilih Gunakan repositori di akun saya.

Source Add source

Source 1 - Primary

Source provider
GitHub

Repository
 Public repository Repository in my GitHub account

GitHub repository
 Refresh

Disconnect GitHub account

▼ **Additional configuration**

Git clone depth

Git clone depth - *optional*

Build Status - *optional*
 Report build statuses to source provider when your builds start and finish

Webhook - *optional*
 Rebuild every time a code change is pushed to this repository

Branch filter - *optional*


Enter a regular expression

6. Di Lingkungan:

Untuk citra Lingkungan, lakukan salah satu hal berikut:

- Untuk menggunakan image Docker yang dikelola oleh AWS CodeBuild, pilih Managed image, lalu buat pilihan dari Sistem operasi, Runtime, Image, dan versi Image. Buat pilihan dari jenis Lingkungan jika tersedia.

- Untuk menggunakan gambar Docker lain, pilih Custom image. Untuk jenis Lingkungan, pilih ARM, Linux, GPU Linux, atau Windows. Jika Anda memilih Registri lain, untuk URL registri eksternal, masukkan nama dan tag gambar Docker di Docker Hub, menggunakan format. *docker repository/docker image name* Jika Anda memilih Amazon ECR, gunakan repositori Amazon ECR dan gambar Amazon ECR untuk memilih gambar Docker di akun Anda. AWS
 - Untuk menggunakan image Docker pribadi, pilih Custom image. Untuk jenis Lingkungan, pilih ARM, Linux, GPU Linux, atau Windows. Untuk registri Gambar, pilih Registri lain, lalu masukkan ARN kredensial untuk gambar Docker pribadi Anda. Kredensialnya harus dibuat oleh Secrets Manager. Untuk informasi lebih lanjut, lihat [Apa itu AWS Secrets Manager?](#) di Panduan Pengguna AWS Secrets Manager .
7. Dalam peran Layanan, lakukan salah satu hal berikut:
- Jika Anda tidak memiliki peran CodeBuild layanan, pilih Peran layanan baru. Di Nama peran, masukkan nama untuk peran baru.
 - Jika Anda memiliki peran CodeBuild layanan, pilih Peran layanan yang ada. Di Peran ARN, pilih peran layanan.

 Note

Saat menggunakan konsol untuk membuat atau memperbarui proyek build, Anda dapat membuat peran CodeBuild layanan secara bersamaan. Secara default, peran hanya berfungsi dengan proyek build tersebut. Jika Anda menggunakan konsol untuk mengaitkan peran layanan ini dengan proyek build lain, peran akan diperbarui agar berfungsi dengan proyek build lainnya. Peran layanan dapat bekerja dengan hingga 10 proyek pembangunan.

8. Di Buildspec, lakukan salah satu hal berikut:
- Pilih Gunakan file buildspec untuk menggunakan file buildspec.yml di direktori root kode sumber.
 - Pilih Sisipkan perintah build untuk menggunakan konsol untuk menyisipkan perintah build.

Lihat informasi yang lebih lengkap di [Referensi Buildspec](#).

9. Di Artefak, untuk Jenis, pilih Amazon S3 untuk menyimpan output build dalam bucket S3.

10. Untuk nama Bucket, pilih nama bucket S3 yang Anda konfigurasi agar berfungsi sebagai situs web pada langkah 1.
11. Jika Anda memilih Insert perintah build di Environment, lalu untuk file Output, masukkan lokasi file dari build yang ingin Anda masukkan ke dalam bucket keluaran. Jika Anda memiliki lebih dari satu lokasi, gunakan koma untuk memisahkan setiap lokasi (misalnya, **appspect.yml, target/my-app.jar**). Untuk informasi selengkapnya, lihat [Artifacts reference-key in the buildspec file](#).
12. Pilih Nonaktifkan enkripsi artefak.
13. Perluas Konfigurasi tambahan dan pilih opsi yang sesuai.
14. Pilih Buat proyek build. Pada halaman proyek build, dalam riwayat Build, pilih Start build untuk menjalankan build.
15. (Opsional) Ikuti petunjuk di [Contoh: Percepat situs web Anda dengan Amazon CloudFront](#) di Panduan Pengembang Amazon S3.

Beberapa sumber input dan sampel artefak keluaran

Anda dapat membuat proyek AWS CodeBuild build dengan lebih dari satu sumber input dan lebih dari satu set artefak keluaran. Contoh ini menunjukkan cara menyiapkan proyek build yang:

- Menggunakan berbagai sumber dan repositori dari berbagai jenis.
- Menerbitkan artefak build ke beberapa bucket S3 dalam satu build.

Dalam contoh ini, Anda membuat proyek build dan menggunakannya untuk menjalankan build. Sampel menggunakan file buildspec proyek build untuk menunjukkan cara menggabungkan lebih dari satu sumber dan membuat lebih dari satu set artefak.

1. Unggah sumber Anda ke satu atau beberapa bucket S3,, Server GitHub Perusahaan CodeCommit GitHub, atau repositori Bitbucket.
2. Pilih sumber mana yang menjadi sumber utama. Ini adalah sumber di mana CodeBuild mencari dan menjalankan file buildspec Anda.
3. Buat proyek build. Untuk informasi selengkapnya, lihat [Membuat proyek pembangunan diAWS CodeBuild](#).
4. Buat proyek build Anda, jalankan build, dan dapatkan informasi tentang build.
5. Jika Anda menggunakan AWS CLI untuk membuat proyek build, input berformat JSON ke `create-project` perintah mungkin terlihat mirip dengan yang berikut:

```
{
  "name": "sample-project",
  "source": {
    "type": "S3",
    "location": "<bucket/sample.zip>"
  },
  "secondarySources": [
    {
      "type": "CODECOMMIT",
      "location": "https://git-codecommit.us-west-2.amazonaws.com/v1/repos/repo",
      "sourceIdentifier": "source1"
    },
    {
      "type": "GITHUB",
      "location": "https://github.com/awslabs/aws-codebuild-jenkins-plugin",
      "sourceIdentifier": "source2"
    }
  ],
  "secondaryArtifacts": [ss
    {
      "type": "S3",
      "location": "<output-bucket>",
      "artifactIdentifier": "artifact1"
    },
    {
      "type": "S3",
      "location": "<other-output-bucket>",
      "artifactIdentifier": "artifact2"
    }
  ],
  "environment": {
    "type": "LINUX_CONTAINER",
    "image": "aws/codebuild/standard:5.0",
    "computeType": "BUILD_GENERAL1_SMALL"
  },
  "serviceRole": "arn:aws:iam::account-ID:role/role-name",
  "encryptionKey": "arn:aws:kms:region-ID:account-ID:key/key-ID"
}
```

Sumber utama Anda didefinisikan di bawah `source` atribut. Semua sumber lain disebut sumber sekunder dan muncul di bawah `secondarySources`. Semua sumber sekunder

dipasang di direktori mereka sendiri. Direktori ini disimpan dalam variabel lingkungan bawaan `CODEBUILD_SRC_DIR_sourceIdentifier`. Untuk informasi selengkapnya, lihat [Variabel lingkungan di lingkungan build](#).

`secondaryArtifacts` atribut berisi daftar definisi artefak. Artefak ini menggunakan `secondary-artifacts` blok file `buildspec` yang bersarang di dalam blok `artifacts`

Artefak sekunder dalam file `buildspec` memiliki struktur yang sama dengan artefak dan dipisahkan oleh pengenal artefaknya.

Note

Di [CodeBuild API](#), artefak sekunder `artifactIdentifier` pada adalah atribut wajib di `CreateProject` dan `UpdateProject`. Ini harus digunakan untuk referensi artefak sekunder.

Menggunakan input berformat JSON sebelumnya, file `buildspec` untuk proyek mungkin terlihat seperti:

```
version: 0.2

phases:
  install:
    runtime-versions:
      java: openjdk11
  build:
    commands:
      - cd $CODEBUILD_SRC_DIR_source1
      - touch file1
      - cd $CODEBUILD_SRC_DIR_source2
      - touch file2

artifacts:
  files:
    - '**.*'
  secondary-artifacts:
    artifact1:
      base-directory: $CODEBUILD_SRC_DIR_source1
      files:
        - file1
    artifact2:
```

```
base-directory: $CODEBUILD_SRC_DIR_source2
files:
  - file2
```

Anda dapat mengganti versi sumber utama menggunakan API dengan `sourceVersion` atribut di `startBuild`. Untuk mengganti satu atau beberapa versi sumber sekunder, gunakan `secondarySourceVersionOverride` atribut.

Input berformat JSON ke `start-build` perintah di mungkin terlihat seperti: AWS CLI

```
{
  "projectName": "sample-project",
  "secondarySourcesVersionOverride": [
    {
      "sourceIdentifier": "source1",
      "sourceVersion": "codecommit-branch"
    },
    {
      "sourceIdentifier": "source2",
      "sourceVersion": "github-branch"
    }
  ]
}
```

Proyek tanpa sampel sumber

Anda dapat mengonfigurasi CodeBuild proyek dengan memilih jenis **NO_SOURCE** sumber saat Anda mengonfigurasi sumber Anda. Ketika jenis sumber Anda **NO_SOURCE**, Anda tidak dapat menentukan file `buildspec` karena proyek Anda tidak memiliki sumber. Sebagai gantinya, Anda harus menentukan string `buildspec` yang diformat YAML dalam `buildspec` atribut input berformat JSON ke perintah CLI. `create-project` Mungkin terlihat seperti ini:

```
{
  "name": "project-name",
  "source": {
    "type": "NO_SOURCE",
    "buildspec": "version: 0.2\n\nphases:\n  build:\n    commands:\n      - command"
  },
  "environment": {
    "type": "LINUX_CONTAINER",
    "image": "aws/codebuild/standard:5.0",
```



```
"computeType": "BUILD_GENERAL1_SMALL",
},
"serviceRole": "arn:aws:iam::account-ID:role/role-name",
"encryptionKey": "arn:aws:kms:region-ID:account-ID:key/key-ID"
}
```

Untuk informasi selengkapnya, lihat [Buat proyek build \(AWS CLI\)](#).

Untuk mempelajari cara membuat pipeline yang menggunakan beberapa input sumber CodeBuild untuk membuat beberapa artefak keluaran, lihat [AWS CodePipeline integrasi dengan CodeBuild dan beberapa sumber input dan sampel artefak keluaran](#)

Versi runtime dalam contoh file buildspec untuk CodeBuild

Jika Anda menggunakan gambar standar Amazon Linux 2 (AL2) versi 1.0 atau yang lebih baru, atau gambar standar Ubuntu versi 2.0 atau yang lebih baru, Anda dapat menentukan satu atau beberapa runtime di `runtime-versions` bagian file buildspec Anda. Contoh ini menunjukkan bagaimana Anda dapat mengubah runtime project, menentukan lebih dari satu runtime, dan menentukan runtime yang bergantung pada runtime lain. Untuk informasi tentang runtime yang didukung, lihat [Gambar Docker disediakan oleh CodeBuild](#).

Note

Jika Anda menggunakan Docker di container build, build Anda harus berjalan dalam mode istimewa. Lihat informasi yang lebih lengkap di [Jalankan build diAWS CodeBuild](#) dan [Membuat proyek pembangunan diAWS CodeBuild](#).

Perbarui versi runtime Anda

Anda dapat mengubah runtime yang digunakan oleh proyek Anda ke versi baru dengan memperbarui `runtime-versions` bagian file buildspec Anda. Contoh berikut menunjukkan cara menentukan java versi 8 dan 11.

- `runtime-versions` Bagian yang menentukan versi 8 dari Java:

```
phases:
  install:
    runtime-versions:
      java: corretto8
```

- `runtime-versions` Bagian yang menentukan versi 11 dari Java:

```
phases:
  install:
    runtime-versions:
      java: corretto11
```

Contoh berikut menunjukkan cara menentukan versi Python yang berbeda menggunakan gambar standar Ubuntu 5.0 atau gambar standar Amazon Linux 2 3.0:

- `runtime-versions` Bagian yang menentukan Python versi 3.7:

```
phases:
  install:
    runtime-versions:
      python: 3.7
```

- `runtime-versions` Bagian yang menentukan Python versi 3.8:

```
phases:
  install:
    runtime-versions:
      python: 3.8
```

Contoh ini menunjukkan proyek yang dimulai dengan runtime Java versi 8, dan kemudian diperbarui ke runtime Java versi 10.

1. Unduh dan instal Maven. Untuk informasi, lihat [Mengunduh Apache Maven dan Menginstal Apache Maven di situs web Apache Maven](#).
2. Beralih ke direktori kosong di komputer atau instance lokal Anda, lalu jalankan perintah Maven ini.

```
mvn archetype:generate "-DgroupId=com.mycompany.app" "-DartifactId=ROOT" "-DarchetypeArtifactId=maven-archetype-webapp" "-DinteractiveMode=false"
```

Jika berhasil, struktur direktori dan file ini dibuat.

```
.
```

```

### ROOT
  ### pom.xml
  ### src
    ### main
      ### resources
      ### webapp
        ### WEB-INF
        #   ### web.xml
        ### index.jsp

```

3. Buat file bernama `buildspec.yml` dengan isi berikut ini. Simpan file di *(root directory name)*/my-web-app direktori.

```

version: 0.2

phases:
  install:
    runtime-versions:
      java: corretto8
  build:
    commands:
      - java -version
      - mvn package
artifacts:
  files:
    - '**/*'
  base-directory: 'target/my-web-app'

```

Dalam file `buildspec`:

- `runtime-versions` Bagian ini menentukan bahwa proyek menggunakan versi 8 dari runtime Java.
- `- java -version` Perintah menampilkan versi Java yang digunakan oleh proyek Anda saat dibangun.

Struktur file Anda sekarang akan terlihat seperti ini.

```

(root directory name)
### my-web-app
  ### src
  #   ### main

```

```
#   ### resources
#   ### webapp
#       ### WEB-INF
#           ### web.xml
#               ### index.jsp
### buildspec.yml
### pom.xml
```

4. Unggah isi my-web-app direktori ke bucket input S3 atau repositori CodeCommit, GitHub, atau Bitbucket.

Important

Jangan mengunggah (*root directory name*) atau (*root directory name*)/my-web-app, hanya direktori dan file di (*root directory name*)/my-web-app.

Jika Anda menggunakan bucket input S3, pastikan untuk membuat file ZIP yang berisi struktur direktori dan file, lalu unggah ke bucket input. Jangan menambahkan (*root directory name*) atau (*root directory name*)/my-web-app ke file ZIP, hanya direktori dan file di (*root directory name*)/my-web-app.

5. Buka AWS CodeBuild konsol di <https://console.aws.amazon.com/codesuite/codebuild/home>.
6. Buat proyek build. Lihat informasi yang lebih lengkap di [Buat proyek build \(konsol\)](#) dan [Menjalankan build \(konsol\)](#). Biarkan semua pengaturan pada nilai default mereka, kecuali untuk pengaturan ini.
 - Untuk Lingkungan:
 - Untuk gambar Lingkungan, pilih Gambar terkelola.
 - Untuk sistem operasi, pilih Amazon Linux 2.
 - Untuk Runtime, pilih Standar.
 - Untuk Gambar, pilih aws/codebuild/amazonlinux2-x86_64-standard:4.0.
7. Pilih Mulai membangun.
8. Pada konfigurasi Build, terima defaultnya, lalu pilih Start build.
9. Setelah build selesai, lihat output build di tab Build logs. Anda akan melihat output yang serupa dengan yang berikut:

```
[Container] Date Time Phase is DOWNLOAD_SOURCE
[Container] Date Time CODEBUILD_SRC_DIR=/codebuild/output/src460614277/src
```

```
[Container] Date Time YAML location is /codebuild/output/src460614277/src/buildspec.yml
[Container] Date Time Processing environment variables
[Container] Date Time Selecting 'java' runtime version 'corretto8' based on manual selections...
[Container] Date Time Running command echo "Installing Java version 8 ..."
Installing Java version 8 ...

[Container] Date Time Running command export JAVA_HOME="$JAVA_8_HOME"

[Container] Date Time Running command export JRE_HOME="$JRE_8_HOME"

[Container] Date Time Running command export JDK_HOME="$JDK_8_HOME"

[Container] Date Time Running command for tool_path in "$JAVA_8_HOME"/bin/* "$JRE_8_HOME"/bin/*;
```

10. Perbarui runtime-versions bagian dengan Java versi 11:

```
install:
  runtime-versions:
    java: corretto11
```

11. Setelah Anda menyimpan perubahan, jalankan build Anda lagi dan lihat output build. Anda akan melihat bahwa versi Java yang diinstal adalah 11. Anda akan melihat output yang serupa dengan yang berikut:

```
[Container] Date Time Phase is DOWNLOAD_SOURCE
[Container] Date Time CODEBUILD_SRC_DIR=/codebuild/output/src460614277/src
[Container] Date Time YAML location is /codebuild/output/src460614277/src/buildspec.yml
[Container] Date Time Processing environment variables
[Container] Date Time Selecting 'java' runtime version 'corretto11' based on manual selections...
Installing Java version 11 ...

[Container] Date Time Running command export JAVA_HOME="$JAVA_11_HOME"

[Container] Date Time Running command export JRE_HOME="$JRE_11_HOME"

[Container] Date Time Running command export JDK_HOME="$JDK_11_HOME"
```

```
[Container] Date Time Running command for tool_path in "$JAVA_11_HOME"/bin/*
"$JRE_11_HOME"/bin/*;
```

Tentukan dua runtime

Anda dapat menentukan lebih dari satu runtime dalam proyek CodeBuild build yang sama. Proyek contoh ini menggunakan dua file sumber: satu yang menggunakan runtime Go dan satu yang menggunakan runtime Node.js.

1. Membuat sebuah direktori bernama `my-source`.
2. Di dalam `my-source` direktori, buat direktori bernama `golang-app`.
3. Buat file bernama `hello.go` dengan isi berikut ini. Simpan file di `golang-app` direktori.

```
package main
import "fmt"

func main() {
    fmt.Println("hello world from golang")
    fmt.Println("1+1 =", 1+1)
    fmt.Println("7.0/3.0 =", 7.0/3.0)
    fmt.Println(true && false)
    fmt.Println(true || false)
    fmt.Println(!true)
    fmt.Println("good bye from golang")
}
```

4. Di dalam `my-source` direktori, buat direktori bernama `nodejs-app`. Itu harus pada tingkat yang sama dengan `golang-app` direktori.
5. Buat file bernama `index.js` dengan isi berikut ini. Simpan file di `nodejs-app` direktori.

```
console.log("hello world from nodejs");
console.log("1+1 =" + (1+1));
console.log("7.0/3.0 =" + 7.0/3.0);
console.log(true && false);
console.log(true || false);
console.log(!true);
console.log("good bye from nodejs");
```

6. Buat file bernama `package.json` dengan isi berikut ini. Simpan file di `nodejs-app` direktori.

```
{
  "name": "mycompany-app",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"run some tests here\""
  },
  "author": "",
  "license": "ISC"
}
```

7. Buat file bernama `buildspec.yml` dengan isi berikut ini. Simpan file di `my-source` direktori, pada tingkat yang sama dengan `golang-app` direktori `nodejs-app` dan `runtime-versions`. Bagian ini menentukan runtime Node.js versi 12 dan Go versi 1.13.

```
version: 0.2


phases:
  install:
    runtime-versions:
      golang: 1.13
      nodejs: 12
  build:
    commands:
      - echo Building the Go code...
      - cd $CODEBUILD_SRC_DIR/golang-app
      - go build hello.go
      - echo Building the Node code...
      - cd $CODEBUILD_SRC_DIR/nodejs-app
      - npm run test
  artifacts:
    secondary-artifacts:
      golang_artifacts:
        base-directory: golang-app
        files:
          - hello
      nodejs_artifacts:
        base-directory: nodejs-app
        files:
          - index.js
```

```
- package.json
```

8. Struktur file Anda sekarang akan terlihat seperti ini.

```
my-source
### golang-app
#   ### hello.go
### nodejs.app
#   ### index.js
#   ### package.json
### buildspec.yml
```

9. Unggah isi `my-source` direktori ke bucket input S3 atau repositori CodeCommit, GitHub, atau Bitbucket.

 **Important**

Jika Anda menggunakan bucket input S3, pastikan untuk membuat file ZIP yang berisi struktur direktori dan file, lalu unggah ke bucket input. Jangan tambahkan `my-source` ke file ZIP, hanya direktori dan file `my-source`.

10. Buka AWS CodeBuild konsol di <https://console.aws.amazon.com/codesuite/codebuild/home>.

11. Buat proyek build. Lihat informasi yang lebih lengkap di [Buat proyek build \(konsol\)](#) dan [Menjalankan build \(konsol\)](#). Biarkan semua pengaturan pada nilai default mereka, kecuali untuk pengaturan ini.

- Untuk Lingkungan:
 - Untuk gambar Lingkungan, pilih Gambar terkelola.
 - Untuk sistem operasi, pilih Amazon Linux 2.
 - Untuk Runtime, pilih Standar.
 - Untuk Gambar, pilih `aws/codebuild/amazonlinux2-x86_64-standard:4.0`.

12. Pilih Buat proyek build.

13. Pilih Mulai membangun.

14. Pada konfigurasi Build, terima defaultnya, lalu pilih Start build.

15. Setelah build selesai, lihat output build di tab Build logs. Anda semestinya melihat output yang serupa berikut. Ini menunjukkan output dari runtime Go dan Node.js. Ini juga menunjukkan output dari aplikasi Go dan Node.js.


```
[Container] Date Time Processing environment variables
[Container] Date Time Selecting 'golang' runtime version '1.13' based on manual
  selections...
[Container] Date Time Selecting 'nodejs' runtime version '12' based on manual
  selections...
[Container] Date Time Running command echo "Installing Go version 1.13 ..."
Installing Go version 1.13 ...

[Container] Date Time Running command echo "Installing Node.js version 12 ..."
Installing Node.js version 12 ...

[Container] Date Time Running command n $NODE_12_VERSION
  installed : v12.20.1 (with npm 6.14.10)

[Container] Date Time Moving to directory /codebuild/output/src819694850/src
[Container] Date Time Registering with agent
[Container] Date Time Phases found in YAML: 2
[Container] Date Time  INSTALL: 0 commands
[Container] Date Time  BUILD: 1 commands
[Container] Date Time Phase complete: DOWNLOAD_SOURCE State: SUCCEEDED
[Container] Date Time Phase context status code: Message:
[Container] Date Time Entering phase INSTALL
[Container] Date Time Phase complete: INSTALL State: SUCCEEDED
[Container] Date Time Phase context status code: Message:
[Container] Date Time Entering phase PRE_BUILD
[Container] Date Time Phase complete: PRE_BUILD State: SUCCEEDED
[Container] Date Time Phase context status code: Message:
[Container] Date Time Entering phase BUILD
[Container] Date Time Running command echo Building the Go code...
Building the Go code...

[Container] Date Time Running command cd $CODEBUILD_SRC_DIR/golang-app

[Container] Date Time Running command go build hello.go

[Container] Date Time Running command echo Building the Node code...
Building the Node code...

[Container] Date Time Running command cd $CODEBUILD_SRC_DIR/nodejs-app

[Container] Date Time Running command npm run test

> mycompany-app@1.0.0 test /codebuild/output/src924084119/src/nodejs-app
```

```
> echo "run some tests here"

run some tests here
```

Contoh versi sumber dengan AWS CodeBuild

Contoh ini menunjukkan cara menentukan versi sumber Anda menggunakan format selain ID komit (juga dikenal sebagai komit SHA). Anda dapat menentukan versi sumber Anda dengan cara berikut:

- Untuk penyedia sumber Amazon S3, gunakan ID versi objek yang mewakili file ZIP input build.
- Untuk CodeCommit, Bitbucket, GitHub, dan GitHub Enterprise Server, gunakan salah satu dari berikut ini:
 - Permintaan tarik sebagai referensi permintaan tarik (misalnya, `refs/pull/1/head`).
 - Cabang sebagai nama cabang.
 - Komit ID.
 - Tag.
 - Referensi dan ID komit. Referensi dapat berupa salah satu dari yang berikut:
 - Tag (misalnya, `refs/tags/mytagv1.0^{full-commit-SHA}`).
 - Cabang (misalnya, `refs/heads/mydevbranch^{full-commit-SHA}`).
 - Permintaan tarik (misalnya, `refs/pull/1/head^{full-commit-SHA}`).

Note

Anda dapat menentukan versi sumber permintaan tarik hanya jika repositori Anda GitHub atau Server GitHub Perusahaan.

Jika Anda menggunakan referensi dan ID komit untuk menentukan versi, `DOWNLOAD_SOURCE` fase build akan lebih cepat daripada jika Anda hanya menyediakan versi saja. Ini karena ketika Anda menambahkan referensi, CodeBuild tidak perlu mengunduh seluruh repositori untuk menemukan komit.

- Anda dapat menentukan versi sumber hanya dengan ID komit, seperti `12345678901234567890123467890123456789`. Jika Anda melakukan ini, CodeBuild harus men-download seluruh repositori untuk menemukan versi.

- Anda dapat menentukan versi sumber dengan referensi dan ID komit dalam format ini: `refs/heads/branchname^{full-commit-SHA}` (misalnya, `refs/heads/main^{12345678901234567890123467890123456789}`). Jika Anda melakukan ini, CodeBuild unduh hanya cabang yang ditentukan untuk menemukan versinya.

Note

Untuk mempercepat `DOWNLOAD_SOURCE` fase build Anda, Anda juga dapat menyetel kedalaman klon Git ke angka yang rendah. CodeBuild mengunduh lebih sedikit versi repositori Anda.

Untuk menentukan versi GitHub repositori dengan ID komit

1. Buka AWS CodeBuild konsol di <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Buat proyek build. Untuk informasi selengkapnya, lihat [Buat proyek build \(konsol\)](#) dan [Menjalankan build \(konsol\)](#). Biarkan semua pengaturan pada nilai default mereka, kecuali untuk pengaturan ini:
 - Dalam Sumber:
 - Untuk penyedia Sumber, pilih GitHub. Jika Anda tidak terhubung GitHub, ikuti instruksi untuk menghubungkan.
 - Untuk Repositori, pilih Repositori publik.
 - Untuk URL Repositori, masukkan. **`https://github.com/aws/aws-sdk-ruby.git`**
 - Di Lingkungan:
 - Untuk gambar Lingkungan, pilih Gambar terkelola.
 - Untuk sistem operasi, pilih Amazon Linux 2.
 - Untuk Runtime, pilih Standar.
 - Untuk Gambar, pilih `aws/codebuild/amazonlinux2-x86_64-standard:4.0`.
3. Untuk spesifikasi Build, pilih Insert build command, lalu pilih Switch to editor.
4. Dalam perintah Build, ganti teks placeholder dengan yang berikut ini:

```
version: 0.2
```

```
phases:
```

```
install:
  runtime-versions:
    ruby: 2.6
build:
  commands:
    - echo $CODEBUILD_RESOLVED_SOURCE_VERSION
```

`runtime-versions` Bagian ini diperlukan saat Anda menggunakan gambar standar Ubuntu 2.0. Di sini, runtime Ruby versi 2.6 ditentukan, tetapi Anda dapat menggunakan runtime apa pun. `echo` Perintah menampilkan versi kode sumber yang disimpan dalam variabel `CODEBUILD_RESOLVED_SOURCE_VERSION` lingkungan.

5. Pada konfigurasi Build, terima defaultnya, lalu pilih Start build.
6. Untuk versi Sumber, masukkan **046e8b67481d53bdc86c3f6affdd5d1afae6d369**. Ini adalah SHA dari komit di <https://github.com/aws/aws-sdk-ruby.git> repositori.
7. Pilih Mulai membangun.
8. Ketika build selesai, Anda akan melihat yang berikut:
 - Pada tab Build logs, versi sumber proyek mana yang digunakan. Inilah contohnya.

```
[Container] Date Time Running command echo $CODEBUILD_RESOLVED_SOURCE_VERSION
046e8b67481d53bdc86c3f6affdd5d1afae6d369

[Container] Date Time Phase complete: BUILD State: SUCCEEDED
```

- Pada tab variabel Lingkungan, versi sumber Terselesaikan cocok dengan ID komit yang digunakan untuk membuat build.
- Pada tab Detail fase, durasi `DOWNLOAD_SOURCE` fase.

Langkah-langkah ini menunjukkan cara membuat build menggunakan versi sumber yang sama. Kali ini, versi sumber ditentukan menggunakan referensi dengan ID komit.

Untuk menentukan versi GitHub repositori dengan ID komit dan referensi

1. Dari panel navigasi kiri, pilih Membangun proyek, lalu pilih proyek yang Anda buat sebelumnya.
2. Pilih Mulai membangun.
3. Dalam versi Sumber, masukkan `refs/heads/main^{046e8b67481d53bdc86c3f6affdd5d1afae6d369}`. Ini adalah ID komit yang sama dan referensi ke cabang dalam format `refs/heads/branchname^{full-commit-SHA}`.

4. Pilih Mulai membangun.
5. Ketika build selesai, Anda akan melihat yang berikut:
 - Pada tab Build logs, versi sumber proyek mana yang digunakan. Inilah contohnya.

```
[Container] Date Time Running command echo $CODEBUILD_RESOLVED_SOURCE_VERSION  
046e8b67481d53bdc86c3f6affdd5d1afae6d369
```

```
[Container] Date Time Phase complete: BUILD State: SUCCEEDED
```

- Pada tab variabel Lingkungan, versi sumber Terselesaikan cocok dengan ID komit yang digunakan untuk membuat build.
- Pada tab Detail fase, durasi DOWNLOAD_SOURCE fase harus lebih pendek dari durasi saat Anda hanya menggunakan ID komit untuk menentukan versi sumber Anda.

Sampel repositori sumber pihak ketiga untuk CodeBuild

Topik

- [Permintaan tarik Bitbucket dan sampel filter webhook untuk CodeBuild](#)
- [GitHub Contoh Server Perusahaan untuk CodeBuild](#)
- [GitHub permintaan tarik dan sampel filter webhook untuk CodeBuild](#)

Permintaan tarik Bitbucket dan sampel filter webhook untuk CodeBuild

AWS CodeBuild mendukung webhooks ketika repositori sumber adalah Bitbucket. Ini berarti bahwa untuk proyek CodeBuild build yang memiliki kode sumbernya disimpan dalam repositori Bitbucket, webhook dapat digunakan untuk membangun kembali kode sumber setiap kali perubahan kode didorong ke repositori. Untuk informasi selengkapnya, lihat [Acara webhook Bitbucket](#).

Contoh ini menunjukkan cara membuat permintaan tarik menggunakan repositori Bitbucket. Ini juga menunjukkan kepada Anda cara menggunakan webhook Bitbucket CodeBuild untuk memicu pembuatan proyek.

Note

Saat menggunakan webhook, pengguna dapat memicu build yang tidak terduga. Untuk mengurangi risiko ini, lihat [Praktik terbaik untuk menggunakan webhooks](#)

Topik

- [Prasyarat](#)
- [Buat proyek build dengan Bitbucket sebagai repositori sumber dan aktifkan webhook](#)
- [Memicu build dengan webhook Bitbucket](#)

Prasyarat

Untuk menjalankan sampel ini, Anda harus menghubungkan AWS CodeBuild proyek Anda dengan akun Bitbucket Anda.

Note

CodeBuild telah memperbarui izinnya dengan Bitbucket. Jika sebelumnya Anda menghubungkan proyek Anda ke Bitbucket dan sekarang menerima kesalahan koneksi Bitbucket, Anda harus menyambung kembali untuk memberikan CodeBuild izin untuk mengelola webhook Anda.

Buat proyek build dengan Bitbucket sebagai repositori sumber dan aktifkan webhook

Langkah-langkah berikut menjelaskan cara membuat AWS CodeBuild proyek dengan Bitbucket sebagai repositori sumber dan mengaktifkan webhooks.

1. Buka AWS CodeBuild konsol di <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Jika halaman CodeBuild informasi ditampilkan, pilih Buat proyek build. Jika tidak, pada panel navigasi, perluas Build, pilih Build projects, lalu pilih Create build project.
3. Pilih Buat proyek build.
4. Dalam konfigurasi Proyek:

Nama proyek

Masukkan nama untuk proyek build ini. Membangun nama proyek harus unik di setiap AWS akun. Anda juga dapat menyertakan deskripsi opsional proyek build untuk membantu pengguna lain memahami tujuan proyek ini.

5. Dalam Sumber:

Penyedia sumber

Pilih Bitbucket. Ikuti petunjuk untuk menghubungkan (atau menyambung kembali) dengan Bitbucket dan kemudian pilih Otorisasi.

Repositori

Pilih Repositori di akun Bitbucket saya.

Jika sebelumnya Anda belum terhubung ke akun Bitbucket Anda, masukkan nama pengguna dan kata sandi aplikasi Bitbucket Anda, lalu pilih Simpan kredensi Bitbucket.

Repositori Bitbucket

Masukkan URL untuk repositori Bitbucket Anda.

6. Di Acara webhook sumber utama, pilih yang berikut ini.

Note

Bagian peristiwa webhook sumber utama hanya terlihat jika Anda memilih Repositori di akun Bitbucket saya di langkah sebelumnya.

1. Pilih Bangun ulang setiap kali perubahan kode didorong ke repositori ini saat Anda membuat proyek Anda.
2. Dari jenis Acara, pilih satu atau beberapa acara.
3. Untuk memfilter saat peristiwa memicu build, di bawah Memulai build dalam kondisi ini, tambahkan satu atau beberapa filter opsional.
4. Untuk memfilter saat peristiwa tidak dipicu, di bawah Jangan memulai build dalam kondisi ini, tambahkan satu atau beberapa filter opsional.
5. Pilih Tambahkan grup filter untuk menambahkan grup filter lain, jika diperlukan.

Untuk informasi selengkapnya tentang jenis dan filter acara webhook Bitbucket, lihat [Acara webhook Bitbucket](#)

7. Di Lingkungan:

Citra lingkungan

Pilih salah satu cara berikut:

Untuk menggunakan image Docker yang dikelola oleh AWS CodeBuild:

Pilih Gambar terkelola, lalu buat pilihan dari Sistem operasi, Runtime, Gambar, dan versi Gambar. Buat pilihan dari jenis Lingkungan jika tersedia.

Untuk menggunakan gambar Docker lain:

Pilih gambar Kustom. Untuk jenis Lingkungan, pilih ARM, Linux, GPU Linux, atau Windows. Jika Anda memilih Registri lain, untuk URL registri eksternal, masukkan nama dan tag gambar Docker di Docker Hub, menggunakan format *docker repository/docker image name* Jika Anda memilih Amazon ECR, gunakan repositori Amazon ECR dan gambar Amazon ECR untuk memilih gambar Docker di akun Anda. AWS

Untuk menggunakan image Docker pribadi:

Pilih gambar Kustom. Untuk jenis Lingkungan, pilih ARM, Linux, GPU Linux, atau Windows. Untuk registri Gambar, pilih Registri lain, lalu masukkan ARN kredensial untuk gambar Docker pribadi Anda. Kredensialnya harus dibuat oleh Secrets Manager. Untuk informasi lebih lanjut, lihat [Apa itu AWS Secrets Manager?](#) dalam AWS Secrets Manager User Guide.

Peran layanan

Pilih salah satu cara berikut:

- Jika Anda tidak memiliki peran CodeBuild layanan, pilih Peran layanan baru. Di Nama peran, masukkan nama untuk peran baru.
- Jika Anda memiliki peran CodeBuild layanan, pilih Peran layanan yang ada. Di Peran ARN, pilih peran layanan.

Note

Saat menggunakan konsol untuk membuat atau memperbarui proyek build, Anda dapat membuat peran CodeBuild layanan secara bersamaan. Secara default, peran hanya berfungsi dengan proyek build tersebut. Jika Anda menggunakan konsol untuk mengaitkan peran layanan ini dengan proyek build lain, peran akan diperbarui agar

berfungsi dengan proyek build lainnya. Peran layanan dapat bekerja dengan hingga 10 proyek pembangunan.

8. Di Buildspec, lakukan salah satu hal berikut:

- Pilih Gunakan file buildspec untuk menggunakan file buildspec.yml di direktori root kode sumber.
- Pilih Sisipkan perintah build untuk menggunakan konsol untuk menyisipkan perintah build.

Lihat informasi yang lebih lengkap di [Referensi Buildspec](#).

9. Dalam Artefak:

Jenis

Pilih salah satu cara berikut:

- Jika Anda tidak ingin membuat artefak keluaran build, pilih Tidak ada artefak.
- Untuk menyimpan output build dalam bucket S3, pilih Amazon S3, lalu lakukan hal berikut:
 - Jika Anda ingin menggunakan nama proyek Anda untuk file atau folder ZIP keluaran build, biarkan Nama kosong. Jika tidak, masukkan nama. Secara default, nama artefak adalah nama proyek. Jika Anda ingin menggunakan nama yang berbeda, masukkan di kotak nama artefak. Jika Anda ingin mengeluarkan file ZIP, sertakan ekstensi zip.
 - Untuk nama Bucket, pilih nama bucket keluaran.
 - Jika Anda memilih Sisipkan perintah build sebelumnya dalam prosedur ini, untuk file Output, masukkan lokasi file dari build yang ingin Anda masukkan ke dalam file atau folder ZIP keluaran build. Untuk beberapa lokasi, pisahkan setiap lokasi dengan koma (misalnya, `appspect.yml, target/my-app.jar`). Untuk informasi lebih lanjut, lihat deskripsi files di [Sintaks Buildspec](#).

Konfigurasi tambahan

Perluas Konfigurasi tambahan dan atur opsi yang sesuai.

10. Pilih Buat proyek build. Pada halaman Review, pilih Start build untuk menjalankan build.

Memicu build dengan webhook Bitbucket

Untuk proyek yang menggunakan webhook Bitbucket, AWS CodeBuild buat build saat repositori Bitbucket mendeteksi perubahan kode sumber Anda.

1. Buka AWS CodeBuild konsol di <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Pada panel navigasi, pilih Build projects, lalu pilih project yang terkait dengan repositori Bitbucket dengan webhook. Untuk informasi tentang membuat proyek webhook Bitbucket, lihat [the section called “Buat proyek build dengan Bitbucket sebagai repositori sumber dan aktifkan webhook”](#)
3. Buat beberapa perubahan dalam kode di repositori Bitbucket proyek Anda.
4. Buat permintaan tarik pada repositori Bitbucket Anda. Untuk informasi selengkapnya, lihat [Membuat permintaan tarik](#).
5. Pada halaman webhooks Bitbucket, pilih Lihat permintaan untuk melihat daftar peristiwa terbaru.
6. Pilih Lihat detail untuk melihat detail tentang respons yang dikembalikan oleh CodeBuild. Mungkin terlihat seperti ini:

```
"response": "Webhook received and build started: https://us-east-1.console.aws.amazon.com/codebuild/home..."
"statusCode": 200
```

7. Arahkan ke halaman permintaan tarik Bitbucket untuk melihat status build.

GitHub Contoh Server Perusahaan untuk CodeBuild

AWS CodeBuild mendukung GitHub Enterprise Server sebagai repositori sumber. Contoh ini menunjukkan cara menyiapkan CodeBuild proyek Anda ketika repositori GitHub Enterprise Server Anda memiliki sertifikat yang diinstal. Ini juga menunjukkan cara mengaktifkan webhook sehingga CodeBuild membangun kembali kode sumber setiap kali perubahan kode didorong ke repositori Server Perusahaan Anda GitHub .

Prasyarat

1. Buat token akses pribadi untuk CodeBuild proyek Anda. Kami menyarankan Anda membuat pengguna GitHub Enterprise dan membuat token akses pribadi untuk pengguna ini. Salin ke clipboard Anda sehingga dapat digunakan saat Anda membuat proyek Anda. CodeBuild Untuk informasi selengkapnya, lihat [Membuat token akses pribadi untuk baris perintah](#) di situs web GitHub Bantuan.

Saat Anda membuat token akses pribadi, sertakan lingkup repo dalam definisi.

Select scopes

Scopes define the access for personal tokens. [Read more about OAuth scopes.](#)

<input checked="" type="checkbox"/> repo	Full control of private repositories
<input checked="" type="checkbox"/> repo:status	Access commit status
<input checked="" type="checkbox"/> repo_deployment	Access deployment status
<input checked="" type="checkbox"/> public_repo	Access public repositories

2. Unduh sertifikat Anda dari Server GitHub Perusahaan. CodeBuild menggunakan sertifikat untuk membuat koneksi SSL terpercaya ke repositori.

Klien Linux/macOS:

Dari jendela terminal, jalankan perintah berikut:

```
echo -n | openssl s_client -connect HOST:PORTNUMBER \  
| sed -ne '/-BEGIN CERTIFICATE-/,/-END CERTIFICATE-/p' > /folder/filename.pem
```


Ganti placeholder dalam perintah dengan nilai-nilai berikut:

TUAN RUMAH. Alamat IP repositori Server GitHub Perusahaan Anda.

PORTNUMBER. Nomor port yang Anda gunakan untuk menghubungkan (misalnya, 443).

folder. Folder tempat Anda mengunduh sertifikat Anda.

nama file. Nama file file sertifikat Anda.

 Important

Simpan sertifikat sebagai file.pem.


Klien Windows:

Gunakan browser Anda untuk mengunduh sertifikat Anda dari GitHub Enterprise Server. Untuk melihat detail sertifikat situs, pilih ikon gembok. Untuk informasi tentang cara mengekspor sertifikat, lihat dokumentasi browser Anda.

 Important

Simpan sertifikat sebagai file.pem.


3. Unggah file sertifikat Anda ke bucket S3. Untuk informasi tentang cara membuat bucket S3, lihat [Bagaimana cara membuat Bucket S3?](#) Untuk informasi tentang cara mengunggah objek ke bucket S3, lihat [Bagaimana cara mengunggah file dan folder ke bucket?](#)

 Note

Ember ini harus berada di AWS wilayah yang sama dengan bangunan Anda. Misalnya, jika Anda menginstruksikan CodeBuild untuk menjalankan bangunan di Wilayah Timur AS (Ohio), ember harus berada di Wilayah Timur AS (Ohio).

Buat proyek build dengan GitHub Enterprise Server sebagai repositori sumber dan aktifkan webhook (konsol)


1. Buka AWS CodeBuild konsol di <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Jika halaman CodeBuild informasi ditampilkan, pilih Buat proyek build. Jika tidak, pada panel navigasi, perluas Build, pilih Build projects, lalu pilih Create build project.
3. Di Nama proyek, masukkan nama untuk proyek pembangunan ini. Membangun nama proyek harus unik di setiap AWS akun. Anda juga dapat menyertakan deskripsi opsional proyek build untuk membantu pengguna lain memahami tujuan proyek ini.
4. Di Sumber, di penyedia Sumber, pilih GitHub Perusahaan.
 - Untuk Token Akses Pribadi, tempel token yang Anda salin ke clipboard Anda dan pilih Simpan Token. Di URL Repositori, masukkan URL untuk repositori Server GitHub Perusahaan Anda.

 Note

Anda hanya perlu memasukkan dan menyimpan token akses pribadi sekali. Semua AWS CodeBuild proyek future menggunakan token ini.

- Di URL Repositori, masukkan jalur ke repositori Anda, termasuk nama repositori.
- Perluas Additional configuration (Konfigurasi tambahan).

- Pilih Rebuild setiap kali perubahan kode didorong ke repositori ini untuk membangun kembali setiap kali perubahan kode didorong ke repositori ini.
- Pilih Aktifkan SSL yang tidak aman untuk mengabaikan peringatan SSL saat Anda terhubung ke repositori proyek GitHub Enterprise Server Anda.

 Note

Kami menyarankan Anda menggunakan Aktifkan SSL yang tidak aman hanya untuk pengujian. Ini tidak boleh digunakan di lingkungan produksi.

Source Add source

Source 1 - Primary

Source provider

GitHub Enterprise

Repository URL

https://<host-name>/<user-name>/<repository-name>

Disconnect GitHub Enterprise account

▼ Additional configuration
Git clone depth, Insecure SSL

Git clone depth - *optional*

1

Webhook - *optional*

Rebuild every time a code change is pushed to this repository

Branch filter - *optional*

Enter a regular expression

Insecure SSL - *optional*
Enable this flag to ignore SSL warnings while connecting to project source.

Enable insecure SSL

5. Di Lingkungan:

Untuk citra Lingkungan, lakukan salah satu hal berikut:

- Untuk menggunakan image Docker yang dikelola oleh AWS CodeBuild, pilih Managed image, lalu buat pilihan dari Sistem operasi, Runtime, Image, dan versi Image. Buat pilihan dari jenis Lingkungan jika tersedia.
- Untuk menggunakan gambar Docker lain, pilih Custom image. Untuk jenis Lingkungan, pilih ARM, Linux, GPU Linux, atau Windows. Jika Anda memilih Registri lain, untuk URL registri

eksternal, masukkan nama dan tag gambar Docker di Docker Hub, menggunakan format. *docker repository/docker image name* Jika Anda memilih Amazon ECR, gunakan repositori Amazon ECR dan gambar Amazon ECR untuk memilih gambar Docker di akun Anda. AWS

- Untuk menggunakan image Docker pribadi, pilih Custom image. Untuk jenis Lingkungan, pilih ARM, Linux, GPU Linux, atau Windows. Untuk registri Gambar, pilih Registri lain, lalu masukkan ARN kredensial untuk gambar Docker pribadi Anda. Kredensialnya harus dibuat oleh Secrets Manager. Untuk informasi lebih lanjut, lihat [Apa itu AWS Secrets Manager?](#) di Panduan Pengguna AWS Secrets Manager .

6. Dalam peran Layanan, lakukan salah satu hal berikut:

- Jika Anda tidak memiliki peran CodeBuild layanan, pilih Peran layanan baru. Di Nama peran, masukkan nama untuk peran baru.
- Jika Anda memiliki peran CodeBuild layanan, pilih Peran layanan yang ada. Di Peran ARN, pilih peran layanan.

Note

Saat menggunakan konsol untuk membuat atau memperbarui proyek build, Anda dapat membuat peran CodeBuild layanan secara bersamaan. Secara default, peran hanya berfungsi dengan proyek build tersebut. Jika Anda menggunakan konsol untuk mengaitkan peran layanan ini dengan proyek build lain, peran akan diperbarui agar berfungsi dengan proyek build lainnya. Peran layanan dapat bekerja dengan hingga 10 proyek pembangunan.

7. Perluas Additional configuration (Konfigurasi tambahan).

Jika Anda CodeBuild ingin bekerja dengan VPC Anda:

- Untuk VPC, pilih ID VPC yang digunakan. CodeBuild
- Untuk Subnet VPC, pilih subnet yang menyertakan sumber daya yang digunakan. CodeBuild
- Untuk grup Keamanan VPC, pilih grup keamanan yang CodeBuild digunakan untuk memungkinkan akses ke sumber daya di VPC.

Untuk informasi selengkapnya, lihat [Gunakan AWS CodeBuild dengan Amazon Virtual Private Cloud](#).

8. Di Buildspec, lakukan salah satu hal berikut:

- Pilih Gunakan file buildspec untuk menggunakan file buildspec.yml di direktori root kode sumber.
- Pilih Sisipkan perintah build untuk menggunakan konsol untuk menyisipkan perintah build.

Lihat informasi yang lebih lengkap di [Referensi Buildspec](#).

9. Di Artefak, untuk Jenis, lakukan salah satu hal berikut:

- Jika Anda tidak ingin membuat artefak keluaran build, pilih Tidak ada artefak.
- Untuk menyimpan output build dalam bucket S3, pilih Amazon S3, lalu lakukan hal berikut:
 - Jika Anda ingin menggunakan nama proyek Anda untuk file atau folder ZIP keluaran build, biarkan Nama kosong. Jika tidak, masukkan nama. Secara default, nama artefak adalah nama proyek. Jika Anda ingin menggunakan nama yang berbeda, masukkan di kotak nama artefak. Jika Anda ingin mengeluarkan file ZIP, sertakan ekstensi zip.
 - Untuk nama Bucket, pilih nama bucket keluaran.
 - Jika Anda memilih Sisipkan perintah build sebelumnya dalam prosedur ini, untuk file Output, masukkan lokasi file dari build yang ingin Anda masukkan ke dalam file atau folder ZIP keluaran build. Untuk beberapa lokasi, pisahkan setiap lokasi dengan koma (misalnya, `appspect.yml, target/my-app.jar`). Untuk informasi lebih lanjut, lihat deskripsi files di [Sintaks Buildspec](#).

10. Untuk jenis Cache, pilih salah satu dari berikut ini:

- Jika Anda tidak ingin menggunakan cache, pilih Tidak ada cache.
- Jika Anda ingin menggunakan cache Amazon S3, pilih Amazon S3, lalu lakukan hal berikut:
 - Untuk Bucket, pilih nama bucket S3 tempat cache disimpan.
 - (Opsional) Untuk awalan jalur Cache, masukkan awalan jalur Amazon S3. Nilai awalan jalur Cache mirip dengan nama direktori. Itu memungkinkan Anda untuk menyimpan cache di bawah direktori yang sama dalam ember.

Important

Jangan menambahkan garis miring (/) ke ujung awalan jalur.

- Jika Anda ingin menggunakan cache lokal, pilih Lokal, lalu pilih satu atau beberapa mode cache lokal.

Note

Mode cache lapisan Docker hanya tersedia untuk Linux. Jika Anda memilihnya, proyek Anda harus berjalan dalam mode istimewa.

Menggunakan cache menghemat banyak waktu pembuatan karena bagian lingkungan build yang dapat digunakan kembali disimpan dalam cache dan digunakan di seluruh build. Untuk informasi tentang menentukan cache dalam file buildspec, lihat [Sintaks Buildspec](#) Untuk informasi selengkapnya tentang caching, lihat [Membangun caching di AWS CodeBuild](#).

11. Pilih Buat proyek build. Pada halaman proyek build, pilih Start build.
12. Jika Anda mengaktifkan webhook di Sumber, kotak dialog Buat webhook akan ditampilkan dengan nilai untuk URL Payload dan Rahasia.

Important

Kotak dialog Create webhook hanya muncul sekali. Salin URL payload dan kunci rahasia. Anda membutuhkannya saat menambahkan webhook di GitHub Enterprise Server.

Jika Anda perlu membuat URL payload dan kunci rahasia lagi, Anda harus terlebih dahulu menghapus webhook dari repositori GitHub Enterprise Server Anda. Dalam CodeBuild proyek Anda, kosongkan kotak centang Webhook dan kemudian pilih Simpan. Anda kemudian dapat membuat atau memperbarui CodeBuild proyek dengan kotak centang Webhook yang dipilih. Kotak dialog Create webhook muncul lagi.

13. Di GitHub Enterprise Server, pilih repositori tempat CodeBuild proyek Anda disimpan.
14. Pilih Pengaturan, pilih Hooks & services, lalu pilih Add webhook.
15. Masukkan URL payload dan kunci rahasia, terima default untuk bidang lainnya, lalu pilih Tambahkan webhook.

requests 0 Projects 0 Wiki Pulse Graphs Settings

Webhooks / Add webhook

We'll send a POST request to the URL below with details of any subscribed events. You can also specify which data format you'd like to receive (JSON, x-www-form-urlencoded, etc). More information can be found in [our developer documentation](#).

Payload URL *

Content type

application/json

Secret

By default, we verify SSL certificates when delivering payloads. [Disable SSL verification](#)

Which events would you like to trigger this webhook?

Just the push event.

Send me everything.

Let me select individual events.

Active
We will deliver event details when this hook is triggered.


[Add webhook](#)

16. Kembali ke CodeBuild proyek Anda. Tutup kotak dialog Create webhook dan pilih Start build.

GitHub permintaan tarik dan sampel filter webhook untuk CodeBuild

AWS CodeBuild mendukung webhook saat repositori sumber. GitHub Ini berarti bahwa untuk proyek CodeBuild build yang memiliki kode sumbernya disimpan dalam GitHub repositori, webhook

dapat digunakan untuk membangun kembali kode sumber setiap kali perubahan kode didorong ke repositori. Untuk CodeBuild sampel, lihat [AWS CodeBuild Sampel](#).

 Note

Saat menggunakan webhook, pengguna dapat memicu build yang tidak terduga. Untuk mengurangi risiko ini, lihat [Praktik terbaik untuk menggunakan webhooks](#)

Buat proyek build dengan GitHub sebagai repositori sumber dan aktifkan webhook (konsol)

1. Buka AWS CodeBuild konsol di <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Jika halaman CodeBuild informasi ditampilkan, pilih Buat proyek build. Jika tidak, pada panel navigasi, perluas Build, pilih Build projects, lalu pilih Create build project.
3. Pilih Buat proyek build.
4. Dalam konfigurasi Proyek:

Nama proyek

Masukkan nama untuk proyek build ini. Membangun nama proyek harus unik di setiap AWS akun. Anda juga dapat menyertakan deskripsi opsional proyek build untuk membantu pengguna lain memahami tujuan proyek ini.

5. Dalam Sumber:

Penyedia sumber

Pilih GitHub. Ikuti petunjuk untuk menghubungkan (atau menyambung kembali) dengan GitHub dan kemudian pilih Otorisasi.

Repositori

Pilih Repositori di akun saya GitHub.

GitHub repositori

Masukkan URL untuk GitHub repositori Anda.

6. Di Acara webhook sumber utama, pilih yang berikut ini.

Note

Bagian peristiwa webhook sumber utama hanya terlihat jika Anda memilih Repositori di GitHub akun saya pada langkah sebelumnya.

1. Pilih Bangun ulang setiap kali perubahan kode didorong ke repositori ini saat Anda membuat proyek Anda.
2. Dari jenis Acara, pilih satu atau beberapa acara.
3. Untuk memfilter saat peristiwa memicu build, di bawah Memulai build dalam kondisi ini, tambahkan satu atau beberapa filter opsional.
4. Untuk memfilter saat peristiwa tidak dipicu, di bawah Jangan memulai build dalam kondisi ini, tambahkan satu atau beberapa filter opsional.
5. Pilih Tambahkan grup filter untuk menambahkan grup filter lain, jika diperlukan.

Untuk informasi selengkapnya tentang jenis dan filter acara GitHub webhook, lihat [GitHub acara webhook](#).

7. Di Lingkungan:**Citra lingkungan**

Pilih salah satu cara berikut:

Untuk menggunakan image Docker yang dikelola oleh AWS CodeBuild:

Pilih Gambar terkelola, lalu buat pilihan dari Sistem operasi, Runtime, Gambar, dan versi Gambar. Buat pilihan dari jenis Lingkungan jika tersedia.

Untuk menggunakan gambar Docker lain:

Pilih gambar Kustom. Untuk jenis Lingkungan, pilih ARM, Linux, GPU Linux, atau Windows. Jika Anda memilih Registri lain, untuk URL registri eksternal, masukkan nama dan tag gambar Docker di Docker Hub, menggunakan format *docker repository/docker image name* Jika Anda memilih Amazon ECR, gunakan repositori Amazon ECR dan gambar Amazon ECR untuk memilih gambar Docker di akun Anda. AWS

Untuk menggunakan image Docker pribadi:

Pilih gambar Kustom. Untuk jenis Lingkungan, pilih ARM, Linux, GPU Linux, atau Windows. Untuk registri Gambar, pilih Registri lain, lalu masukkan ARN kredensial untuk gambar Docker pribadi Anda. Kredensialnya harus dibuat oleh Secrets Manager. Untuk informasi lebih lanjut, lihat [Apa itu AWS Secrets Manager?](#) dalam AWS Secrets Manager User Guide.

Peran layanan

Pilih salah satu cara berikut:

- Jika Anda tidak memiliki peran CodeBuild layanan, pilih Peran layanan baru. Di Nama peran, masukkan nama untuk peran baru.
- Jika Anda memiliki peran CodeBuild layanan, pilih Peran layanan yang ada. Di Peran ARN, pilih peran layanan.

Note

Saat menggunakan konsol untuk membuat atau memperbarui proyek build, Anda dapat membuat peran CodeBuild layanan secara bersamaan. Secara default, peran hanya berfungsi dengan proyek build tersebut. Jika Anda menggunakan konsol untuk mengaitkan peran layanan ini dengan proyek build lain, peran akan diperbarui agar berfungsi dengan proyek build lainnya. Peran layanan dapat bekerja dengan hingga 10 proyek pembangunan.

8. Di Buildspec, lakukan salah satu hal berikut:

- Pilih Gunakan file buildspec untuk menggunakan file buildspec.yml di direktori root kode sumber.
- Pilih Sisipkan perintah build untuk menggunakan konsol untuk menyisipkan perintah build.

Lihat informasi yang lebih lengkap di [Referensi Buildspec](#).

9. Dalam Artefak:

Jenis

Pilih salah satu cara berikut:

- Jika Anda tidak ingin membuat artefak keluaran build, pilih Tidak ada artefak.

- Untuk menyimpan output build dalam bucket S3, pilih Amazon S3, lalu lakukan hal berikut:
 - Jika Anda ingin menggunakan nama proyek Anda untuk file atau folder ZIP keluaran build, biarkan Nama kosong. Jika tidak, masukkan nama. Secara default, nama artefak adalah nama proyek. Jika Anda ingin menggunakan nama yang berbeda, masukkan di kotak nama artefak. Jika Anda ingin mengeluarkan file ZIP, sertakan ekstensi zip.
 - Untuk nama Bucket, pilih nama bucket keluaran.
 - Jika Anda memilih Sisipkan perintah build sebelumnya dalam prosedur ini, untuk file Output, masukkan lokasi file dari build yang ingin Anda masukkan ke dalam file atau folder ZIP keluaran build. Untuk beberapa lokasi, pisahkan setiap lokasi dengan koma (misalnya, `appspect.yml, target/my-app.jar`). Untuk informasi lebih lanjut, lihat deskripsi files di [Sintaks Buildspec](#).

Konfigurasi tambahan

Perluas Konfigurasi tambahan dan atur opsi yang sesuai.

10. Pilih Buat proyek build. Pada halaman Review, pilih Start build untuk menjalankan build.

Pemeriksaan verifikasi

1. Buka AWS CodeBuild konsol di <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Di panel navigasi, pilih Membangun proyek.
3. Lakukan salah satu dari cara berikut:
 - Pilih tautan untuk proyek build dengan webhook yang ingin Anda verifikasi, lalu pilih Detail build.
 - Pilih tombol di samping proyek build dengan webhook yang ingin Anda verifikasi, pilih Lihat detail, lalu pilih tab Build details.
4. Di acara webhook sumber utama, pilih tautan URL Webhook.
5. Di GitHub repositori Anda, pada halaman Pengaturan, di bawah Webhooks, verifikasi bahwa Permintaan Tarik dan Push dipilih.
6. Di pengaturan GitHub profil Anda, di bawah Pengaturan pribadi, Aplikasi, Aplikasi OAuth Resmi, Anda akan melihat bahwa aplikasi Anda telah diizinkan untuk mengakses AWS Wilayah yang Anda pilih.

Gunakan versi semantik untuk memberi nama contoh artefak build

Contoh ini berisi contoh file buildspec yang menunjukkan cara menentukan nama artefak yang dibuat pada waktu pembuatan. Nama yang ditentukan dalam file buildspec dapat menggabungkan perintah Shell dan variabel lingkungan untuk membuatnya unik. Nama yang Anda tentukan dalam file buildspec akan menggantikan nama yang Anda masukkan di konsol saat membuat project.

Jika Anda membangun beberapa kali, menggunakan nama artefak yang ditentukan dalam file buildspec dapat memastikan nama file artefak keluaran Anda unik. Misalnya, Anda dapat menggunakan tanggal dan stempel waktu yang dimasukkan ke dalam nama artefak pada waktu pembuatan.

Jika Anda ingin mengganti nama artefak yang Anda masukkan di konsol dengan nama di file buildspec, lakukan hal berikut:

1. Tetapkan proyek build Anda untuk mengganti nama artefak dengan nama di file buildspec.
 - Jika Anda menggunakan konsol untuk membuat proyek build, pilih Aktifkan versi semantik. Untuk informasi selengkapnya, lihat [Buat proyek build \(konsol\)](#).
 - Jika Anda menggunakan AWS CLI, atur `overrideArtifactName` ke `true` dalam file berformat JSON diteruskan ke `create-project` Untuk informasi selengkapnya, lihat [Buat proyek build \(AWS CLI\)](#).
 - Jika Anda menggunakan AWS CodeBuild API, setel `overrideArtifactName` flag pada `ProjectArtifacts` objek saat proyek dibuat atau diperbarui atau build dimulai.
2. Tentukan nama dalam file buildspec. Gunakan contoh file buildspec berikut sebagai panduan.

Contoh Linux ini menunjukkan cara menentukan nama artefak yang menyertakan tanggal pembuatan build:

```
version: 0.2
phases:
  build:
    commands:
      - rspec HelloWorld_spec.rb
artifacts:
  files:
    - '**/*'
  name: myname-${date +%Y-%m-%d}
```

Contoh Linux ini menunjukkan cara menentukan nama artefak yang menggunakan variabel CodeBuild lingkungan. Untuk informasi selengkapnya, lihat [Variabel lingkungan di lingkungan build](#).

```
version: 0.2
phases:
  build:
    commands:
      - rspec HelloWorld_spec.rb
artifacts:
  files:
    - '**/*'
  name: myname-$AWS_REGION
```

Contoh Windows ini menunjukkan cara menentukan nama artefak yang menyertakan tanggal dan waktu pembuatan pembuatan:

```
version: 0.2
env:
  variables:
    TEST_ENV_VARIABLE: myArtifactName
phases:
  build:
    commands:
      - cd samples/helloworld
      - dotnet restore
      - dotnet run
artifacts:
  files:
    - '**/*'
  name: $Env:TEST_ENV_VARIABLE-$(Get-Date -UFormat "%Y%m%d-%H%M%S")
```

Contoh Windows ini menunjukkan cara menentukan nama artefak yang menggunakan variabel yang dideklarasikan dalam file buildspec dan variabel lingkungan. Untuk informasi selengkapnya, lihat [Variabel lingkungan di lingkungan build](#).

```
version: 0.2
env:
  variables:
    TEST_ENV_VARIABLE: myArtifactName
phases:
  build:
```



```
commands:
  - cd samples/helloworld
  - dotnet restore
  - dotnet run
artifacts:
  files:
    - '**/*'
name: $Env:TEST_ENV_VARIABLE-$Env:AWS_REGION
```

Untuk informasi selengkapnya, lihat [Buat referensi spesifikasi untuk CodeBuild](#).

Sampel Microsoft Windows untuk CodeBuild

Sampel ini menggunakan lingkungan AWS CodeBuild build yang menjalankan Microsoft Windows Server 2019, .NET Framework, dan .NET Core SDK untuk membuat file runtime dari kode yang ditulis dalam F # dan Visual Basic.

Important

Menjalankan sampel ini dapat mengakibatkan biaya ke AWS akun Anda. Ini termasuk kemungkinan biaya untuk CodeBuild dan untuk AWS sumber daya dan tindakan yang terkait dengan Amazon S3, AWS KMS, dan CloudWatch Log. Untuk informasi selengkapnya, lihat [CodeBuild harga](#), [harga Amazon S3](#), [AWS Key Management Service harga](#), dan [harga Amazon CloudWatch](#).

Menjalankan sampel

Untuk menjalankan sampel ini

1. Buat file seperti yang dijelaskan di bagian “Struktur direktori” dan “File” dari topik ini, lalu unggah ke bucket input S3 atau GitHub repositori CodeCommit atau.

Important

Jangan mengunggah(*root directory name*), hanya file di dalamnya(*root directory name*).

Jika Anda menggunakan bucket input S3, pastikan untuk membuat file ZIP yang berisi file, lalu unggah ke bucket input. Jangan tambahkan (*root directory name*) ke file ZIP, hanya file di dalamnya(*root directory name*).

2. Buat proyek build. Proyek build harus menggunakan `mcr.microsoft.com/dotnet/framework/sdk:4.8` image untuk membangun proyek .NET Framework.

Jika Anda menggunakan AWS CLI untuk membuat proyek build, input berformat JSON ke `create-project` perintah mungkin terlihat mirip dengan ini. (Ganti placeholder dengan nilai Anda sendiri.)

```
{
  "name": "sample-windows-build-project",
  "source": {
    "type": "S3",
    "location": "codebuild-region-ID-account-ID-input-bucket/windows-build-input-artifact.zip"
  },
  "artifacts": {
    "type": "S3",
    "location": "codebuild-region-ID-account-ID-output-bucket",
    "packaging": "ZIP",
    "name": "windows-build-output-artifact.zip"
  },
  "environment": {
    "type": "WINDOWS_SERVER_2019_CONTAINER",
    "image": "mcr.microsoft.com/dotnet/framework/sdk:4.8",
    "computeType": "BUILD_GENERAL1_MEDIUM"
  },
  "serviceRole": "arn:aws:iam::account-ID:role/role-name",
  "encryptionKey": "arn:aws:kms:region-ID:account-ID:key/key-ID"
}
```

3. Jalankan build, dan ikuti langkah-langkahnya [Jalankan build](#).
4. Untuk mendapatkan artefak keluaran build, di bucket keluaran S3 Anda, unduh *windows-build-output-artifact.zip* file ke komputer atau instance lokal Anda. Ekstrak konten untuk sampai ke runtime dan file lainnya.
 - File runtime untuk sampel F # menggunakan .NET Framework, `FSharpHelloWorld.exe`, dapat ditemukan di `FSharpHelloWorld\bin\Debug` direktori.

- File runtime untuk sampel Visual Basic menggunakan `.NET FrameworkVBHelloWorld.exe`, dapat ditemukan di `VBHelloWorld\bin\Debug` direktori.

Struktur direktori

Sampel ini mengasumsikan struktur direktori berikut.

F # dan .NET Framework

```
(root directory name)
### buildspec.yml
### FSharpHelloWorld.sln
### FSharpHelloWorld
### App.config
### AssemblyInfo.fs
### FSharpHelloWorld.fsproj
### Program.fs
```

Visual Basic dan .NET Framework

```
(root directory name)
### buildspec.yml
### VBHelloWorld.sln
### VBHelloWorld
### App.config
### HelloWorld.vb
### VBHelloWorld.vbproj
### My Project
### Application.Designer.vb
### Application.myapp
### AssemblyInfo.vb
### Resources.Designer.vb
### Resources.resx
### Settings.Designer.vb
### Settings.settings
```

Berkas

Sampel ini menggunakan file-file berikut.

F # dan .NET Framework

`buildspec.yml`(dalam(*root directory name*)):

```
version: 0.2

env:
  variables:
    SOLUTION: .\FSharpHelloWorld.sln
    PACKAGE_DIRECTORY: .\packages
    DOTNET_FRAMEWORK: 4.8

phases:
  build:
    commands:
      - '& nuget restore $env:SOLUTION -PackagesDirectory $env:PACKAGE_DIRECTORY'
      - '& msbuild -p:FrameworkPathOverride="C:\Program Files (x86)\Reference
Assemblies\Microsoft\Framework\.NETFramework\v$env:DOTNET_FRAMEWORK" $env:SOLUTION'
artifacts:
  files:
    - .\FSharpHelloWorld\bin\Debug\*
```

`FSharpHelloWorld.sln`(dalam(*root directory name*)):

```
Microsoft Visual Studio Solution File, Format Version 12.00
# Visual Studio 14
VisualStudioVersion = 14.0.25420.1
MinimumVisualStudioVersion = 10.0.40219.1
Project("{F2A71F9B-5D33-465A-A702-920D77279786}") = "FSharpHelloWorld",
  "FSharpHelloWorld\FSharpHelloWorld.fsproj", "{D60939B6-526D-43F4-9A89-577B2980DF62}"
EndProject
Global
  GlobalSection(SolutionConfigurationPlatforms) = preSolution
    Debug|Any CPU = Debug|Any CPU
    Release|Any CPU = Release|Any CPU
  EndGlobalSection
  GlobalSection(ProjectConfigurationPlatforms) = postSolution
    {D60939B6-526D-43F4-9A89-577B2980DF62}.Debug|Any CPU.ActiveCfg = Debug|Any CPU
    {D60939B6-526D-43F4-9A89-577B2980DF62}.Debug|Any CPU.Build.0 = Debug|Any CPU
    {D60939B6-526D-43F4-9A89-577B2980DF62}.Release|Any CPU.ActiveCfg = Release|Any CPU
    {D60939B6-526D-43F4-9A89-577B2980DF62}.Release|Any CPU.Build.0 = Release|Any CPU
  EndGlobalSection
  GlobalSection(SolutionProperties) = preSolution
```

```
HideSolutionNode = FALSE
EndGlobalSection
EndGlobal
```

App.config(dalam(*root directory name*)\FSharpHelloWorld):

```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
  <startup>
    <supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.8" />
  </startup>
</configuration>
```

AssemblyInfo.fs(dalam(*root directory name*)\FSharpHelloWorld):

```
namespace FSharpHelloWorld.AssemblyInfo

open System.Reflection
open System.Runtime.CompilerServices
open System.Runtime.InteropServices

// General Information about an assembly is controlled through the following
// set of attributes. Change these attribute values to modify the information
// associated with an assembly.
[<assembly: AssemblyTitle("FSharpHelloWorld")>]
[<assembly: AssemblyDescription("")>]
[<assembly: AssemblyConfiguration("")>]
[<assembly: AssemblyCompany("")>]
[<assembly: AssemblyProduct("FSharpHelloWorld")>]
[<assembly: AssemblyCopyright("Copyright © 2017")>]
[<assembly: AssemblyTrademark("")>]
[<assembly: AssemblyCulture("")>]

// Setting ComVisible to false makes the types in this assembly not visible
// to COM components. If you need to access a type in this assembly from
// COM, set the ComVisible attribute to true on that type.
[<assembly: ComVisible(false)>]

// The following GUID is for the ID of the typelib if this project is exposed to COM
[<assembly: Guid("d60939b6-526d-43f4-9a89-577b2980df62")>]

// Version information for an assembly consists of the following four values:
//
```

```
// Major Version
// Minor Version
// Build Number
// Revision
//
// You can specify all the values or you can default the Build and Revision Numbers
// by using the '*' as shown below:
// [<assembly: AssemblyVersion("1.0.*")>]
[<assembly: AssemblyVersion("1.0.0.0")>]
[<assembly: AssemblyFileVersion("1.0.0.0")>]

do
  ()
```

FSharpHelloWorld.fsproj(dalam(*root directory name*)\FSharpHelloWorld):

```
<?xml version="1.0" encoding="utf-8"?>
<Project ToolsVersion="14.0" DefaultTargets="Build" xmlns="http://
schemas.microsoft.com/developer/msbuild/2003">
  <Import Project="$(MSBuildExtensionsPath)\
$(MSBuildToolsVersion)\Microsoft.Common.props"
  Condition="Exists('$(MSBuildExtensionsPath)\
$(MSBuildToolsVersion)\Microsoft.Common.props')" />
  <PropertyGroup>
    <Configuration Condition=" '$(Configuration)' == '' ">Debug</Configuration>
    <Platform Condition=" '$(Platform)' == '' ">AnyCPU</Platform>
    <SchemaVersion>2.0</SchemaVersion>
    <ProjectGuid>d60939b6-526d-43f4-9a89-577b2980df62</ProjectGuid>
    <OutputType>Exe</OutputType>
    <RootNamespace>FSharpHelloWorld</RootNamespace>
    <AssemblyName>FSharpHelloWorld</AssemblyName>
    <TargetFrameworkVersion>v4.8</TargetFrameworkVersion>
    <AutoGenerateBindingRedirects>true</AutoGenerateBindingRedirects>
    <TargetFSharpCoreVersion>4.4.0.0</TargetFSharpCoreVersion>
    <Name>FSharpHelloWorld</Name>
  </PropertyGroup>
  <PropertyGroup Condition=" '$(Configuration)|$(Platform)' == 'Debug|AnyCPU' ">
    <DebugSymbols>true</DebugSymbols>
    <DebugType>full</DebugType>
    <Optimize>>false</Optimize>
    <Tailcalls>>false</Tailcalls>
    <OutputPath>bin\Debug\<</OutputPath>
    <DefineConstants>DEBUG;TRACE</DefineConstants>
```

```

    <WarningLevel>3</WarningLevel>
    <PlatformTarget>AnyCPU</PlatformTarget>
    <DocumentationFile>bin\Debug\FSharpHelloWorld.XML</DocumentationFile>
    <Prefer32Bit>true</Prefer32Bit>
</PropertyGroup>
<PropertyGroup Condition=" '$(Configuration)|$(Platform)' == 'Release|AnyCPU' ">
    <DebugType>pdbonly</DebugType>
    <Optimize>true</Optimize>
    <Tailcalls>true</Tailcalls>
    <OutputPath>bin\Release\</OutputPath>
    <DefineConstants>TRACE</DefineConstants>
    <WarningLevel>3</WarningLevel>
    <PlatformTarget>AnyCPU</PlatformTarget>
    <DocumentationFile>bin\Release\FSharpHelloWorld.XML</DocumentationFile>
    <Prefer32Bit>true</Prefer32Bit>
</PropertyGroup>
<ItemGroup>
    <Reference Include="mscorlib" />
    <Reference Include="FSharp.Core, Version=$(TargetFSharpCoreVersion),
Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a">
        <Private>True</Private>
    </Reference>
    <Reference Include="System" />
    <Reference Include="System.Core" />
    <Reference Include="System.Numerics" />
</ItemGroup>
<ItemGroup>
    <Compile Include="AssemblyInfo.fs" />
    <Compile Include="Program.fs" />
    <None Include="App.config" />
</ItemGroup>
<PropertyGroup>
    <MinimumVisualStudioVersion Condition="'$(MinimumVisualStudioVersion)' == ''>11</
MinimumVisualStudioVersion>
</PropertyGroup>
<Choose>
    <When Condition="'$(VisualStudioVersion)' == '11.0'">
        <PropertyGroup Condition="Exists('$(MSBuildExtensionsPath32)\..\Microsoft SDKs\F#
\3.0\Framework\v4.0\Microsoft.FSharp.Targets')">
            <FSharpTargetsPath>$(MSBuildExtensionsPath32)\..\Microsoft SDKs\F#
\3.0\Framework\v4.0\Microsoft.FSharp.Targets</FSharpTargetsPath>
        </PropertyGroup>
    </When>
    <Otherwise>

```

```

    <PropertyGroup Condition="Exists('$(MSBuildExtensionsPath32)\Microsoft
\VisualStudio\v$(VisualStudioVersion)\FSharp\Microsoft.FSharp.Targets')">
      <FSharpTargetsPath>$(MSBuildExtensionsPath32)\Microsoft\VisualStudio\v
$(VisualStudioVersion)\FSharp\Microsoft.FSharp.Targets</FSharpTargetsPath>
    </PropertyGroup>
  </Otherwise>
</Choose>
<Import Project="$(FSharpTargetsPath)" />
<!-- To modify your build process, add your task inside one of the targets below and
uncomment it.
    Other similar extension points exist, see Microsoft.Common.targets.
<Target Name="BeforeBuild">
</Target>
<Target Name="AfterBuild">
</Target>
-->
</Project>

```

Program.fs(dalam(*root directory name*)\FSharpHelloWorld):

```

// Learn more about F# at http://fsharp.org
// See the 'F# Tutorial' project for more help.

[<EntryPoint>]
let main argv =
    printfn "Hello World"
    0 // return an integer exit code

```

Visual Basic dan .NET Framework

buildspec.yml(dalam(*root directory name*)):

```

version: 0.2

env:
  variables:
    SOLUTION: .\VBHelloWorld.sln
    PACKAGE_DIRECTORY: .\packages
    DOTNET_FRAMEWORK: 4.8

phases:
  build:
    commands:

```



```

- '& "C:\ProgramData\chocolatey\bin\NuGet.exe" restore $env:SOLUTION -
PackagesDirectory $env:PACKAGE_DIRECTORY'
- '& "C:\Program Files (x86)\MSBuild\14.0\Bin\MSBuild.exe" -
p:FrameworkPathOverride="C:\Program Files (x86)\Reference Assemblies\Microsoft
\Framework\.NETFramework\v$env:DOTNET_FRAMEWORK" $env:SOLUTION'
artifacts:
  files:
    - .\VBHelloWorld\bin\Debug\*

```

VBHelloWorld.sln(dalam(*root directory name*)):

```

Microsoft Visual Studio Solution File, Format Version 12.00
# Visual Studio 14
VisualStudioVersion = 14.0.25420.1
MinimumVisualStudioVersion = 10.0.40219.1
Project("{F184B08F-C81C-45F6-A57F-5ABD9991F28F}") = "VBHelloWorld", "VBHelloWorld
\VBHelloWorld.vbproj", "{4DCEC446-7156-4FE6-8CCC-219E34DD409D}"
EndProject
Global
  GlobalSection(SolutionConfigurationPlatforms) = preSolution
    Debug|Any CPU = Debug|Any CPU
    Release|Any CPU = Release|Any CPU
  EndGlobalSection
  GlobalSection(ProjectConfigurationPlatforms) = postSolution
    {4DCEC446-7156-4FE6-8CCC-219E34DD409D}.Debug|Any CPU.ActiveCfg = Debug|Any CPU
    {4DCEC446-7156-4FE6-8CCC-219E34DD409D}.Debug|Any CPU.Build.0 = Debug|Any CPU
    {4DCEC446-7156-4FE6-8CCC-219E34DD409D}.Release|Any CPU.ActiveCfg = Release|Any CPU
    {4DCEC446-7156-4FE6-8CCC-219E34DD409D}.Release|Any CPU.Build.0 = Release|Any CPU
  EndGlobalSection
  GlobalSection(SolutionProperties) = preSolution
    HideSolutionNode = FALSE
  EndGlobalSection
EndGlobal

```

App.config(dalam(*root directory name*)\VBHelloWorld):

```

<?xml version="1.0" encoding="utf-8" ?>
<configuration>
  <startup>
    <supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.8" />
  </startup>
</configuration>

```

HelloWorld.vb(dalam(*root directory name*)\VBHelloWorld):

```
Module HelloWorld

    Sub Main()
        MsgBox("Hello World")
    End Sub

End Module
```

VBHelloWorld.vbproj(dalam(*root directory name*)\VBHelloWorld):

```
<?xml version="1.0" encoding="utf-8"?>
<Project ToolsVersion="14.0" DefaultTargets="Build" xmlns="http://
schemas.microsoft.com/developer/msbuild/2003">
  <Import Project="$(MSBuildExtensionsPath)\
$(MSBuildToolsVersion)\Microsoft.Common.props"
  Condition="Exists('$(MSBuildExtensionsPath)\
$(MSBuildToolsVersion)\Microsoft.Common.props')" />
  <PropertyGroup>
    <Configuration Condition=" '$(Configuration)' == '' ">Debug</Configuration>
    <Platform Condition=" '$(Platform)' == '' ">AnyCPU</Platform>
    <ProjectGuid>{4DCEC446-7156-4FE6-8CCC-219E34DD409D}</ProjectGuid>
    <OutputType>Exe</OutputType>
    <StartupObject>VBHelloWorld.HelloWorld</StartupObject>
    <RootNamespace>VBHelloWorld</RootNamespace>
    <AssemblyName>VBHelloWorld</AssemblyName>
    <FileAlignment>512</FileAlignment>
    <MyType>Console</MyType>
    <TargetFrameworkVersion>v4.8</TargetFrameworkVersion>
    <AutoGenerateBindingRedirects>true</AutoGenerateBindingRedirects>
  </PropertyGroup>
  <PropertyGroup Condition=" '$(Configuration)|$(Platform)' == 'Debug|AnyCPU' ">
    <PlatformTarget>AnyCPU</PlatformTarget>
    <DebugSymbols>true</DebugSymbols>
    <DebugType>full</DebugType>
    <DefineDebug>true</DefineDebug>
    <DefineTrace>true</DefineTrace>
    <OutputPath>bin\Debug\<</OutputPath>
    <DocumentationFile>VBHelloWorld.xml</DocumentationFile>
    <NoWarn>42016,41999,42017,42018,42019,42032,42036,42020,42021,42022</NoWarn>
  </PropertyGroup>
  <PropertyGroup Condition=" '$(Configuration)|$(Platform)' == 'Release|AnyCPU' ">
```

```
<PlatformTarget>AnyCPU</PlatformTarget>
<DebugType>pdbonly</DebugType>
<DefineDebug>>false</DefineDebug>
<DefineTrace>>true</DefineTrace>
<Optimize>>true</Optimize>
<OutputPath>bin\Release\<</OutputPath>
<DocumentationFile>VBHelloWorld.xml</DocumentationFile>
<NoWarn>42016,41999,42017,42018,42019,42032,42036,42020,42021,42022</NoWarn>
</PropertyGroup>
<PropertyGroup>
  <OptionExplicit>On</OptionExplicit>
</PropertyGroup>
<PropertyGroup>
  <OptionCompare>Binary</OptionCompare>
</PropertyGroup>
<PropertyGroup>
  <OptionStrict>Off</OptionStrict>
</PropertyGroup>
<PropertyGroup>
  <OptionInfer>On</OptionInfer>
</PropertyGroup>
<ItemGroup>
  <Reference Include="System" />
  <Reference Include="System.Data" />
  <Reference Include="System.Deployment" />
  <Reference Include="System.Xml" />
  <Reference Include="System.Core" />
  <Reference Include="System.Xml.Linq" />
  <Reference Include="System.Data.DataSetExtensions" />
  <Reference Include="System.Net.Http" />
</ItemGroup>
<ItemGroup>
  <Import Include="Microsoft.VisualBasic" />
  <Import Include="System" />
  <Import Include="System.Collections" />
  <Import Include="System.Collections.Generic" />
  <Import Include="System.Data" />
  <Import Include="System.Diagnostics" />
  <Import Include="System.Linq" />
  <Import Include="System.Xml.Linq" />
  <Import Include="System.Threading.Tasks" />
</ItemGroup>
<ItemGroup>
  <Compile Include="HelloWorld.vb" />
</ItemGroup>
```

```
<Compile Include="My Project\AssemblyInfo.vb" />
<Compile Include="My Project\Application.Designer.vb">
  <AutoGen>True</AutoGen>
  <DependentUpon>Application.myapp</DependentUpon>
</Compile>
<Compile Include="My Project\Resources.Designer.vb">
  <AutoGen>True</AutoGen>
  <DesignTime>True</DesignTime>
  <DependentUpon>Resources.resx</DependentUpon>
</Compile>
<Compile Include="My Project\Settings.Designer.vb">
  <AutoGen>True</AutoGen>
  <DependentUpon>Settings.settings</DependentUpon>
  <DesignTimeSharedInput>True</DesignTimeSharedInput>
</Compile>
</ItemGroup>
<ItemGroup>
  <EmbeddedResource Include="My Project\Resources.resx">
    <Generator>VbMyResourcesResXFileCodeGenerator</Generator>
    <LastGenOutput>Resources.Designer.vb</LastGenOutput>
    <CustomToolNamespace>My.Resources</CustomToolNamespace>
    <SubType>Designer</SubType>
  </EmbeddedResource>
</ItemGroup>
<ItemGroup>
  <None Include="My Project\Application.myapp">
    <Generator>MyApplicationCodeGenerator</Generator>
    <LastGenOutput>Application.Designer.vb</LastGenOutput>
  </None>
  <None Include="My Project\Settings.settings">
    <Generator>SettingsSingleFileGenerator</Generator>
    <CustomToolNamespace>My</CustomToolNamespace>
    <LastGenOutput>Settings.Designer.vb</LastGenOutput>
  </None>
  <None Include="App.config" />
</ItemGroup>
<Import Project="$(MSBuildToolsPath)\Microsoft.VisualBasic.targets" />
<!-- To modify your build process, add your task inside one of the targets below and
uncomment it.
    Other similar extension points exist, see Microsoft.Common.targets.
<Target Name="BeforeBuild">
</Target>
<Target Name="AfterBuild">
</Target>
```

```
-->
</Project>
```

Application.Designer.vb(dalam(*root directory name*)\VBHelloWorld\My Project):

```
'-----
' <auto-generated>
'   This code was generated by a tool.
'   Runtime Version:4.0.30319.42000
'
'   Changes to this file may cause incorrect behavior and will be lost if
'   the code is regenerated.
' </auto-generated>
'-----

Option Strict On
Option Explicit On
```

Application.myapp(dalam(*root directory name*)\VBHelloWorld\My Project):

```
<?xml version="1.0" encoding="utf-8"?>
<MyApplicationData xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <MySubMain>false</MySubMain>
  <SingleInstance>false</SingleInstance>
  <ShutdownMode>0</ShutdownMode>
  <EnableVisualStyles>true</EnableVisualStyles>
  <AuthenticationMode>0</AuthenticationMode>
  <ApplicationType>2</ApplicationType>
  <SaveMySettingsOnExit>true</SaveMySettingsOnExit>
</MyApplicationData>
```

AssemblyInfo.vb(dalam(*root directory name*)\VBHelloWorld\My Project):

```
Imports System
Imports System.Reflection
Imports System.Runtime.InteropServices

' General Information about an assembly is controlled through the following
' set of attributes. Change these attribute values to modify the information
' associated with an assembly.
```

```
' Review the values of the assembly attributes

<Assembly: AssemblyTitle("VBHelloWorld")>
<Assembly: AssemblyDescription("")>
<Assembly: AssemblyCompany("")>
<Assembly: AssemblyProduct("VBHelloWorld")>
<Assembly: AssemblyCopyright("Copyright © 2017")>
<Assembly: AssemblyTrademark("")>

<Assembly: ComVisible(False)>

'The following GUID is for the ID of the typelib if this project is exposed to COM
<Assembly: Guid("137c362b-36ef-4c3e-84ab-f95082487a5a")>

' Version information for an assembly consists of the following four values:
'
' Major Version
' Minor Version
' Build Number
' Revision
'
' You can specify all the values or you can default the Build and Revision Numbers
' by using the '*' as shown below:
' <Assembly: AssemblyVersion("1.0.*")>

<Assembly: AssemblyVersion("1.0.0.0")>
<Assembly: AssemblyFileVersion("1.0.0.0")>
```

Resources.Designer.vb(dalam(*root directory name*)\VBHelloWorld\My Project):

```
'-----
' <auto-generated>
' This code was generated by a tool.
' Runtime Version:4.0.30319.42000
'
' Changes to this file may cause incorrect behavior and will be lost if
' the code is regenerated.
' </auto-generated>
'-----

Option Strict On
Option Explicit On
```

Namespace My.Resources

```

'This class was auto-generated by the StronglyTypedResourceBuilder
'class via a tool like ResGen or Visual Studio.
'To add or remove a member, edit your .ResX file then rerun ResGen
'with the /str option, or rebuild your VS project.
'''<summary>
''' A strongly-typed resource class, for looking up localized strings, etc.
'''</summary>

<Global.System.CodeDom.Compiler.GeneratedCodeAttribute("System.Resources.Tools.StronglyTypedRe
"4.0.0.0"), _
Global.System.Diagnostics.DebuggerNonUserCodeAttribute(), _
Global.System.Runtime.CompilerServices.CompilerGeneratedAttribute(), _
Global.Microsoft.VisualBasic.HideModuleNameAttribute()> _
Friend Module Resources

    Private resourceMan As Global.System.Resources.ResourceManager

    Private resourceCulture As Global.System.Globalization.CultureInfo

    '''<summary>
    ''' Returns the cached ResourceManager instance used by this class.
    '''</summary>

<Global.System.ComponentModel.EditorBrowsableAttribute(Global.System.ComponentModel.EditorBrow
-
    Friend ReadOnly Property ResourceManager() As
Global.System.Resources.ResourceManager
    Get
        If Object.ReferenceEquals(resourceMan, Nothing) Then
            Dim temp As Global.System.Resources.ResourceManager = New
Global.System.Resources.ResourceManager("VBHelloWorld.Resources",
GetType(Resources).Assembly)
            resourceMan = temp
        End If
        Return resourceMan
    End Get
End Property

    '''<summary>
    ''' Overrides the current thread's CurrentUICulture property for all
    ''' resource lookups using this strongly typed resource class.
    '''</summary>

```

```

<Global.System.ComponentModel.EditorBrowsableAttribute(Global.System.ComponentModel.EditorBrowsable
-
    Friend Property Culture() As Global.System.Globalization.CultureInfo
        Get
            Return resourceCulture
        End Get
        Set(ByVal value As Global.System.Globalization.CultureInfo)
            resourceCulture = value
        End Set
    End Property
End Module
End Namespace

```

Resources.resx(dalam(*root directory name*)\VBHelloWorld\My Project):

```

<?xml version="1.0" encoding="utf-8"?>
<root>
  <!--

```

Microsoft ResX Schema

Version 2.0

The primary goals of this format is to allow a simple XML format that is mostly human readable. The generation and parsing of the various data types are done through the TypeConverter classes associated with the data types.

Example:

... ado.net/XML headers & schema ...

```
<resheader name="resmimetype">text/microsoft-resx</resheader>
```

```
<resheader name="version">2.0</resheader>
```

```
<resheader name="reader">System.Resources.ResXResourceReader,
System.Windows.Forms, ...</resheader>
```

```
<resheader name="writer">System.Resources.ResXResourceWriter,
System.Windows.Forms, ...</resheader>
```

```
<data name="Name1"><value>this is my long string</value><comment>this is a
comment</comment></data>
```

```
<data name="Color1" type="System.Drawing.Color, System.Drawing">Blue</data>
```

```
<data name="Bitmap1" mimetype="application/x-microsoft.net.object.binary.base64">
```

```
<value>[base64 mime encoded serialized .NET Framework object]</value>
```

```
</data>
```



```

<data name="Icon1" type="System.Drawing.Icon, System.Drawing"
mimetype="application/x-microsoft.net.object.bytearray.base64">
  <value>[base64 mime encoded string representing a byte array form of the .NET
Framework object]</value>
  <comment>This is a comment</comment>
</data>

```

There are any number of "resheader" rows that contain simple name/value pairs.

Each data row contains a name, and value. The row also contains a type or mimetype. Type corresponds to a .NET class that support text/value conversion through the TypeConverter architecture. Classes that don't support this are serialized and stored with the mimetype set.

The mimetype is used for serialized objects, and tells the ResXResourceReader how to depersist the object. This is currently not extensible. For a given mimetype the value must be set accordingly:

Note - application/x-microsoft.net.object.binary.base64 is the format that the ResXResourceWriter will generate, however the reader can read any of the formats listed below.

```

mimetype: application/x-microsoft.net.object.binary.base64
value   : The object must be serialized with
         : System.Serialization.Formatters.Binary.BinaryFormatter
         : and then encoded with base64 encoding.

```

```

mimetype: application/x-microsoft.net.object.soap.base64
value   : The object must be serialized with
         : System.Runtime.Serialization.Formatters.Soap.SoapFormatter
         : and then encoded with base64 encoding.

```

```

mimetype: application/x-microsoft.net.object.bytearray.base64
value   : The object must be serialized into a byte array
         : using a System.ComponentModel.TypeConverter
         : and then encoded with base64 encoding.

```

```
-->
```

```

<xsd:schema id="root" xmlns="" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:msdata="urn:schemas-microsoft-com:xml-msdata">
  <xsd:element name="root" msdata:IsDataSet="true">
    <xsd:complexType>
      <xsd:choice maxOccurs="unbounded">

```

```

<xsd:element name="metadata">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="value" type="xsd:string" minOccurs="0" />
    </xsd:sequence>
    <xsd:attribute name="name" type="xsd:string" />
    <xsd:attribute name="type" type="xsd:string" />
    <xsd:attribute name="mimetype" type="xsd:string" />
  </xsd:complexType>
</xsd:element>
<xsd:element name="assembly">
  <xsd:complexType>
    <xsd:attribute name="alias" type="xsd:string" />
    <xsd:attribute name="name" type="xsd:string" />
  </xsd:complexType>
</xsd:element>
<xsd:element name="data">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="value" type="xsd:string" minOccurs="0"
msdata:Ordinal="1" />
      <xsd:element name="comment" type="xsd:string" minOccurs="0"
msdata:Ordinal="2" />
    </xsd:sequence>
    <xsd:attribute name="name" type="xsd:string" msdata:Ordinal="1" />
    <xsd:attribute name="type" type="xsd:string" msdata:Ordinal="3" />
    <xsd:attribute name="mimetype" type="xsd:string" msdata:Ordinal="4" />
  </xsd:complexType>
</xsd:element>
<xsd:element name="resheader">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="value" type="xsd:string" minOccurs="0"
msdata:Ordinal="1" />
    </xsd:sequence>
    <xsd:attribute name="name" type="xsd:string" use="required" />
  </xsd:complexType>
</xsd:element>
</xsd:choice>
</xsd:complexType>
</xsd:element>
</xsd:schema>
<resheader name="resmimetype">
  <value>text/microsoft-resx</value>

```

```

</resheader>
<resheader name="version">
  <value>2.0</value>
</resheader>
<resheader name="reader">
  <value>System.Resources.ResXResourceReader, System.Windows.Forms, Version=2.0.0.0,
Culture=neutral, PublicKeyToken=b77a5c561934e089</value>
</resheader>
<resheader name="writer">
  <value>System.Resources.ResXResourceWriter, System.Windows.Forms, Version=2.0.0.0,
Culture=neutral, PublicKeyToken=b77a5c561934e089</value>
</resheader>
</root>

```

Settings.Designer.vb(dalam(*root directory name*)\VBHelloWorld\My Project):

```

' -----
' <auto-generated>
'   This code was generated by a tool.
'   Runtime Version:4.0.30319.42000
'
'   Changes to this file may cause incorrect behavior and will be lost if
'   the code is regenerated.
' </auto-generated>
' -----

Option Strict On
Option Explicit On

Namespace My

  <Global.System.Runtime.CompilerServices.CompilerGeneratedAttribute(), _
Global.System.CodeDom.Compiler.GeneratedCodeAttribute("Microsoft.VisualStudio.Editors.Settings
"11.0.0.0"), _
Global.System.ComponentModel.EditorBrowsableAttribute(Global.System.ComponentModel.EditorBrowsable
-
  Partial Friend NotInheritable Class MySettings
    Inherits Global.System.Configuration.ApplicationSettingsBase

```

```

Private Shared defaultInstance As MySettings =
CType(Global.System.Configuration.ApplicationSettingsBase.Synchronized(New
MySettings), MySettings)

#Region "My.Settings Auto-Save Functionality"
  #If _MyType = "WindowsForms" Then
    Private Shared addedHandler As Boolean

    Private Shared addedHandlerLockObject As New Object

    <Global.System.Diagnostics.DebuggerNonUserCodeAttribute(),
Global.System.ComponentModel.EditorBrowsableAttribute(Global.System.ComponentModel.EditorBrowsable
-
    Private Shared Sub AutoSaveSettings(ByVal sender As Global.System.Object, ByVal
e As Global.System.EventArgs)
      If My.Application.SaveMySettingsOnExit Then
        My.Settings.Save()
      End If
    End Sub
  #End If
#End Region

Public Shared ReadOnly Property [Default]() As MySettings
  Get

    #If _MyType = "WindowsForms" Then
      If Not addedHandler Then
        SyncLock addedHandlerLockObject
          If Not addedHandler Then
            AddHandler My.Application.Shutdown, AddressOf AutoSaveSettings
            addedHandler = True
          End If
        End SyncLock
      End If
    #End If
    Return defaultInstance
  End Get
End Property
End Class
End Namespace

Namespace My

  <Global.Microsoft.VisualBasic.HideModuleNameAttribute(), _

```

```
Global.System.Diagnostics.DebuggerNonUserCodeAttribute(), _
Global.System.Runtime.CompilerServices.CompilerGeneratedAttribute())> _
Friend Module MySettingsProperty

    <Global.System.ComponentModel.Design.HelpKeywordAttribute("My.Settings")> _
    Friend ReadOnly Property Settings() As Global.VBHelloWorld.My.MySettings
        Get
            Return Global.VBHelloWorld.My.MySettings.Default
        End Get
    End Property
End Module
End Namespace
```

Settings.settings(dalam(*root directory name*)\VBHelloWorld\My Project):

```
<?xml version='1.0' encoding='utf-8'?>
<SettingsFile xmlns="http://schemas.microsoft.com/VisualStudio/2004/01/settings"
CurrentProfile="(Default)" UseMySettingsClassName="true">
  <Profiles>
    <Profile Name="(Default)" />
  </Profiles>
  <Settings />
</SettingsFile>
```

Rencanakan build in AWS CodeBuild

Sebelum Anda menggunakan AWS CodeBuild, Anda harus menjawab pertanyaan-pertanyaan ini:

1. Di mana kode sumber disimpan? CodeBuild saat ini mendukung pembangunan dari penyedia repositori kode sumber berikut. Kode sumber harus berisi file spesifikasi build (buildspec). Buildspec adalah kumpulan perintah build dan pengaturan terkait, dalam format YAMAL, yang CodeBuild digunakan untuk menjalankan build. Anda dapat mendeklarasikan buildspec dalam definisi proyek build.

Penyedia repositori	Diperlukan	Dokumentasi
CodeCommit	Nama repositori. (Opsional) Komit ID yang terkait dengan kode sumber.	Lihat topik ini di Panduan AWS CodeCommit Pengguna: Buat CodeCommit repositori Buat komit di CodeCommit
Amazon S3	Masukkan nama bucket. Nama objek yang sesuai dengan file ZIP input build yang berisi kode sumber. (Opsional) ID Versi yang terkait dengan file ZIP input build.	Lihat topik ini di Panduan Memulai Amazon S3: Buat bucket Tambahkan objek ke bucket
GitHub	Nama repositori.	Lihat topik ini di situs web GitHub Bantuan:

Penyedia repositori	Diperlukan	Dokumentasi
	(Opsional) Komit ID yang terkait dengan kode sumber.	Buat repo
Bitbucket	Nama repositori. (Opsional) Komit ID yang terkait dengan kode sumber.	Lihat topik ini di situs web dokumentasi Bitbucket Cloud: Buat repositori

- Perintah build mana yang perlu Anda jalankan dan dalam urutan apa? Secara default, CodeBuild mengunduh input build dari penyedia yang Anda tentukan dan mengunggah output build ke bucket yang Anda tentukan. Anda menggunakan buildspec untuk menginstruksikan cara mengubah input build yang diunduh menjadi output build yang diharapkan. Lihat informasi yang lebih lengkap di [Referensi Buildspec](#).
- Runtime dan alat apa yang Anda perlukan untuk menjalankan build? Misalnya, apakah Anda membangun untuk Java, Ruby, Python, atau Node.js? Apakah build membutuhkan Maven atau Ant atau kompiler untuk Java, Ruby, atau Python? Apakah build membutuhkan Git, theAWS CLI, atau alat lainnya?

CodeBuild menjalankan build di lingkungan build yang menggunakan gambar Docker. Gambar Docker ini harus disimpan dalam jenis repositori yang didukung oleh CodeBuild. Ini termasuk repositori image CodeBuild Docker, Docker Hub, dan Amazon Elastic Container Registry (Amazon ECR) Registry (Amazon ECR). Untuk informasi selengkapnya tentang repositori gambar CodeBuild Docker, lihat [Gambar Docker disediakan oleh CodeBuild](#)

- Apakah Anda memerlukan AWS sumber daya yang tidak disediakan secara otomatis oleh CodeBuild? Jika demikian, kebijakan keamanan apa yang dibutuhkan sumber daya tersebut? Misalnya, Anda mungkin perlu memodifikasi peran CodeBuild layanan agar memungkinkan CodeBuild untuk bekerja dengan sumber daya tersebut.
- Apakah Anda CodeBuild ingin bekerja dengan VPC Anda? Jika demikian, Anda memerlukan ID VPC, ID subnet, dan ID grup keamanan untuk konfigurasi VPC Anda. Untuk informasi selengkapnya, lihat [GunakanAWS CodeBuilddengan Amazon Virtual Private Cloud](#).

Setelah Anda menjawab pertanyaan-pertanyaan ini, Anda harus memiliki pengaturan dan sumber daya yang Anda butuhkan untuk menjalankan build dengan sukses. Untuk menjalankan build Anda, Anda dapat:

- Gunakan AWS CodeBuild konsol, AWS CLI, atau AWS SDK. Untuk informasi selengkapnya, lihat [Jalankan build](#).
- Buat atau identifikasi pipeline AWS CodePipeline, lalu tambahkan tindakan build atau test yang menginstruksikan CodeBuild untuk menguji kode secara otomatis, menjalankan build, atau keduanya. Lihat informasi yang lebih lengkap di [Gunakan CodePipeline dengan CodeBuild](#).

Buat referensi spesifikasi untuk CodeBuild

Topik ini memberikan informasi referensi penting tentang file spesifikasi build (buildspec). Buildspec adalah kumpulan perintah build dan pengaturan terkait, dalam format YAML, yang CodeBuild digunakan untuk menjalankan build. Anda dapat menyertakan buildspec sebagai bagian dari kode sumber atau Anda dapat menentukan buildspec saat membuat proyek build. Untuk informasi tentang cara kerja spesifikasi build, lihat [Cara kerja CodeBuild](#).

Topik

- [Nama file Buildspec dan lokasi penyimpanan](#)
- [Sintaks Buildspec](#)
- [Contoh Buildspec](#)
- [Versi Buildspec](#)
- [Batch membangun referensi buildspec](#)
- [GitHubReferensi buildspec pelari aksi](#)

Nama file Buildspec dan lokasi penyimpanan

Jika Anda menyertakan buildspec sebagai bagian dari kode sumber, secara default, file buildspec harus diberi nama `buildspec.yml` dan ditempatkan di root direktori sumber Anda.

Anda dapat mengganti nama dan lokasi file buildspec default. Sebagai contoh, Anda dapat:

- Gunakan file buildspec yang berbeda untuk build yang berbeda di repositori yang sama, seperti `dan.buildspec_debug.yml` `buildspec_release.yml`

- Simpan file buildspec di tempat lain selain root direktori sumber Anda, seperti `config/buildspec.yml` atau di bucket S3. Bucket S3 harus berada di AWS Wilayah yang sama dengan proyek build Anda. Tentukan file buildspec menggunakan ARN (misalnya, `arn:aws:s3:::<my-codebuild-sample2>/buildspec.yml`)

Anda hanya dapat menentukan satu buildspec untuk proyek build, terlepas dari nama file buildspec.

Untuk mengganti nama file buildspec default, lokasi, atau keduanya, lakukan salah satu hal berikut:

- Jalankan `update-project` perintah AWS CLI `create-project` or, atur buildspec nilai ke jalur ke file buildspec alternatif relatif terhadap nilai variabel lingkungan bawaan. `CODEBUILD_SRC_DIR` Anda juga dapat melakukan hal yang setara dengan `create project` operasi di AWS SDK. Untuk informasi selengkapnya, lihat [Membuat proyek pembangunan](#) atau [Mengubah pengaturan proyek pembangunan](#).
- Jalankan AWS CLI `start-build` perintah, atur `buildspecOverride` nilai ke jalur ke file buildspec alternatif relatif terhadap nilai variabel lingkungan bawaan. `CODEBUILD_SRC_DIR` Anda juga dapat melakukan hal yang setara dengan `start build` operasi di AWS SDK. Untuk informasi selengkapnya, lihat [Jalankan build](#).
- Dalam AWS CloudFormation template, atur `BuildSpec` properti `Source` dalam sumber daya tipe `AWS::CodeBuild::Project` ke jalur ke file buildspec alternatif relatif terhadap nilai variabel lingkungan bawaan. `CODEBUILD_SRC_DIR` Untuk informasi selengkapnya, lihat `BuildSpec` properti di [sumber AWS CodeBuild proyek](#) di Panduan AWS CloudFormation Pengguna.

Sintaks Buildspec

[File buildspec harus diekspresikan dalam format YAMB.](#)

Jika perintah berisi karakter, atau string karakter, yang tidak didukung oleh YAMB, Anda harus melampirkan perintah dalam tanda kutip ("). Perintah berikut diapit tanda kutip karena titik dua (:) diikuti oleh spasi tidak diperbolehkan di YAMAL. Tanda kutip dalam perintah diloloskan (").

```
"export PACKAGE_NAME=$(cat package.json | grep name | head -1 | awk -F: '{ print $2 }' | sed 's/[\",,]//g')"
```

Buildspec memiliki sintaks berikut:

```
version: 0.2
```

```
run-as: Linux-user-name

env:
  shell: shell-tag
  variables:
    key: "value"
    key: "value"
  parameter-store:
    key: "value"
    key: "value"
  exported-variables:
    - variable
    - variable
  secrets-manager:
    key: secret-id:json-key:version-stage:version-id
  git-credential-helper: no | yes

proxy:
  upload-artifacts: no | yes
  logs: no | yes

batch:
  fast-fail: false | true
  # build-list:
  # build-matrix:
  # build-graph:

phases:
  install:
    run-as: Linux-user-name
    on-failure: ABORT | CONTINUE
    runtime-versions:
      runtime: version
      runtime: version
    commands:
      - command
      - command
    finally:
      - command
      - command
    # steps:
  pre_build:
    run-as: Linux-user-name
    on-failure: ABORT | CONTINUE
```

```
commands:  
  - command  
  - command  
finally:  
  - command  
  - command  
# steps:  
build:  
  run-as: Linux-user-name  
  on-failure: ABORT | CONTINUE  
  commands:  
    - command  
    - command  
  finally:  
    - command  
    - command  
  # steps:  
post_build:  
  run-as: Linux-user-name  
  on-failure: ABORT | CONTINUE  
  commands:  
    - command  
    - command  
  finally:  
    - command  
    - command  
  # steps:  
reports:  
  report-group-name-or-arn:  
  files:  
    - location  
    - location  
  base-directory: location  
  discard-paths: no | yes  
  file-format: report-format  
artifacts:  
  files:  
    - location  
    - location  
  name: artifact-name  
  discard-paths: no | yes  
  base-directory: location  
  exclude-paths: excluded paths  
  enable-symlinks: no | yes
```

```

s3-prefix: prefix
secondary-artifacts:
  artifactIdentifier:
    files:
      - location
      - location
    name: secondary-artifact-name
    discard-paths: no | yes
    base-directory: location
  artifactIdentifier:
    files:
      - location
      - location
    discard-paths: no | yes
    base-directory: location
cache:
  paths:
    - path
    - path

```

Buildspec berisi berikut ini:

versi

Pemetaan yang diperlukan. Merupakan versi buildspec. Kami menyarankan Anda menggunakan 0.2.

Note

Meskipun versi 0.1 masih didukung, kami sarankan Anda menggunakan versi 0.2 bila memungkinkan. Untuk informasi selengkapnya, lihat [Versi Buildspec](#).

jalankan-sebagai

Urutan opsional. Hanya tersedia untuk pengguna Linux. Menentukan pengguna Linux yang menjalankan perintah dalam file buildspec ini. `run-as` memberikan izin baca dan jalankan pengguna yang ditentukan. Saat Anda menentukan `run-as` di bagian atas file buildspec, itu berlaku secara global untuk semua perintah. Jika Anda tidak ingin menentukan pengguna untuk semua perintah file buildspec, Anda dapat menentukan satu untuk perintah dalam satu fase dengan menggunakan `run-as` di salah satu blok `phases`. Jika tidak `run-as` ditentukan, maka semua perintah berjalan sebagai pengguna `root`.

env

Urutan opsional. Merupakan informasi untuk satu atau lebih variabel lingkungan kustom.

Note

Untuk melindungi informasi sensitif, berikut ini disembunyikan di CodeBuild log:

- AWS ID kunci akses. Untuk informasi selengkapnya, lihat [Mengelola Kunci Akses untuk Pengguna IAM](#) di Panduan AWS Identity and Access Management Pengguna.
- String ditentukan menggunakan Parameter Store. Untuk informasi selengkapnya, lihat [Systems Manager Parameter Store](#) dan [Systems Manager Parameter Store Console Walkthrough](#) di Panduan Pengguna Amazon EC2 Systems Manager.
- String ditentukan menggunakan AWS Secrets Manager. Untuk informasi selengkapnya, lihat [Manajemen kunci](#).

env/cangkang

Urutan opsional. Menentukan shell yang didukung untuk sistem operasi Linux atau Windows.

Untuk sistem operasi Linux, tag shell yang didukung adalah:

- bash
- /bin/sh

Untuk sistem operasi Windows, tag shell yang didukung adalah:

- powershell.exe
- cmd.exe

env/variabel

Diperlukan jika env ditentukan, dan Anda ingin menentukan variabel lingkungan kustom dalam teks biasa. Berisi pemetaan skalar *kunci/nilai*, di mana setiap pemetaan mewakili variabel lingkungan kustom tunggal dalam teks biasa. *kunci* adalah nama variabel lingkungan kustom, dan *nilai adalah nilai* variabel itu.

Important

Kami sangat tidak menyarankan penyimpanan nilai sensitif dalam variabel lingkungan. Variabel lingkungan dapat ditampilkan dalam teks biasa menggunakan alat seperti

CodeBuild konsol dan file AWS CLI. Untuk nilai sensitif, sebaiknya gunakan parameter-store atau secrets-manager pemetaan sebagai gantinya, seperti yang dijelaskan nanti di bagian ini.

Setiap variabel lingkungan yang Anda tetapkan menggantikan variabel lingkungan yang ada. Misalnya, jika image Docker sudah berisi variabel lingkungan bernama MY_VAR dengan nilai `my_value`, dan Anda menetapkan variabel lingkungan bernama MY_VAR dengan nilai `other_value`, `my_value` maka diganti dengan `other_value`. Demikian pula, jika image Docker sudah berisi variabel lingkungan bernama PATH dengan nilai `/usr/local/sbin:/usr/local/bin`, dan Anda menetapkan variabel lingkungan bernama PATH dengan nilai `$PATH:/usr/share/ant/bin,/usr/local/sbin:/usr/local/bin` maka digantikan oleh nilai `$PATH:/usr/share/ant/bin` literal.

Jangan mengatur variabel lingkungan apa pun dengan nama yang dimulai dengan `CODEBUILD_`. Awalan ini dicadangkan untuk penggunaan internal.

Jika variabel lingkungan dengan nama yang sama didefinisikan di beberapa tempat, nilainya ditentukan sebagai berikut:

- Nilai dalam panggilan operasi start build lebih diutamakan. Anda dapat menambahkan atau mengganti variabel lingkungan saat membuat build. Untuk informasi selengkapnya, lihat [Jalankan build diAWS CodeBuild](#).
- Nilai dalam definisi proyek build akan diutamakan berikutnya. Anda dapat menambahkan variabel lingkungan di tingkat proyek saat Anda membuat atau mengedit proyek. Lihat informasi yang lebih lengkap di [Membuat proyek pembangunan diAWS CodeBuild](#) dan [Mengubah pengaturan proyek build diAWS CodeBuild](#).
- Nilai dalam deklarasi buildspec diutamakan paling rendah.

env/parameter-toko

Diperlukan jika env ditentukan, dan Anda ingin mengambil variabel lingkungan kustom yang disimpan di Amazon EC2 Systems Manager Parameter Store. Berisi pemetaan skalar *key/nilai*, di mana setiap pemetaan mewakili variabel lingkungan kustom tunggal yang disimpan di Amazon EC2 Systems Manager Parameter Store. *key* adalah nama yang Anda gunakan nanti dalam perintah build untuk merujuk ke variabel lingkungan kustom ini, dan *value* adalah nama variabel lingkungan kustom yang disimpan di Amazon EC2 Systems Manager Parameter Store. Untuk menyimpan nilai sensitif, lihat [Systems Manager Parameter Store](#) and [Walkthrough: Membuat dan menguji parameter String \(konsol\)](#) di Panduan Pengguna Amazon EC2 Systems Manager.

⚠ Important

CodeBuild Untuk memungkinkan mengambil variabel lingkungan kustom yang disimpan di Amazon EC2 Systems Manager Parameter Store, Anda harus menambahkan tindakan `ssm:GetParameters` ke peran layanan CodeBuild Anda. Untuk informasi selengkapnya, lihat [Membuat peran CodeBuild layanan](#).

Variabel lingkungan apa pun yang Anda ambil dari Amazon EC2 Systems Manager Parameter Store menggantikan variabel lingkungan yang ada. Misalnya, jika image Docker sudah berisi variabel lingkungan bernama `MY_VAR` dengan nilai `my_value`, dan Anda mengambil variabel lingkungan bernama `MY_VAR` dengan nilai `other_value`, maka `my_value` diganti dengan `other_value`. Demikian pula, jika image Docker sudah berisi variabel lingkungan bernama `PATH` dengan nilai `/usr/local/sbin:/usr/local/bin`, dan Anda mengambil variabel lingkungan bernama `PATH` dengan nilai `$PATH:/usr/share/ant/bin`, maka digantikan `/usr/local/sbin:/usr/local/bin` oleh nilai literal `$PATH:/usr/share/ant/bin`.

Jangan menyimpan variabel lingkungan apa pun dengan nama yang dimulai dengan `CODEBUILD_`. Awalan ini dicadangkan untuk penggunaan internal.

Jika variabel lingkungan dengan nama yang sama didefinisikan di beberapa tempat, nilainya ditentukan sebagai berikut:

- Nilai dalam panggilan operasi start build lebih diutamakan. Anda dapat menambahkan atau mengganti variabel lingkungan saat membuat build. Untuk informasi selengkapnya, lihat [Jalankan build diAWS CodeBuild](#).
- Nilai dalam definisi proyek build akan diutamakan berikutnya. Anda dapat menambahkan variabel lingkungan di tingkat proyek saat Anda membuat atau mengedit proyek. Lihat informasi yang lebih lengkap di [Membuat proyek pembangunan diAWS CodeBuild](#) dan [Mengubah pengaturan proyek build diAWS CodeBuild](#).
- Nilai dalam deklarasi `buildspec` diutamakan paling rendah.

env/rahasia-manajer

Diperlukan jika Anda ingin mengambil variabel lingkungan khusus yang disimpan di AWS Secrets Manager. Tentukan Secrets Manager `reference-key` menggunakan pola berikut:

`<key>: <secret-id>: <json-key>: <version-stage>: <version-id>`

<key>

(Wajib) Nama variabel lingkungan lokal. Gunakan nama ini untuk mengakses variabel selama pembuatan.

<secret-id>

(Wajib) Nama atau Amazon Resource Name (ARN) yang berfungsi sebagai pengidentifikasi unik untuk rahasia tersebut. Untuk mengakses rahasia di AWS akun Anda, cukup tentukan nama rahasia. Untuk mengakses rahasia di AWS akun yang berbeda, tentukan rahasia ARN.

<json-key>

(Opsional) Menentukan nama kunci pasangan nilai kunci Secrets Manager yang nilainya ingin Anda ambil. Jika Anda tidak menentukan `json-key`, CodeBuild mengambil seluruh teks rahasia.

<version-stage>

(Opsional) Menentukan versi rahasia yang ingin Anda ambil dengan label pementasan yang dilampirkan ke versi. Label pementasan digunakan untuk melacak versi yang berbeda selama proses rotasi. Jika Anda menggunakan `version-stage`, jangan tentukan `version-id`. Jika Anda tidak menentukan tahap versi atau ID versi, defaultnya adalah mengambil versi dengan nilai tahap versi. `AWSCURRENT`

<version-id>

(Opsional) Menentukan identifier unik dari versi rahasia yang ingin Anda gunakan. Jika Anda menentukan `version-id`, jangan tentukan `version-stage`. Jika Anda tidak menentukan tahap versi atau ID versi, defaultnya adalah mengambil versi dengan nilai tahap versi. `AWSCURRENT`

Dalam contoh berikut, `TestSecret` adalah nama pasangan kunci-nilai yang disimpan di Secrets Manager. Kuncinya `TestSecret` adalah `MY_SECRET_VAR`. Anda mengakses variabel selama pembuatan menggunakan `LOCAL_SECRET_VAR` nama.

```
env:
  secrets-manager:
    LOCAL_SECRET_VAR: "TestSecret:MY_SECRET_VAR"
```

Untuk informasi selengkapnya, lihat [Apa itu AWS Secrets Manager](#) dalam Panduan Pengguna AWS Secrets Manager .

env/variabel yang diekspor

Pemetaan opsional. Digunakan untuk daftar variabel lingkungan yang ingin Anda ekspor. Tentukan nama setiap variabel yang ingin Anda ekspor pada baris terpisah di bawah `exported-variables`. Variabel yang ingin Anda ekspor harus tersedia di wadah Anda selama pembuatan. Variabel yang Anda ekspor dapat menjadi variabel lingkungan.

Variabel lingkungan yang diekspor digunakan bersama dengan AWS CodePipeline untuk mengekspor variabel lingkungan dari tahap pembuatan saat ini ke tahap selanjutnya dalam pipeline. Untuk informasi selengkapnya, lihat [Bekerja dengan variabel](#) di Panduan AWS CodePipeline Pengguna.

Selama build, nilai variabel tersedia dimulai dengan `install` fase. Ini dapat diperbarui antara awal `install` fase dan akhir `post_build` fase. Setelah `post_build` fase berakhir, nilai variabel yang diekspor tidak dapat berubah.

Note

Berikut ini tidak dapat diekspor:

- Rahasia Penyimpanan Parameter Amazon EC2 Systems Manager yang ditentukan dalam proyek build.
- Rahasia Secrets Manager yang ditentukan dalam proyek build
- Variabel lingkungan yang dimulai dengan `AWS_`.

env/git-credential-helper

Pemetaan opsional. Digunakan untuk menunjukkan jika CodeBuild menggunakan pembantu kredensi Git untuk memberikan kredensi Git. `yes` jika digunakan. Jika tidak, `no` atau tidak ditentukan. Untuk informasi selengkapnya, lihat [gitcredentials](#) di situs web Git.

Note

`git-credential-helper` tidak didukung untuk build yang dipicu oleh webhook untuk repositori Git publik.

proxy

Urutan opsional. Digunakan untuk merepresentasikan setelan jika Anda menjalankan build di server proxy eksplisit. Untuk informasi selengkapnya, lihat [Jalankan CodeBuild di server proxy eksplisit](#).

proxy/upload-artefak

Pemetaan opsional. Setel ke `yes` jika Anda ingin build di server proxy eksplisit untuk mengunggah artefak. Default-nya adalah `no`.

proksi/log

Pemetaan opsional. Setel ke `yes` build Anda di server proxy eksplisit untuk membuat CloudWatch log. Default-nya adalah `no`.

fase

Urutan yang diperlukan. Merupakan perintah yang CodeBuild berjalan selama setiap fase build.

Note

Di `buildspec` versi 0.1, CodeBuild jalankan setiap perintah dalam instance terpisah dari shell default di lingkungan build. Ini berarti bahwa setiap perintah berjalan secara terpisah dari semua perintah lainnya. Oleh karena itu, secara default, Anda tidak dapat menjalankan satu perintah yang bergantung pada status perintah sebelumnya (misalnya, mengubah direktori atau mengatur variabel lingkungan). Untuk mengatasi batasan ini, kami sarankan Anda menggunakan versi 0.2, yang memecahkan masalah ini. Jika Anda harus menggunakan `buildspec` versi 0.1, kami merekomendasikan pendekatannya di [Kerang dan perintah dalam membangun lingkungan](#)

fase/*/ run-as

Urutan opsional. Gunakan dalam fase build untuk menentukan pengguna Linux yang menjalankan perintahnya. Jika juga `run-as` ditentukan secara global untuk semua perintah di bagian atas file `buildspec`, maka pengguna tingkat fase diutamakan. Misalnya, jika secara global `run-as` menentukan `User-1`, dan untuk `install` fase hanya `run-as` pernyataan yang menentukan `User-2`, maka semua perintah di kemudian file `buildspec` dijalankan sebagai `User-1` kecuali perintah dalam fase, yang dijalankan sebagai `User-2`. `install`

fase/*/kegagalan

Urutan opsional. Menentukan tindakan yang harus diambil jika kegagalan terjadi selama fase. Ini dapat berupa salah satu dari nilai berikut:

- ABORT- Batalkan bangunan.
- CONTINUE- Lanjutkan ke fase berikutnya.

Jika properti ini tidak ditentukan, proses kegagalan mengikuti fase transisi seperti yang ditunjukkan pada [Transisi fase](#).

fase/*/ akhirnya

Blok opsional. Perintah yang ditentukan dalam `finally` blok dijalankan setelah perintah di `commands` blok. Perintah dalam `finally` blok dijalankan bahkan jika perintah di `commands` blok gagal. Misalnya, jika `commands` blok berisi tiga perintah dan yang pertama gagal, CodeBuild lewati dua perintah yang tersisa dan jalankan perintah apa pun di `finally` blok. Fase ini berhasil ketika semua perintah di `commands` dan `finally` blok berjalan dengan sukses. Jika ada perintah dalam fase gagal, fase gagal.

Nama fase build yang diizinkan adalah:

fase/instal

Urutan opsional. Merupakan perintah, jika ada, yang CodeBuild berjalan selama instalasi. Kami menyarankan Anda menggunakan `install` fase hanya untuk menginstal paket di lingkungan build. Misalnya, Anda mungkin menggunakan fase ini untuk menginstal kerangka pengujian kode seperti Mocha atau RSpec.

fase/instal/versi runtime

Urutan opsional. Versi runtime didukung dengan Ubuntu standard image 5.0 atau yang lebih baru dan Amazon Linux 2 standard image 4.0 atau yang lebih baru. Jika ditentukan, setidaknya satu runtime harus disertakan dalam bagian ini. Tentukan runtime menggunakan versi tertentu, versi mayor diikuti oleh `.x` untuk menentukan yang CodeBuild menggunakan versi mayor tersebut dengan versi minor terbarunya, atau `latest` untuk menggunakan versi mayor dan minor terbaru (misalnya, `ruby: 3.2` `nodejs: 18.x`, atau `java: latest`). Anda dapat menentukan runtime menggunakan angka atau variabel lingkungan. Misalnya, jika Anda menggunakan Amazon Linux 2 image standar 4.0, maka berikut ini menentukan bahwa versi 17 dari Java, versi minor terbaru dari python versi 3, dan versi yang terkandung dalam variabel

lingkungan Ruby diinstal. Untuk informasi selengkapnya, lihat [Gambar Docker disediakan oleh CodeBuild](#).

```
phases:
  install:
    runtime-versions:
      java: corretto8
      python: 3.x
      ruby: "$MY_RUBY_VAR"
```

Anda dapat menentukan satu atau beberapa runtime di `runtime-versions` bagian file `buildspec` Anda. Jika runtime Anda bergantung pada runtime lain, Anda juga dapat menentukan runtime dependennya dalam file `buildspec`. Jika Anda tidak menentukan runtime apa pun dalam file `buildspec`, CodeBuild pilih runtime default yang tersedia di gambar yang Anda gunakan. Jika Anda menentukan satu atau beberapa runtime, hanya CodeBuild menggunakan runtime tersebut. Jika runtime dependen tidak ditentukan, CodeBuild coba pilih runtime dependen untuk Anda.

Jika dua runtime yang ditentukan bertentangan, build gagal. Misalnya, `android: 29` dan `java: openjdk11` konflik, jadi jika keduanya ditentukan, build gagal.

Untuk informasi selengkapnya tentang runtime yang tersedia, lihat [Runtime yang tersedia](#).

Note

Jika Anda menentukan `runtime-versions` bagian dan menggunakan gambar selain Ubuntu Standard Image 2.0 atau yang lebih baru, atau Amazon Linux 2 (AL2) image standar 1.0 atau yang lebih baru, build akan mengeluarkan peringatan, "Skipping install of runtimes. Runtime version selection is not supported by this build image."

fase/instal/perintah

Urutan opsional. Berisi urutan skalar, di mana setiap skalar mewakili satu perintah yang CodeBuild berjalan selama instalasi. CodeBuild menjalankan setiap perintah, satu per satu, dalam urutan yang tercantum, dari awal hingga akhir.

fase/pre_build

Urutan opsional. Merupakan perintah, jika ada, yang CodeBuild berjalan sebelum build. Misalnya, Anda dapat menggunakan fase ini untuk masuk ke Amazon ECR, atau Anda dapat menginstal dependensi npm.

fases/pre_build/ perintah

Urutan yang `pre_build` diperlukan jika ditentukan. Berisi urutan skalar, di mana setiap skalar mewakili satu perintah yang CodeBuild berjalan sebelum build. CodeBuild menjalankan setiap perintah, satu per satu, dalam urutan yang tercantum, dari awal hingga akhir.

fase/membangun

Urutan opsional. Merupakan perintah, jika ada, yang CodeBuild berjalan selama pembuatan. Misalnya, Anda mungkin menggunakan fase ini untuk menjalankan Mocha, RSpec, atau sbt.

fase/build/ perintah

Diperlukan `build` jika ditentukan. Berisi urutan skalar, di mana setiap skalar mewakili satu perintah yang CodeBuild berjalan selama pembuatan. CodeBuild menjalankan setiap perintah, satu per satu, dalam urutan yang tercantum, dari awal hingga akhir.

fase/post_build

Urutan opsional. Merupakan perintah, jika ada, yang CodeBuild berjalan setelah build. Misalnya, Anda dapat menggunakan Maven untuk mengemas artefak build ke dalam file JAR atau WAR, atau Anda dapat mendorong image Docker ke Amazon ECR. Kemudian Anda dapat mengirim pemberitahuan build melalui Amazon SNS.

fases/post_build/ perintah

Diperlukan `post_build` jika ditentukan. Berisi urutan skalar, di mana setiap skalar mewakili satu perintah yang CodeBuild berjalan setelah build. CodeBuild menjalankan setiap perintah, satu per satu, dalam urutan yang tercantum, dari awal hingga akhir.

melaporkan

report-group-name-or-arn

Urutan opsional. Menentukan grup laporan yang dikirimkan laporan. Sebuah proyek dapat memiliki maksimal lima kelompok laporan. Tentukan ARN dari grup laporan yang ada, atau

nama grup laporan baru. Jika Anda menentukan nama, CodeBuild membuat grup laporan menggunakan nama proyek Anda dan nama yang Anda tentukan dalam format `<project-name>-<report-group-name>`. Nama grup laporan juga dapat diatur menggunakan variabel lingkungan dalam buildspec seperti. `$REPORT_GROUP_NAME` Untuk informasi selengkapnya, lihat [Laporkan penamaan grup](#).

`<report-group>laporan// file`

Urutan yang diperlukan. Merupakan lokasi yang berisi data mentah hasil tes yang dihasilkan oleh laporan. Berisi urutan skalar, dengan setiap skalar mewakili lokasi terpisah di mana CodeBuild dapat menemukan file pengujian, relatif terhadap lokasi build asli atau, jika disetel, `file.base-directory` Lokasi dapat mencakup yang berikut:

- Satu file (misalnya, `my-test-report-file.json`).
- Sebuah file tunggal dalam subdirektori (misalnya, `my-subdirectory/my-test-report-file.json` atau `my-parent-subdirectory/my-subdirectory/my-test-report-file.json`).
- `'**/*'` mewakili semua file secara rekursif.
- `my-subdirectory/*` mewakili semua file dalam subdirektori bernama `my-subdirectory`.
- `my-subdirectory/**/*` mewakili semua file rekursif mulai dari subdirektori bernama `my-subdirectory`.

`<report-group>laporan// format file`

Pemetaan opsional. Merupakan format file laporan. Jika tidak ditentukan, JUNITXML digunakan. Nilai ini tidak peka huruf besar/kecil. Kemungkinan nilainya adalah:

Laporan pengujian

CUCUMBERJSON

Mentimun JSON

JUNITXML

JUnit XML

NUNITXML

NUnit XML

NUNIT3XML

NUnit 3 XML

TESTNGXML

TestNG XML

VISUALSTUDIOTRX

Studio Visual TRX

Laporan cakupan kode

CLOVERXML

XML semanggi

COBERTURAXML

XML Cobertura

JACOCOXML

JaCoCo XML

SIMPLECOV

SimpleCov JSON

Note

CodeBuild [menerima laporan cakupan kode JSON yang dihasilkan oleh simplecov, bukan simplecov-json.](#)

```
<report-group>laporan//direktori dasar
```

Pemetaan opsional. Merupakan satu atau beberapa direktori tingkat atas, relatif terhadap lokasi build asli, yang CodeBuild digunakan untuk menentukan tempat menemukan file pengujian mentah.

```
<report-group>laporan// buangan-jalur
```

Opsional. Menentukan apakah direktori file laporan diratakan dalam output. Jika ini tidak ditentukan, atau berisino, file laporan adalah output dengan struktur direktori mereka utuh. Jika ini berisiyes, semua file pengujian ditempatkan di direktori output yang sama. Misalnya, jika jalur ke

hasil tes adalah `com/myapp/mytests/TestResult.xml`, menentukan `yes` akan menempatkan file ini di `/TestResult.xml`.

artefak

Urutan opsional. Merupakan informasi tentang di mana CodeBuild dapat menemukan output build dan bagaimana CodeBuild mempersiapkannya untuk diunggah ke bucket keluaran S3. Urutan ini tidak diperlukan jika, misalnya, Anda sedang membangun dan mendorong image Docker ke Amazon ECR, atau Anda menjalankan pengujian unit pada kode sumber Anda, tetapi tidak membangunnya.

Note

Metadata Amazon S3 memiliki CodeBuild header bernama yang berisi CodeBuild build `x-amz-meta-codebuild-buildarn` yang menerbitkan `buildArn` artefak ke Amazon S3. Ditambahkan untuk memungkinkan pelacakan sumber untuk notifikasi dan referensi dari mana artefak tersebut dihasilkan. `buildArn`

artefak/file

Urutan yang diperlukan. Merupakan lokasi yang berisi artefak keluaran build di lingkungan build. Berisi urutan skalar, dengan setiap skalar mewakili lokasi terpisah di mana CodeBuild dapat menemukan artefak keluaran build, relatif terhadap lokasi build asli atau, jika disetel, direktori dasar. Lokasi dapat mencakup yang berikut:

- Satu file (misalnya, `my-file.jar`).
- Sebuah file tunggal dalam subdirektori (misalnya, `my-subdirectory/my-file.jar` atau `my-parent-subdirectory/my-subdirectory/my-file.jar`).
- `**/*` mewakili semua file secara rekursif.
- `my-subdirectory/*` mewakili semua file dalam subdirektori bernama `my-subdirectory`.
- `my-subdirectory/**/*` mewakili semua file rekursif mulai dari subdirektori bernama `my-subdirectory`.

Bila Anda menentukan lokasi artefak keluaran build, CodeBuild dapat menemukan lokasi build asli di lingkungan build. Anda tidak perlu menambahkan lokasi keluaran artefak build Anda dengan jalur ke lokasi build asli atau menentukan `./` atau serupa. Jika Anda ingin mengetahui jalur

ke lokasi ini, Anda dapat menjalankan perintah seperti `echo $CODEBUILD_SRC_DIR` selama pembuatan. Lokasi untuk setiap lingkungan bangunan mungkin sedikit berbeda.

artefak/nama

Nama opsional. Menentukan nama untuk artefak build Anda. Nama ini digunakan ketika salah satu dari berikut ini benar.

- Anda menggunakan CodeBuild API untuk membuat build dan `overrideArtifactName` flag disetel pada `ProjectArtifacts` objek saat proyek diperbarui, proyek dibuat, atau build dimulai.
- Anda menggunakan CodeBuild konsol untuk membuat build, nama ditentukan dalam file `buildspec`, dan Anda memilih Aktifkan versi semantik saat membuat atau memperbarui proyek. Untuk informasi selengkapnya, lihat [Buat proyek build \(konsol\)](#).

Anda dapat menentukan nama dalam file `buildspec` yang dihitung pada waktu pembuatan. Nama yang ditentukan dalam file `buildspec` menggunakan bahasa perintah Shell. Misalnya, Anda dapat menambahkan tanggal dan waktu untuk nama artefak Anda sehingga selalu unik. Nama artefak unik mencegah artefak ditimpa. Untuk informasi selengkapnya, lihat [Bahasa perintah Shell](#).

- Ini adalah contoh nama artefak yang ditambahkan dengan tanggal artefak dibuat.

```
version: 0.2
phases:
  build:
    commands:
      - rspec HelloWorld_spec.rb
artifacts:
  files:
    - '**/*'
  name: myname-$(date +%Y-%m-%d)
```

- Ini adalah contoh nama artefak yang menggunakan variabel CodeBuild lingkungan. Untuk informasi selengkapnya, lihat [Variabel lingkungan di lingkungan build](#).

```
version: 0.2
phases:
  build:
    commands:
      - rspec HelloWorld_spec.rb
artifacts:
  files:
    - '**/*'
```

```
name: myname- $\$$ AWS_REGION
```

- Ini adalah contoh nama artefak yang menggunakan variabel CodeBuild lingkungan dengan tanggal pembuatan artefak ditambahkan padanya.

```
version: 0.2
phases:
  build:
    commands:
      - rspec HelloWorld_spec.rb
artifacts:
  files:
    - '**/*'
  name:  $\$$ AWS_REGION- $\$($ date +%Y-%m-%d)
```

Anda dapat menambahkan informasi jalur ke nama sehingga artefak bernama ditempatkan di direktori berdasarkan jalur dalam nama. Dalam contoh ini, artefak build ditempatkan di output di bawahbuilds/<build number>/my-artifacts.

```
version: 0.2
phases:
  build:
    commands:
      - rspec HelloWorld_spec.rb
artifacts:
  files:
    - '**/*'
  name: builds/ $\$$ CODEBUILD_BUILD_NUMBER/my-artifacts
```

artefak/jalur buang

Opsional. Menentukan apakah direktori artefak build diratakan dalam output. Jika ini tidak ditentukan, atau berisino, artefak build adalah keluaran dengan struktur direktorinya utuh. Jika ini berisiyes, semua artefak build ditempatkan di direktori keluaran yang sama. Misalnya, jika jalur ke file dalam artefak keluaran build adalahcom/mycompany/app/HelloWorld.java, menentukan yes akan menempatkan file ini. /HelloWorld.java

artefak/direktori dasar

Pemetaan opsional. Merupakan satu atau beberapa direktori tingkat atas, relatif terhadap lokasi build asli, yang CodeBuild digunakan untuk menentukan file dan subdirektori mana yang akan disertakan dalam artefak keluaran build. Nilai yang valid meliputi:

- Sebuah direktori tingkat atas tunggal (misalnya, `my-directory`).
- `'my-directory*'` mewakili semua direktori tingkat atas dengan nama yang dimulai dengan `my-directory`

Direktori tingkat atas yang cocok tidak disertakan dalam artefak keluaran build, hanya file dan subdirektori.

Anda dapat menggunakan `files` dan `discard-paths` untuk lebih membatasi file dan subdirektori mana yang disertakan. Misalnya, untuk struktur direktori berikut:

```
.
### my-build-1
#   ### my-file-1.txt
### my-build-2
    ### my-file-2.txt
    ### my-subdirectory
        ### my-file-3.txt
```

Dan untuk artifacts urutan berikut:

```
artifacts:
  files:
    - '*/my-file-3.txt'
  base-directory: my-build-2
```

Subdirektori dan file berikut akan disertakan dalam artefak keluaran build:

```
.
### my-subdirectory
    ### my-file-3.txt
```

Sedangkan untuk artifacts urutan berikut:

```
artifacts:
  files:
    - '**/*'
  base-directory: 'my-build*'
  discard-paths: yes
```

File-file berikut akan disertakan dalam artefak keluaran build:

```
.  
### my-file-1.txt  
### my-file-2.txt  
### my-file-3.txt
```

artefak/tidak termasuk jalur

Pemetaan opsional. Merupakan satu atau lebih jalur, relatif terhadap `base-directory`, yang CodeBuild akan dikecualikan dari artefak build. Karakter tanda bintang (*) cocok dengan karakter nol atau karakter lain dari komponen nama tanpa melintasi batas folder. Tanda bintang ganda (**) cocok dengan nol atau lebih karakter dari komponen nama di semua direktori.

Contoh jalur pengecualian meliputi yang berikut:

- Untuk mengecualikan file dari semua direktori: "**/*file-name*/**/*"
- Untuk mengecualikan semua folder titik: "**/.*/**/*"
- Untuk mengecualikan semua file titik: "**/*.*

artefak/aktifkan-symlink

Opsional. Jika jenis output ZIP, menentukan apakah link simbolik internal dipertahankan dalam file ZIP. Jika ini berisi `yes`, semua tautan simbolis internal di sumber akan disimpan dalam artefak file ZIP.

artefak/awalan s3

Opsional. Menentukan awalan yang digunakan saat artefak dikeluarkan ke bucket Amazon S3 dan jenis namespace adalah. `BUILD_ID` Saat digunakan, jalur keluaran di ember adalah `<s3-prefix>/<build-id>/<name>.zip`.

artefak/artefak sekunder

Urutan opsional. Merupakan satu atau lebih definisi artefak sebagai pemetaan antara pengidentifikasi artefak dan definisi artefak. Setiap pengidentifikasi artefak di blok ini harus cocok dengan artefak yang ditentukan dalam `secondaryArtifacts` atribut proyek Anda. Setiap definisi terpisah memiliki sintaks yang sama dengan `artifacts` blok di atas.

Note

[artifacts/files](#) Urutan selalu diperlukan, bahkan ketika hanya ada artefak sekunder yang ditentukan.

Misalnya, jika proyek Anda memiliki struktur berikut:

```
{
  "name": "sample-project",
  "secondaryArtifacts": [
    {
      "type": "S3",
      "location": "<output-bucket1>",
      "artifactIdentifier": "artifact1",
      "name": "secondary-artifact-name-1"
    },
    {
      "type": "S3",
      "location": "<output-bucket2>",
      "artifactIdentifier": "artifact2",
      "name": "secondary-artifact-name-2"
    }
  ]
}
```

Kemudian buildspec Anda terlihat seperti berikut:

```
version: 0.2

phases:
build:
  commands:
    - echo Building...
artifacts:
  files:
    - '**/*'
  secondary-artifacts:
    artifact1:
      files:
        - directory/file1
      name: secondary-artifact-name-1
    artifact2:
      files:
        - directory/file2
      name: secondary-artifact-name-2
```

cache

Urutan opsional. Merupakan informasi tentang di mana CodeBuild dapat menyiapkan file untuk mengunggah cache ke bucket cache S3. Urutan ini tidak diperlukan jika jenis cache proyekNo Cache.

cache/jalur

Urutan yang diperlukan. Merupakan lokasi cache. Berisi urutan skalar, dengan setiap skalar mewakili lokasi terpisah di mana CodeBuild dapat menemukan artefak keluaran build, relatif terhadap lokasi build asli atau, jika disetel, direktori dasar. Lokasi dapat mencakup yang berikut:

- Satu file (misalnya, `my-file.jar`).
- Sebuah file tunggal dalam subdirektori (misalnya, `my-subdirectory/my-file.jar` atau `my-parent-subdirectory/my-subdirectory/my-file.jar`).
- `**/*` mewakili semua file secara rekursif.
- `my-subdirectory/*` mewakili semua file dalam subdirektori bernama `my-subdirectory`.
- `my-subdirectory/**/*` mewakili semua file rekursif mulai dari subdirektori bernama `my-subdirectory`.

Important

Karena deklarasi buildspec harus YAMM yang valid, spasi dalam deklarasi buildspec menjadi penting. Jika jumlah spasi dalam deklarasi buildspec Anda tidak valid, build mungkin akan segera gagal. Anda dapat menggunakan validator YAMAL untuk menguji apakah deklarasi buildspec Anda adalah YAMAL yang valid.

Jika Anda menggunakan AWS CLI, atau AWS SDK untuk mendeklarasikan buildspec saat Anda membuat atau memperbarui proyek build, buildspec harus berupa string tunggal yang diekspresikan dalam format YAMAL, bersama dengan spasi putih yang diperlukan dan karakter escape baris baru. Ada contoh di bagian selanjutnya.

Jika Anda menggunakan CodeBuild atau AWS CodePipeline konsol alih-alih file buildspec.yml, Anda dapat menyisipkan perintah untuk fase saja. `build` Alih-alih menggunakan sintaks sebelumnya, Anda mencantumkan, dalam satu baris, semua perintah yang ingin Anda jalankan selama fase build. Untuk beberapa perintah, pisahkan setiap perintah dengan `&&` (misalnya, `mvn test && mvn package`).

Anda dapat menggunakan CodeBuild atau CodePipeline konsol alih-alih file buildspec.yml untuk menentukan lokasi artefak keluaran build di lingkungan build. Alih-alih

menggunakan sintaks sebelumnya, Anda mencantumkan, dalam satu baris, semua lokasi. Untuk beberapa lokasi, pisahkan setiap lokasi dengan koma (misalnya, `buildspec.yml`, `target/my-app.jar`).

Contoh Buildspec

Berikut adalah contoh file `buildspec.yml`.

```
version: 0.2

env:
  variables:
    JAVA_HOME: "/usr/lib/jvm/java-8-openjdk-amd64"
  parameter-store:
    LOGIN_PASSWORD: /CodeBuild/dockerLoginPassword

phases:
  install:
    commands:
      - echo Entered the install phase...
      - apt-get update -y
      - apt-get install -y maven
    finally:
      - echo This always runs even if the update or install command fails
  pre_build:
    commands:
      - echo Entered the pre_build phase...
      - docker login -u User -p $LOGIN_PASSWORD
    finally:
      - echo This always runs even if the login command fails
  build:
    commands:
      - echo Entered the build phase...
      - echo Build started on `date`
      - mvn install
    finally:
      - echo This always runs even if the install command fails
  post_build:
    commands:
      - echo Entered the post_build phase...
      - echo Build completed on `date`
```

```

reports:
  arn:aws:codebuild:your-region:your-aws-account-id:report-group/report-group-name-1:
    files:
      - "**/*"
    base-directory: 'target/tests/reports'
    discard-paths: no
  reportGroupCucumberJson:
    files:
      - 'cucumber/target/cucumber-tests.xml'
    discard-paths: yes
    file-format: CUCUMBERJSON # default is JUNITXML
artifacts:
  files:
    - target/messageUtil-1.0.jar
  discard-paths: yes
  secondary-artifacts:
    artifact1:
      files:
        - target/artifact-1.0.jar
      discard-paths: yes
    artifact2:
      files:
        - target/artifact-2.0.jar
      discard-paths: yes
cache:
  paths:
    - '/root/.m2/**/*'

```

Berikut adalah contoh buildspec sebelumnya, dinyatakan sebagai string tunggal, untuk digunakan dengan, atau SDK AWS CLI. AWS

```

"version: 0.2\n\nenv:\n  variables:\n    JAVA_HOME: \"/usr/lib/jvm/java-8-openjdk-
amd64\""\n  parameter-store:\n    LOGIN_PASSWORD: /CodeBuild/dockerLoginPassword\n
phases:\n\n  install:\n    commands:\n      - echo Entered the install phase...\n
- apt-get update -y\n      - apt-get install -y maven\n    finally:\n      - echo This
always runs even if the update or install command fails\n  pre_build:\n    commands:
\n      - echo Entered the pre_build phase...\n      - docker login -u User -p
$LOGIN_PASSWORD\n    finally:\n      - echo This always runs even if the login command
fails\n  build:\n    commands:\n      - echo Entered the build phase...\n      - echo
Build started on `date`\n      - mvn install\n    finally:\n      - echo This always
runs even if the install command fails\n  post_build:\n    commands:\n      - echo
Entered the post_build phase...\n      - echo Build completed on `date`\n\n reports:

```



```
\n reportGroupJUnitXml:\n   files:\n     - \"**/*\" \n   base-directory: 'target/
tests/reports'\n   discard-paths: false\n reportGroupCucumberJson:\n   files:\n
- 'cucumber/target/cucumber-tests.xml'\n   file-format: CUCUMBERJSON\n\nartifacts:\n
files:\n   - target/messageUtil-1.0.jar\n   discard-paths: yes\n   secondary-artifacts:
\n   artifact1:\n     files:\n       - target/messageUtil-1.0.jar\n     discard-
paths: yes\n   artifact2:\n     files:\n       - target/messageUtil-1.0.jar\n
discard-paths: yes\n   cache:\n     paths:\n       - '/root/.m2/**/*'"
```

Berikut adalah contoh perintah dalam build fase, untuk digunakan dengan CodeBuild atau CodePipeline konsol.

```
echo Build started on `date` && mvn install
```

Dalam contoh ini:

- Variabel lingkungan khusus, dalam teks biasa, dengan kunci JAVA_HOME dan nilai/usr/lib/jvm/java-8-openjdk-amd64, diatur.
- Variabel lingkungan kustom bernama dockerLoginPassword Anda disimpan di Amazon EC2 Systems Manager Parameter Store direferensikan nanti dalam perintah build dengan menggunakan kunci. LOGIN_PASSWORD
- Anda tidak dapat mengubah nama fase build ini. Perintah yang dijalankan dalam contoh ini adalah apt-get update -y dan apt-get install -y maven (untuk menginstal Apache Maven), mvn install (untuk mengkompilasi, menguji, dan mengemas kode sumber ke dalam artefak keluaran build dan menginstal artefak keluaran build di repositori internalnya), docker login (untuk masuk ke Docker dengan kata sandi yang sesuai dengan nilai variabel lingkungan khusus yang Anda tetapkan di Amazon dockerLoginPassword EC2 Systems Manager Parameter Store), dan beberapa perintah. echo echoPerintah disertakan di sini untuk menunjukkan bagaimana CodeBuild menjalankan perintah dan urutan menjalankannya.
- filesmewakili file yang akan diunggah ke lokasi keluaran build. Dalam contoh ini, CodeBuild upload file messageUtil-1.0.jar tunggal. messageUtil-1.0.jarFile dapat ditemukan di direktori relatif bernama target di lingkungan build. Karena discard-paths: yes ditentukan, messageUtil-1.0.jar diunggah langsung (dan bukan ke target direktori perantara). Nama file messageUtil-1.0.jar dan nama direktori relatif target didasarkan pada cara Apache Maven membuat dan menyimpan artefak keluaran build untuk contoh ini saja. Dalam skenario Anda sendiri, nama file dan direktori ini akan berbeda.
- reportsmewakili dua grup laporan yang menghasilkan laporan selama pembuatan:

- `arn:aws:codebuild:your-region:your-aws-account-id:report-group/report-group-name-1` menentukan ARN dari grup laporan. Hasil pengujian yang dihasilkan oleh kerangka pengujian ada di `target/tests/reports` direktori. Format file adalah `JunitXml` dan jalur tidak dihapus dari file yang berisi hasil pengujian.
- `reportGroupCucumberJson` menentukan grup laporan baru. Jika nama proyek adalah `my-project`, grup laporan dengan nama akan `my-project-reportGroupCucumberJson` dibuat saat build dijalankan. Hasil pengujian yang dihasilkan oleh kerangka pengujian ada di `cucumber/target/cucumber-tests.xml`. Format file pengujian adalah `CucumberJson` dan jalur dihapus dari file yang berisi hasil pengujian.

Versi Buildspec

Tabel berikut mencantumkan versi buildspec dan perubahan antar versi.

Versi	Perubahan
0.2	<ul style="list-style-type: none"> • <code>environment_variables</code> telah diubah namanya menjadi <code>env</code> • <code>plaintext</code> telah diubah namanya menjadi <code>variables</code> • <code>typeProperti</code> untuk <code>artifacts</code> telah usang. • Di versi 0.1, AWS CodeBuild jalankan setiap perintah build dalam instance terpisah dari shell default di lingkungan build. Di versi 0.2, CodeBuild jalankan semua perintah build dalam instance yang sama dari shell default di lingkungan build.
0.1	Ini adalah definisi awal dari format spesifikasi build.

Batch membangun referensi buildspec

Topik ini berisi referensi buildspec untuk properti batch build.

batch

Pemetaan opsional. Pengaturan batch build untuk proyek.

batch/cepat gagal

Tidak wajib. Menentukan perilaku membangun batch ketika satu atau lebih tugas membangun gagal.

false

Nilai default. Semua membangun berjalan akan selesai.

true

Semua build yang berjalan akan dihentikan ketika salah satu tugas build gagal.

Secara default, semua tugas build batch berjalan dengan pengaturan build seperti `env` dan `phases`, ditentukan dalam file `buildspec`. Anda dapat mengganti pengaturan build default dengan menentukan berbeda `env` nilai atau file `buildspec` yang berbeda dalam `batch/<batch-type>/buildspecparameter`.

Isi dari `batch` properti bervariasi berdasarkan jenis batch membangun yang ditentukan. Jenis build batch yang mungkin adalah:

- [batch/build-graph](#)
- [batch/build-list](#)
- [batch/build-matrix](#)

batch/build-graph

Menentukan grafik build. Sebuah grafik membangun mendefinisikan satu set tugas yang memiliki dependensi pada tugas-tugas lain dalam batch. Untuk informasi selengkapnya, lihat [Membangun grafik](#).

Elemen ini berisi array tugas build. Setiap tugas build berisi properti berikut.

pengenal

Diperlukan. Pengidentifikasi tugas.

buildspec

Tidak wajib. Nama jalur dan file dari file buildspec yang akan digunakan untuk tugas ini. Jika parameter ini tidak ditentukan, file buildspec saat ini digunakan.

debug-sesi

Tidak wajib. Nilai Boolean yang menunjukkan apakah sesi debugging diaktifkan untuk membangun batch ini. Untuk informasi lebih lanjut tentang debugging sesi, lihat [Melihat build yang sedang berjalan di Session Manager](#).

false

Sesi debugging dinonaktifkan.

true

Sesi debugging diaktifkan.

tergantung pada

Tidak wajib. Array pengidentifikasi tugas bahwa tugas ini tergantung pada. Tugas ini tidak akan berjalan sampai tugas-tugas ini selesai.

env

Tidak wajib. Membangun lingkungan menimpa untuk tugas. Ini dapat berisi properti berikut:

komputasi tipe

Identifier dari jenis komputasi yang akan digunakan untuk tugas. Lihat [computeType](#) di [the section called “Bangun mode dan tipe komputasi lingkungan”](#) untuk nilai yang mungkin.

gambar

Pengenal gambar yang akan digunakan untuk tugas. Lihat [Pengidentifikasi gambar](#) di [the section called “Gambar Docker disediakan oleh CodeBuild”](#) untuk nilai yang mungkin.

mode istimewa

Nilai Boolean yang menunjukkan apakah akan menjalankan daemon Docker di dalam wadah Docker. Mengatur `true` hanya jika proyek membangun digunakan untuk membangun gambar Docker. Jika tidak, build yang mencoba berinteraksi dengan daemon Docker akan gagal. Pengaturan default-nya adalah `false`.

tipe

Pengenalan jenis lingkungan yang akan digunakan untuk tugas. Lihat [Tipe lingkungan](#) di [the section called “Bangun mode dan tipe komputasi lingkungan”](#) untuk nilai yang mungkin.

variabel

Variabel lingkungan yang akan hadir dalam membangun lingkungan. Lihat [env/variabel](#) untuk informasi selengkapnya.

abaikan kegagalan

Tidak wajib. Nilai Boolean yang menunjukkan jika kegagalan tugas membangun ini dapat diabaikan.

false

Nilai default. Jika tugas build ini gagal, build batch akan gagal.

true

Jika tugas build ini gagal, batch build masih bisa berhasil.

Berikut ini adalah contoh entri buildspec:

```
batch:
  fast-fail: false
  build-graph:
    - identifier: build1
      env:
        variables:
          BUILD_ID: build1
      ignore-failure: false
    - identifier: build2
      buildspec: build2.yml
      env:
        variables:
          BUILD_ID: build2
      depend-on:
        - build1
    - identifier: build3
      env:
        variables:
          BUILD_ID: build3
      depend-on:
```

```
- build2
```

batch/build-list

Menentukan daftar build. Daftar build digunakan untuk menentukan sejumlah tugas yang berjalan secara paralel. Untuk informasi selengkapnya, lihat [Membangun daftar](#).

Elemen ini berisi array tugas build. Setiap tugas build berisi properti berikut.

pengenal

Diperlukan. Pengidentifikasi tugas.

buildspec

Tidak wajib. Nama jalur dan file dari file buildspec yang akan digunakan untuk tugas ini. Jika parameter ini tidak ditentukan, file buildspec saat ini digunakan.

debug-sesi

Tidak wajib. Nilai Boolean yang menunjukkan apakah sesi debugging diaktifkan untuk membangun batch ini. Untuk informasi lebih lanjut tentang debugging sesi, lihat [Melihat build yang sedang berjalan di Session Manager](#).

false

Sesi debugging dinonaktifkan.

true

Sesi debugging diaktifkan.

env

Tidak wajib. Membangun lingkungan menimpa untuk tugas. Ini dapat berisi properti berikut:

komputasi tipe

Identifier dari jenis komputasi yang akan digunakan untuk tugas. Lihat [computeType](#) di [the section called “Bangun mode dan tipe komputasi lingkungan”](#) untuk nilai yang mungkin.

gambar

Pengenal gambar yang akan digunakan untuk tugas. Lihat [Pengidentifikasi gambar](#) di [the section called “Gambar Docker disediakan oleh CodeBuild”](#) untuk nilai yang mungkin.

mode istimewa

Nilai Boolean yang menunjukkan apakah akan menjalankan daemon Docker di dalam wadah Docker. Mengatur `true` hanya jika proyek membangun digunakan untuk membangun gambar Docker. Jika tidak, build yang mencoba berinteraksi dengan daemon Docker akan gagal. Pengaturan default-nya adalah `false`.

tipe

Pengenalan jenis lingkungan yang akan digunakan untuk tugas. Lihat [Tipe lingkungan](#) di [the section called “Bangun mode dan tipe komputasi lingkungan”](#) untuk nilai yang mungkin.

variabel

Variabel lingkungan yang akan hadir dalam membangun lingkungan. Lihat [env/variabel](#) untuk informasi selengkapnya.

abaikan kegagalan

Tidak wajib. Nilai Boolean yang menunjukkan jika kegagalan tugas membangun ini dapat diabaikan.

false

Nilai default. Jika tugas build ini gagal, build batch akan gagal.

true

Jika tugas build ini gagal, batch build masih bisa berhasil.

Berikut ini adalah contoh entri buildspec:

```
batch:
  fast-fail: false
  build-list:
    - identifier: build1
      env:
        variables:
          BUILD_ID: build1
      ignore-failure: false
    - identifier: build2
      buildspec: build2.yml
      env:
        variables:
```

```
BUILD_ID: build2
ignore-failure: true
```

batch/build-matrix

Menentukan matriks build. Sebuah membangun matriks mendefinisikan tugas dengan konfigurasi yang berbeda yang berjalan secara paralel. CodeBuild membuat build terpisah untuk setiap kombinasi konfigurasi yang mungkin. Untuk informasi selengkapnya, lihat [Membangun matriks](#).

statis

Properti statis berlaku untuk semua tugas membangun.

abaikan kegagalan

Tidak wajib. Nilai Boolean yang menunjukkan jika kegagalan tugas membangun ini dapat diabaikan.

false

Nilai default. Jika tugas build ini gagal, build batch akan gagal.

true

Jika tugas build ini gagal, batch build masih bisa berhasil.

env

Tidak wajib. Membangun lingkungan menimpa untuk semua tugas.

mode istimewa

Nilai Boolean yang menunjukkan apakah akan menjalankan daemon Docker di dalam wadah Docker. Mengatur `true` hanya jika proyek membangun digunakan untuk membangun gambar Docker. Jika tidak, build yang mencoba berinteraksi dengan daemon Docker akan gagal. Pengaturan default-nya adalah `false`.

tipe

Pengenalan jenis lingkungan yang akan digunakan untuk tugas. Lihat [Tipe lingkungan di the section called “Bangun mode dan tipe komputasi lingkungan”](#) untuk nilai yang mungkin.

dinamis

Properti dinamis mendefinisikan membangun matriks.

buildspec

Tidak wajib. Array yang berisi nama path dan file dari file buildspec untuk digunakan untuk tugas-tugas ini. Jika parameter ini tidak ditentukan, file buildspec saat ini digunakan.

env

Tidak wajib. Lingkungan build menimpa untuk tugas-tugas ini.

komputasi tipe

Array yang berisi pengidentifikasi jenis komputasi yang akan digunakan untuk tugas-tugas ini. Lihat `computeType` [the section called “Bangun mode dan tipe komputasi lingkungan”](#) untuk nilai yang mungkin.

gambar

Array yang berisi pengidentifikasi gambar yang akan digunakan untuk tugas-tugas ini. Lihat `imageIdentifier` [the section called “Gambar Docker disediakan oleh CodeBuild”](#) untuk nilai yang mungkin.

variabel

Array yang berisi variabel lingkungan yang akan hadir dalam membangun lingkungan untuk tugas-tugas ini. Lihat [env/variabel](#) untuk informasi selengkapnya.

Berikut ini adalah contoh entri build matrix:

```
batch:
  build-matrix:
    static:
      ignore-failure: false
    dynamic:
      buildspec:
        - matrix1.yml
        - matrix2.yml
      env:
        variables:
          MY_VAR:
            - VALUE1
            - VALUE2
            - VALUE3
```

Untuk informasi selengkapnya, lihat [Membangun matriks](#).

GitHubReferensi buildspec pelari aksi

Topik ini berisi referensi buildspec untuk properti runner GitHub Action.

langkah

Urutan opsional. Langkah-langkah yang digunakan untuk menjalankan perintah dan tindakan diCodeBuild. Untuk informasi selengkapnya, lihat [GitHubPelari aksi diAWS CodeBuild](#).

Note

Setiap fase build mendukung daftar `commands` atau daftar `steps`, tetapi keduanya tidak dapat digunakan dalam fase yang sama.

Setiap langkah build berisi properti berikut.

id

Tidak wajib. Identifier untuk langkah yang dapat digunakan untuk referensi langkah dalam [konteks](#) lain.

jika

Tidak wajib. Pernyataan kondisional yang dapat digunakan untuk mencegah langkah berjalan kecuali kondisi terpenuhi. [Pernyataan ini dapat menggunakan konteks yang didukung, seperti referensi variabel lingkungan dariCodeBuild, serta ekspresi.](#)

nama

Tidak wajib. Nama untuk langkahnya. Jika nama tidak ditentukan, nama akan default ke teks yang ditentukan dalam `run` perintah.

menggunakan

Tindakan yang berjalan untuk langkah. Beberapa tindakan mengharuskan Anda untuk mengatur input menggunakan `with`. Referensi README tindakan untuk menentukan input mana yang diperlukan. Untuk informasi selengkapnya, lihat [Yang manaGitHubTindakan yang bisa saya gunakan?](#)

Jika `uses` ditentukan dalam fase build Anda, itu tidak dapat digunakan dengan `run`.

Note

Disarankan agar Anda menyertakan versi tindakan yang Anda gunakan. Ini dapat dilakukan dengan menentukan tag Git ref, SHA, atau Docker. Untuk informasi selengkapnya, lihat [sintaks steps.uses](#).

lari

Perintah yang menjalankan program baris perintah. Ini bisa berupa perintah baris tunggal atau perintah multi-baris. Secara default, perintah ini dijalankan menggunakan shell non-login. Untuk memilih shell yang berbeda, gunakan `shell`.

Jika `run` ditentukan dalam fase build Anda, itu tidak dapat digunakan dengan `uses`.

cangkang

Tidak wajib. Shell ditentukan untuk urutan ini. Untuk parameter shell yang didukung, lihat [steps.shell](#). Jika tidak ditentukan, shell yang digunakan adalah `bash`. Jika `bash` tidak tersedia, shell yang digunakan.

dengan

Tidak wajib. Sebuah peta parameter masukan didefinisikan oleh tindakan. Setiap parameter adalah pasangan kunci/nilai.

dengan.args

Tidak wajib. String yang mendefinisikan masukan untuk container Docker.

dengan.entrypoint

Tidak wajib. Entrypoint Docker ditentukan untuk Dockerfile.

env

Tidak wajib. Variabel yang ditentukan untuk langkah-langkah untuk digunakan dalam lingkungan.

continue-on-error

Tidak wajib. Sebuah nilai Boolean yang menunjukkan jika kegagalan urutan langkah ini dapat diabaikan.

false

Nilai default. Jika urutan langkah ini gagal, build akan gagal.

`true`

Jika urutan langkah ini gagal, build masih bisa berhasil.

`timeout-menit`

Tidak wajib. Jumlah maksimum menit langkah dapat dijalankan sebelum mengakhiri. Secara default, tidak ada batas waktu. Jika batas waktu langkah melebihi batas waktu build, langkah akan berhenti saat batas waktu build tercapai.

Berikut ini adalah contoh menggunakan [Super-Linter](#) GitHub Action:

```
version: 0.2
phases:
  build:
    steps:
      - name: Lint Code Base
        uses: github/super-linter@v5
        env:
          VALIDATE_ALL_CODEBASE: true
          USE_FIND_ALGORITHM: true
          FILTER_REGEX_INCLUDE: '/github/workspace/buildspec.yml'
```

Membangun referensi lingkungan untuk AWS CodeBuild

Saat Anda menelepon AWS CodeBuild untuk menjalankan build, Anda harus memberikan informasi tentang lingkungan build. Lingkungan build mewakili kombinasi sistem operasi, runtime bahasa pemrograman, dan alat yang CodeBuild digunakan untuk menjalankan build. Untuk informasi tentang cara kerja lingkungan penerapan, lihat [Cara kerja CodeBuild](#).

Lingkungan penerapan gambar Docker. Untuk informasi, lihat [glosarium Docker di situs Docker](#)
[Docker Docker](#) Docker Docker Docker Docker Docker

Bila Anda memberikan informasi CodeBuild tentang lingkungan build, Anda menentukan pengenalan image Docker dalam jenis repositori yang didukung. Ini termasuk CodeBuild Docker gambar Docker, dan Docker Container Elastic Container Registry (Amazon ECR) yang memiliki izin untuk diakses oleh AWS akun Anda.

- Kami menyarankan Anda menggunakan gambar Docker yang disimpan di repositori gambar CodeBuild Docker, karena mereka dioptimalkan untuk digunakan dengan layanan. Untuk informasi selengkapnya, lihat [Gambar Docker disediakan oleh CodeBuild](#).

- Untuk mendapatkan pengenalan image Docker yang tersedia untuk umum yang disimpan di Docker Hub, lihat [Mencari Repositori](#) di situs web Docker Docs.
- Untuk mempelajari cara bekerja dengan image Docker yang disimpan di repositori Amazon ECR di AWS akun Anda, lihat [Sampel Amazon ECR](#).

Selain pengenalan gambar Docker, Anda juga menentukan sekumpulan sumber daya komputasi yang digunakan lingkungan build. Untuk informasi selengkapnya, lihat [Bangun mode dan tipe komputasi lingkungan](#).

Topik

- [Gambar Docker disediakan oleh CodeBuild](#)
- [Bangun mode dan tipe komputasi lingkungan](#)
- [Kerangka dan perintah dalam membangun lingkungan](#)
- [Variabel lingkungan di lingkungan build](#)
- [Tugas latar belakang di lingkungan build](#)

Gambar Docker disediakan oleh CodeBuild

Gambar yang didukung adalah versi utama terbaru dari gambar yang tersedia CodeBuild dan diperbarui dengan pembaruan versi minor dan patch. CodeBuild mengoptimalkan durasi penyediaan build dengan gambar yang didukung dengan menyimpannya di Amazon Machine [Images \(AMI\) mesin](#). Jika Anda ingin mendapatkan keuntungan dari caching dan meminimalkan durasi penyediaan build, pilih Selalu gunakan gambar terbaru untuk versi runtime ini di bagian Versi gambar CodeBuild konsol, bukan versi yang lebih terperinci, seperti `aws/codebuild/amazonlinux2-x86_64-standard:4.0-1.0.0`

CodeBuild sering memperbarui daftar gambar Docker untuk menambahkan gambar terbaru dan menghentikan gambar lama. Untuk mendapatkan daftar terbaru, lakukan salah satu hal berikut:

- Di CodeBuild konsol, di halaman Create build project wizard atau Edit Build Project, untuk gambar Lingkungan, pilih Managed image. Pilih dari daftar drop-down versi Sistem Operasi, Runtime, dan Runtime. Untuk informasi selengkapnya, lihat [Buat proyek build \(konsol\)](#) atau [Mengubah setelan proyek build \(konsol\)](#).
- Untuk AWS CLI, jalankan `list-curated-environment-images` perintah:

```
aws codebuild list-curated-environment-images
```

- Untuk AWS SDK, hubungi `ListCuratedEnvironmentImages` operasi untuk bahasa pemrograman target Anda. Lihat informasi yang lebih lengkap di [AWSReferensi SDK dan alat](#).

Gambar dasar platform Windows Server Core 2019 hanya tersedia di wilayah berikut:

- AS Timur (N. Virginia)
- AS Timur (Ohio)
- AS Barat (Oregon)
- Eropa (Irlandia)

Gambar komputasi EC2

AWS CodeBuildmendukung gambar Docker berikut yang tersedia untuk komputasi EC2 di CodeBuild

Platform	Pengidentifikasi gambar	Definisi
Amazon Linux 2	<code>aws/codebuild/amazonlinux2-x86_64-standard:4.0</code>	al2/standar/4.0
Amazon Linux 2023	<code>aws/codebuild/amazonlinux2-x86_64-standard:5.0</code>	al2/standar/5.0
Amazon Linux 2	<code>aws/codebuild/amazonlinux2-x86_64-standard:corretto8</code>	al2/standar/corretto8
Amazon Linux 2	<code>aws/codebuild/amazonlinux2-x86_64-standard:corretto11</code>	al2/standar/corretto11

Platform	Pengidentifikasi gambar	Definisi
Amazon Linux 2	aws/codebuild/amazonlinux2-aarch64-standard:2.0	al2/aarch64/standar/2.0
Amazon Linux 2023	aws/codebuild/amazonlinux2-aarch64-standard:3.0	al2/aarch64/standar/3.0
Ubuntu 20.04	aws/codebuild/standard:5.0	ubuntu/standar/5.0
Ubuntu 22.04	aws/codebuild/standard:6.0	ubuntu/standar/6.0
Ubuntu 22.04	aws/codebuild/standard:7.0	ubuntu/standar/7.0
Inti Server Windows 2019	aws/codebuild/windows-base:2019-1.0	N/A
Inti Server Windows 2019	aws/codebuild/windows-base:2019-2.0	N/A
Inti Server Windows 2019	aws/codebuild/windows-base:2019-3.0	N/A
Inti Server Windows 2022	aws/codebuild/windows-base:2022-1.0	N/A

Lambda menghitung gambar

AWS CodeBuild mendukung gambar Docker berikut yang tersedia untuk AWS Lambda dihitung. CodeBuild

Arsitektur **aarch64**

Platform	Pengidentifikasi gambar	Definisi
Amazon Linux 2	aws/codebuild/amazonlinux-aarch64-lambda-standard:dotnet6	al-lambda/aarch64/dotnet6
Amazon Linux 2	aws/codebuild/amazonlinux-aarch64-lambda-standard:go1.21	al-lambda/aarch64/go1.21
Amazon Linux 2	aws/codebuild/amazonlinux-aarch64-lambda-standard:corretto11	al-lambda/aarch64/corretto11
Amazon Linux 2	aws/codebuild/amazonlinux-aarch64-lambda-standard:corretto17	al-lambda/aarch64/corretto17
Amazon Linux 2023	aws/codebuild/amazonlinux-aarch64-lambda-standard:corretto21	al-lambda/aarch64/corretto21
Amazon Linux 2	aws/codebuild/amazonlinux-aarch64-lambda-standard:nodejs18	al-lambda/aarch64/nodejs18
Amazon Linux 2023	aws/codebuild/amazonlinux-aarch64-lambda-standard:nodejs20	al-lambda/aarch64/nodejs20

Platform	Pengidentifikasi gambar	Definisi
Amazon Linux 2	aws/codebuild/amazonlinux-aarch64-lambda-standard:python3.11	al-lambda/aarch64/python3.11
Amazon Linux 2023	aws/codebuild/amazonlinux-aarch64-lambda-standard:python3.12	al-lambda/aarch64/python3.12
Amazon Linux 2	aws/codebuild/amazonlinux-aarch64-lambda-standard:ruby3.2	al-lambda/aarch64/ruby3.2

Arsitektur **x86_64**

Platform	Pengidentifikasi gambar	Definisi
Amazon Linux 2	aws/codebuild/amazonlinux-x86_64-lambda-standard:dotnet6	al-lambda/x86_64/dotnet6
Amazon Linux 2	aws/codebuild/amazonlinux-x86_64-lambda-standard:go1.21	al-lambda/x86_64/go1.21
Amazon Linux 2	aws/codebuild/amazonlinux-x86_64-lambda-standard:corretto11	al-lambda/x86_64/corretto11
Amazon Linux 2	aws/codebuild/amazonlinux-x86_64-lambda-standard:corretto17	al-lambda/x86_64/corretto17

Platform	Pengidentifikasi gambar	Definisi
	<code>bda-standard:corretto17</code>	
Amazon Linux 2023	<code>aws/codebuild/amazonlinux-x86_64-lambda-standard:corretto21</code>	al-lambda/x86_64/corretto21
Amazon Linux 2	<code>aws/codebuild/amazonlinux-x86_64-lambda-standard:nodejs18</code>	al-lambda/x86_64/nodejs18
Amazon Linux 2023	<code>aws/codebuild/amazonlinux-x86_64-lambda-standard:nodejs20</code>	al-lambda/x86_64/nodejs20
Amazon Linux 2	<code>aws/codebuild/amazonlinux-x86_64-lambda-standard:python3.11</code>	al-lambda/x86_64/python3.11
Amazon Linux 2023	<code>aws/codebuild/amazonlinux-x86_64-lambda-standard:python3.12</code>	al-lambda/x86_64/python3.12
Amazon Linux 2	<code>aws/codebuild/amazonlinux-x86_64-lambda-standard:ruby3.2</code>	al-lambda/x86_64/ruby3.2

Gambar usang

Gambar usang adalah gambar yang tidak lagi di-cache atau diperbarui oleh CodeBuild. Gambar usang tidak lagi menerima pembaruan versi minor atau pembaruan versi tambalan, dan karena tidak lagi diperbarui, penggunaannya mungkin tidak aman. Jika CodeBuild project Anda dikonfigurasi untuk menggunakan versi image yang lebih lama, proses penyediaan akan mengunduh docker image ini dan menggunakannya untuk membuat lingkungan runtime kontainer, yang dapat meningkatkan durasi penyediaan dan durasi build secara keseluruhan.

CodeBuild telah mencela gambar Docker berikut. Anda masih dapat menggunakan gambar ini, tetapi gambar tersebut tidak akan di-cache di host build dan akan menghasilkan waktu penyediaan yang lebih tinggi.

Platform	Pengidentifikasi gambar	Definisi	Tanggal pengusangan
Amazon Linux 2	aws/codebuild/ amazonlinux2- x86_64-st andard:3.0	al2/standar/3.0	9 Mei 2023
Ubuntu 18.04	aws/codebuild/ standard:4.0	ubuntu/standar/4.0	31 Maret 2023
Amazon Linux 2	aws/codebuild/ amazonlinux2- aarch64-s tandard:1.0	al2/aarch64/standa r/1.0	31 Maret 2023
Ubuntu 18.04	aws/codebuild/ standard:3.0	ubuntu/standar/3.0	30 Juni 2022
Amazon Linux 2	aws/codebuild/ amazonlinux2- x86_64-st andard:2.0	al2/standar/2.0	30 Juni 2022

Topik

- [Runtime yang tersedia](#)
- [Versi waktu aktif](#)

Runtime yang tersedia

Anda dapat menentukan satu atau beberapa runtime di `runtime-versions` bagian file `buildspec` Anda. Jika runtime Anda bergantung pada runtime lain, Anda juga dapat menentukan runtime dependennya dalam file `buildspec`. Jika Anda tidak menentukan runtime apa pun dalam file `buildspec`, CodeBuild pilih runtime default yang tersedia di gambar yang Anda gunakan. Jika Anda menentukan satu atau beberapa runtime, hanya CodeBuild menggunakan runtime tersebut. Jika runtime dependen tidak ditentukan, cobalah untuk CodeBuild memilih runtime dependen untuk Anda. Untuk informasi selengkapnya, lihat [Specify runtime versions in the buildspec file](#).

Topik

- [Runtime gambar Linux](#)
- [Runtime gambar Windows](#)

Runtime gambar Linux

Tabel berikut berisi runtime yang tersedia dan gambar Linux standar yang mendukungnya.

Waktu proses platform Ubuntu dan Amazon Linux

Nama runtime	Versi	Citra
dotnet	3.1	Amazon Linux 2 AArch64 standar: 2.0 Standar Ubuntu: 5.0
	5.0	Standar Ubuntu: 5.0
	6.0	Amazon Linux 2 x86_64 Lambda standar: dotnet6 Amazon Linux 2 AArch64 Lambda standar: dotnet6

Nama runtime	Versi	Citra
		Amazon Linux 2 x86_64 standar:4.0 Amazon Linux 2023 x86_64 standar:5.0 Amazon Linux 2023 AArch64 standar: 3.0 Standar Ubuntu: 6.0 Standar Ubuntu: 7.0
	8.0	Amazon Linux 2023 x86_64 standar:5.0 Amazon Linux 2023 AArch64 standar: 3.0 Standar Ubuntu: 7.0
golang	1.12	Amazon Linux 2 AArch64 standar: 2.0
	1.13	Amazon Linux 2 AArch64 standar: 2.0
	1.14	Amazon Linux 2 AArch64 standar: 2.0
	1.15	Standar Ubuntu: 5.0
	1.16	Standar Ubuntu: 5.0
	1.18	Amazon Linux 2 x86_64 standar:4.0 Standar Ubuntu: 6.0

Nama runtime	Versi	Citra
	1.20	Amazon Linux 2023 x86_64 standar:5.0 Amazon Linux 2023 AArch64 standar: 3.0 Standar Ubuntu: 7.0
	1.21	Amazon Linux 2 x86_64 Lambda standar: go1.21 Amazon Linux 2 AArch64 Lambda standar: go1.21 Amazon Linux 2023 x86_64 standar:5.0 Amazon Linux 2023 AArch64 standar: 3.0 Standar Ubuntu: 7.0
jawa	corretto8	Amazon Linux 2 x86_64 standar:corretto8 Amazon Linux 2 AArch64 standar: 2.0 Standar Ubuntu: 5.0

Nama runtime	Versi	Citra
	corretto11	<p>Amazon Linux 2 x86_64 standar:corretto11</p> <p>Amazon Linux 2 x86_64 Lambda standar: corretto11</p> <p>Amazon Linux 2 AArch64 Lambda standar: Corretto11</p> <p>Amazon Linux 2 AArch64 standar: 2.0</p> <p>Standar Ubuntu: 5.0</p>
	corretto17	<p>Amazon Linux 2 x86_64 Lambda standar: Corretto17</p> <p>Amazon Linux 2 AArch64 Lambda standar: Corretto17</p> <p>Amazon Linux 2 x86_64 standar:4.0</p> <p>Amazon Linux 2023 x86_64 standar:5.0</p> <p>Amazon Linux 2023 AArch64 standar: 3.0</p> <p>Standar Ubuntu: 6.0</p> <p>Standar Ubuntu: 7.0</p>

Nama runtime	Versi	Citra
	corretto21	Amazon Linux 2 x86_64 Lambda standar: corretto21 Amazon Linux 2 AArch64 Lambda standar: corretto21 Amazon Linux 2023 x86_64 standar:5.0 Amazon Linux 2023 AArch64 standar: 3.0 Standar Ubuntu: 7.0
nodejs	10	Amazon Linux 2 AArch64 standar: 2.0
	12	Amazon Linux 2 AArch64 standar: 2.0 Standar Ubuntu: 5.0
	14	Standar Ubuntu: 5.0
	16	Amazon Linux 2 x86_64 standar:4.0 Standar Ubuntu: 6.0

Nama runtime	Versi	Citra
	18	<p>Amazon Linux 2 x86_64 Lambda standar: nodejs18</p> <p>Amazon Linux 2 AArch64 Lambda standar: nodejs18</p> <p>Amazon Linux 2023 x86_64 standar:5.0</p> <p>Amazon Linux 2023 AArch64 standar: 3.0</p> <p>Standar Ubuntu: 7.0</p>
	20	<p>Amazon Linux 2 x86_64 Lambda standar: nodejs20</p> <p>Amazon Linux 2 AArch64 Lambda standar: nodejs20</p> <p>Amazon Linux 2023 x86_64 standar:5.0</p> <p>Amazon Linux 2023 AArch64 standar: 3.0</p> <p>Standar Ubuntu: 7.0</p>
php	7.3	<p>Amazon Linux 2 AArch64 standar: 2.0</p> <p>Standar Ubuntu: 5.0</p>
	7.4	<p>Amazon Linux 2 AArch64 standar: 2.0</p> <p>Standar Ubuntu: 5.0</p>
	8.0	Standar Ubuntu: 5.0

Nama runtime	Versi	Citra
	8.1	Amazon Linux 2 x86_64 standar:4.0 Amazon Linux 2023 AArch64 standar: 3.0 Standar Ubuntu: 6.0
	8.2	Amazon Linux 2023 x86_64 standar:5.0 Amazon Linux 2023 AArch64 standar: 3.0 Standar Ubuntu: 7.0
	8.3	Amazon Linux 2023 AArch64 standar: 3.0
python	3.7	Amazon Linux 2 AArch64 standar: 2.0 Standar Ubuntu: 5.0
	3.8	Amazon Linux 2 AArch64 standar: 2.0 Standar Ubuntu: 5.0
	3.9	Amazon Linux 2 x86_64 standar:4.0 Amazon Linux 2 AArch64 standar: 2.0 Standar Ubuntu: 5.0
	3.10	Standar Ubuntu: 6.0

Nama runtime	Versi	Citra
	3.11	<p>Amazon Linux 2 x86_64 Lambda standar: python3.11</p> <p>Amazon Linux 2 AArch64 Lambda standar: python3.11</p> <p>Amazon Linux 2023 x86_64 standar:5.0</p> <p>Amazon Linux 2023 AArch64 standar: 3.0</p> <p>Standar Ubuntu: 7.0</p>
	3.12	<p>Amazon Linux 2 x86_64 Lambda standar: python3.12</p> <p>Amazon Linux 2 AArch64 Lambda standar: python3.12</p> <p>Amazon Linux 2023 x86_64 standar:5.0</p> <p>Amazon Linux 2023 AArch64 standar: 3.0</p> <p>Standar Ubuntu: 7.0</p>
ruby	2.6	<p>Amazon Linux 2 AArch64 standar: 2.0</p> <p>Standar Ubuntu: 5.0</p>
	2.7	<p>Amazon Linux 2 AArch64 standar: 2.0</p> <p>Standar Ubuntu: 5.0</p>

Nama runtime	Versi	Citra
	3.1	Amazon Linux 2 x86_64 standar:4.0 Standar Ubuntu: 6.0
	3.2	Amazon Linux 2 x86_64 Lambda standar: ruby3.2 Amazon Linux 2 AArch64 Lambda standar: ruby3.2 Amazon Linux 2023 x86_64 standar:5.0 Amazon Linux 2023 AArch64 standar: 3.0 Standar Ubuntu: 7.0

Runtime gambar Windows

Gambar dasar Windows Server Core 2019 berisi runtime berikut.

Waktu proses platform Windows

Nama runtime	Standar Windows Server Core 2019: versi 1.0	Standar Windows Server Core 2019: versi 2.0	Standar Windows Server Core 2019: versi 3.0
dotnet	3.1	3.1	6.0
	5.0	6.0	7.0
		7.0	8.0
dotnet sdk	3.1	3.1	8.0
	5.0	6.0	

Nama runtime	Standar Windows Server Core 2019: versi 1.0	Standar Windows Server Core 2019: versi 2.0	Standar Windows Server Core 2019: versi 3.0
		7.0	
golang	1.14	1.18	1.21
gradle	6.7	7.6	8.5
jawa	Corretto11	Corretto11 Corretto17	Corretto21
maven	3.6	3.8	3.9
nodejs	14.15	16.19	20.11
php	7.4	8.1	8.3
PowerShell	7.1	7.2	7.4
python	3.8	3.10	3.12
ruby	2.7	3.1	3.3

Versi waktu aktif

Saat menentukan runtime di [runtime-versions](#) bagian file buildspec, Anda dapat menentukan versi tertentu, versi mayor tertentu, dan versi minor terbaru, atau versi terbaru. Tabel berikut mencantumkan runtime yang tersedia dan cara menentukannya. Tidak semua versi runtime tersedia di semua gambar. Pemilihan versi runtime juga tidak didukung untuk gambar kustom. Untuk informasi selengkapnya, lihat [Runtime yang tersedia](#).

Versi runtime platform Ubuntu dan Amazon Linux 2

Nama runtime	Versi	Versi spesifik	Versi minor mayor dan terbaru tertentu	Versi terbaru
android	28	android: 28	android: 28.x	android: latest
	29	android: 29	android: 29.x	
dotnet	3.1	dotnet: 3.1	dotnet: 3.x	dotnet: latest
	5.0	dotnet: 5.0	dotnet: 5.x	
	6.0	dotnet: 6.0	dotnet: 6.x	
	8.0	dotnet: 8.0	dotnet: 8.x	
golang	1.12	golang: 1.12	golang: 1.x	golang: latest
	1.13	golang: 1.13		
	1.14	golang: 1.14		
	1.15	golang: 1.15		
	1.16	golang: 1.16		
	1.18	golang: 1.18		
	1.20	golang: 1.20		
	1.21	golang: 1.21		
java	corretto8	java: corretto	java: corretto .x	java: latest
	corretto11	java: corretto 1	java: corretto 1.x	
	corretto17	java: corretto 7	java: corretto 7.x	

Nama runtime	Versi	Versi spesifik	Versi minor mayor dan terbaru tertentu	Versi terbaru
	corretto21	java: corretto 1	java: corretto 1.x	
nodejs	10	nodejs: 10	nodejs: 10.x	nodejs: latest
	12	nodejs: 12	nodejs: 12.x	
	14	nodejs: 14	nodejs: 14.x	
	16	nodejs: 16	nodejs: 16.x	
	18	nodejs: 18	nodejs: 18.x	
	20	nodejs: 20	nodejs: 20.x	
php	7.3	php: 7.3	php: 7.x	php: latest
	7.4	php: 7.4		
	8.0	php: 8.0	php: 8.x	
	8.1	php: 8.1		
	8.2	php: 8.2		
	8.3	php: 8.3		
python	3.7	python: 3.7	python: 3.x	python: latest
	3.8	python: 3.8		
	3.9	python: 3.9		
	3.10	python: 3.10		
	3.11	python: 3.11		

Nama runtime	Versi	Versi spesifik	Versi minor mayor dan terbaru tertentu	Versi terbaru
	3.12	python: 3.12		
ruby	2.6	ruby: 2.6	ruby: 2.x	ruby: latest
	2.7	ruby: 2.7		
	3.1	ruby: 3.1	ruby: 3.x	
	3.2	ruby: 3.2		

Anda dapat menggunakan spesifikasi build untuk menginstal komponen lain (misalnya, Apache Maven, AWS CLI, Apache Ant, Mocha, RSpec, atau yang serupa) selama fase build. `install` Lihat informasi yang lebih lengkap di [Contoh Buildspec](#).

Bangun mode dan tipe komputasi lingkungan

Di CodeBuild, Anda dapat menentukan image lingkungan komputasi dan runtime yang CodeBuild digunakan untuk menjalankan build Anda. Compute mengacu pada mesin komputasi (CPU, memori, dan sistem operasi) yang dikelola dan dikelola oleh CodeBuild. Gambar lingkungan runtime adalah image kontainer yang berjalan di atas platform komputasi yang Anda pilih, dan menyertakan alat tambahan yang mungkin dibutuhkan build Anda, seperti file. AWS CLI

Topik

- [Tentang mode komputasi](#)
- [Tentang jenis lingkungan](#)

Tentang mode komputasi

CodeBuild menawarkan mode komputasi berikut:

- EC2
- AWS Lambda

EC2 menawarkan fleksibilitas yang dioptimalkan selama pembuatan Anda dan AWS Lambda menawarkan kecepatan awal yang dioptimalkan. AWS Lambda mendukung build yang lebih cepat karena latensi start-up yang lebih rendah. AWS Lambda juga secara otomatis menskalakan, jadi build tidak menunggu dalam antrian untuk dijalankan. Untuk informasi selengkapnya, lihat [Bekerja dengan AWS Lambda komputasi di AWS CodeBuild](#).

Tentang jenis lingkungan

AWS CodeBuild menyediakan lingkungan build dengan memori, vCPU, dan ruang disk berikut yang tersedia untuk mode komputasi EC2:

Jenis komputasi	Nilai ComputeType Lingkungan	Nilai tipe lingkungan	Memori	vCPU	Ruang disk
ARM Kecil	BUILD_GENERAL1_SMALL	ARM_CONTAINER	4 GB	2	50 GB
ARM Besar	BUILD_GENERAL1_LARGE	ARM_CONTAINER	16 GB	8	50 GB
Linux Kecil ¹	BUILD_GENERAL1_SMALL	LINUX_CONTAINER	3 GB	2	64 GB
Linux Sedang ¹	BUILD_GENERAL1_MEDIUM	LINUX_CONTAINER	7 GB	4	128 GB
Linux Besar ¹	BUILD_GENERAL1_LARGE	LINUX_CONTAINER	15 GB	8	128 GB

Jenis komputasi	Nilai ComputeType Lingkungan	Nilai tipe lingkungan	Memori	vCPU	Ruang disk
Linux xLarge	BUILD_GENERAL1_XLARGE	LINUX_CONTAINER	70 GB	36	256 GB
Linux 2xBesar	BUILD_GENERAL1_2XLARGE	LINUX_CONTAINER	145 GB	72	824 GB (SSD)
Linux GPU Kecil	BUILD_GENERAL1_SMALL	LINUX_GPU_CONTAINER	16 GB	4	220 GB
Linux GPU Besar	BUILD_GENERAL1_LARGE	LINUX_GPU_CONTAINER	255 GB	32	50 GB
Windows Medium	BUILD_GENERAL1_MEDIUM	WINDOWS_SERVER_2019_CONTAINER	7 GB	4	128 GB
Jendela Besar	BUILD_GENERAL1_LARGE	WINDOWS_SERVER_2019_CONTAINER	15 GB	8	128 GB

¹ Versi terbaru dari jenis gambar ini di-cache. Jika Anda menentukan versi yang lebih spesifik, CodeBuild maka berikan versi itu alih-alih versi cache. Ini dapat menghasilkan waktu pembuatan yang lebih lama. Misalnya, untuk mendapatkan manfaat dari caching, tentukan `aws/codebuild/amazonlinux2-x86_64-standard:5.0` alih-alih versi yang lebih terperinci, seperti `aws/codebuild/amazonlinux2-x86_64-standard:5.0-1.0.0`

AWS CodeBuild menyediakan lingkungan build dengan memori dan ruang disk berikut yang tersedia untuk mode AWS Lambda komputasi:

Jenis komputasi	Nilai ComputeType Lingkungan	Nilai tipe lingkungan	Memori	Ruang disk
Lambda Lambda 1GB	BUILD_LAMBDA_1GB	ARM_LAMBDA_CONTAINER	1 GB	10 GB
Lambda Lambda 2GB	BUILD_LAMBDA_2GB	ARM_LAMBDA_CONTAINER	2 GB	10 GB
Lambda Lambda 4GB	BUILD_LAMBDA_4GB	ARM_LAMBDA_CONTAINER	4 GB	10 GB
Lambda Lambda 8GB	BUILD_LAMBDA_8GB	ARM_LAMBDA_CONTAINER	8 GB	10 GB
Lengan Lambda 10GB	BUILD_LAMBDA_10GB	ARM_LAMBDA_CONTAINER	10 GB	10 GB
Linux Lambda 1GB	BUILD_LAMBDA_1GB	LINUX_LAMBDA_CONTAINER	1 GB	10 GB
Linux Lambda 2GB	BUILD_LAMBDA_2GB	LINUX_LAMBDA_CONTAINER	2 GB	10 GB
Linux Lambda 4GB	BUILD_LAMBDA_4GB	LINUX_LAMBDA_CONTAINER	4 GB	10 GB

Jenis komputasi	Nilai ComputeType Lingkungan	Nilai tipe lingkungan	Memori	Ruang disk
Linux Lambda 8GB	BUILD_LAMBDA_8GB	LINUX_LAMBDA_CONTAINER	8 GB	10 GB
Linux Lambda 10GB	BUILD_LAMBDA_10GB	LINUX_LAMBDA_CONTAINER	10 GB	10 GB

Saat menggunakan tipe lingkungan lain, Anda disarankan untuk menggunakan image yang di-cache untuk mengurangi waktu pembuatan.

Ruang disk yang terdaftar untuk setiap lingkungan build hanya tersedia di direktori yang ditentukan oleh variabel `CODEBUILD_SRC_DIR` lingkungan.

Untuk memilih jenis komputasi:

- Di CodeBuild konsol, di halaman Create build project wizard atau Edit Build Project, di Environment expand Additional configuration, lalu pilih salah satu opsi dari Compute type. Untuk informasi selengkapnya, lihat [Buat proyek build \(konsol\)](#) atau [Mengubah setelan proyek build \(konsol\)](#).
- Untuk AWS CLI, jalankan `update-project create-project` or, menentukan `computeType` nilai `environment` objek. Untuk informasi selengkapnya, lihat [Buat proyek build \(AWS CLI\)](#) atau [Mengubah setelan proyek pembangunan \(AWS CLI\)](#).
- Untuk AWS SDK, panggil yang setara dengan `UpdateProject` operasi `CreateProject` atau untuk bahasa pemrograman target Anda, dengan menentukan `computeType` nilai setara objek. `environment` Lihat informasi yang lebih lengkap di [AWSReferensi SDK dan alat](#).

Beberapa jenis lingkungan dan komputasi memiliki batasan ketersediaan Wilayah:

- Jenis komputasi Linux GPU Small (`LINUX_GPU_CONTAINER`) hanya tersedia di Wilayah ini:
 - AS Timur (Virginia Utara)
 - AS Barat (Oregon)
 - Asia Pasifik (Tokyo)
 - Kanada (Pusat)

- Eropa (Frankfurt)
- Eropa (Irlandia)
- Eropa (London)
- Jenis komputasi Linux GPU Large (LINUX_GPU_CONTAINER) hanya tersedia di Wilayah ini:
 - AS Timur (Ohio)
 - AS Timur (Virginia Utara)
 - AS Barat (Oregon)
 - Asia Pasifik (Seoul)
 - Asia Pasifik (Singapura)
 - Asia Pasifik (Sydney)
 - Asia Pasifik (Tokyo)
 - Kanada (Pusat)
 - Tiongkok (Beijing)
 - Tiongkok (Ningxia)
 - Eropa (Frankfurt)
 - Eropa (Irlandia)
 - Eropa (London)
- Jenis lingkungan hanya ARM_CONTAINER tersedia di Wilayah ini:
 - AS Timur (Ohio)
 - AS Timur (Virginia Utara)
 - AS Barat (California Utara)
 - AS Barat (Oregon)
 - Asia Pasifik (Hong Kong)
 - Asia Pasifik (Jakarta)
 - Asia Pasifik (Hyderabad)
 - Asia Pasifik (Mumbai)
 - Asia Pasifik (Osaka)
 - Asia Pasifik (Seoul)
 - Asia Pasifik (Singapura)
 - Asia Pasifik (Sydney)

- Asia Pasifik (Tokyo)
- Kanada (Pusat)
- Tiongkok (Beijing)
- Tiongkok (Ningxia)
- Eropa (Frankfurt)
- Eropa (Irlandia)
- Eropa (London)
- Eropa (Milan)
- Eropa (Paris)
- Eropa (Spanyol)
- Eropa (Stockholm)
- Israel (Tel Aviv)
- Timur Tengah (Bahrain)
- Timur Tengah (UEA)
- Amerika Selatan (Sao Paulo)
- Jenis komputasi hanya BUILD_GENERAL1_2XLARGE tersedia di Wilayah ini:
 - AS Timur (Ohio)
 - AS Timur (Virginia Utara)
 - AS Barat (California Utara)
 - AS Barat (Oregon)
 - Asia Pasifik (Hyderabad)
 - Asia Pasifik (Hong Kong)
 - Asia Pasifik (Jakarta)
 - Asia Pasifik (Melbourne)
 - Asia Pasifik (Mumbai)
 - Asia Pasifik (Seoul)
 - Asia Pasifik (Singapura)
 - Asia Pasifik (Sydney)
 - **Asia Pasifik (Tokyo)**
- Kanada (Pusat)

- Tiongkok (Beijing)
- Tiongkok (Ningxia)
- Eropa (Frankfurt)
- Eropa (Irlandia)
- Eropa (London)
- Eropa (Paris)
- Eropa (Spanyol)
- Eropa (Stockholm)
- Eropa (Zürich)
- Israel (Tel Aviv)
- Timur Tengah (Bahrain)
- Timur Tengah (UEA)
- Amerika Selatan (Sao Paulo)
- Mode komputasi AWS Lambda (`ARM_LAMBDA_CONTAINER` dan `LINUX_LAMBDA_CONTAINER`) hanya tersedia di Wilayah ini:
 - AS Timur (N. Virginia)
 - AS Timur (Ohio)
 - AS Barat (Oregon)
 - Asia Pasifik (Mumbai)
 - Asia Pasifik (Singapura)
 - Asia Pasifik (Sydney)
 - Asia Pasifik (Tokyo)
 - Eropa (Frankfurt)
 - Europe (Ireland)
 - Amerika Selatan (Sao Paulo)

Untuk jenis komputasi `BUILD_GENERAL1_2XLARGE`, gambar Docker hingga 100 GB tanpa kompresi didukung.

Note

Untuk gambar lingkungan build kustom, CodeBuild mendukung gambar Docker hingga 50 GB tanpa kompresi di Linux dan Windows, terlepas dari jenis komputasi. Untuk memeriksa ukuran image build Anda, gunakan Docker untuk menjalankan `docker images REPOSITORY:TAG` perintah.

Anda dapat menggunakan Amazon EFS untuk mengakses lebih banyak ruang di container build Anda. Untuk informasi selengkapnya, lihat [Sampel Amazon Elastic File System untuk AWS CodeBuild](#). Jika Anda ingin memanipulasi ruang disk kontainer selama pembuatan, maka build harus berjalan dalam mode istimewa.

Note

Secara default, daemon Docker diaktifkan untuk build non-VPC. Jika Anda ingin menggunakan kontainer Docker untuk build VPC, [lihat Runtime Privilege dan Kemampuan Linux di situs web Docker Docs dan](#) aktifkan mode istimewa. Juga, Windows tidak mendukung mode istimewa.

Kerang dan perintah dalam membangun lingkungan

Anda menyediakan satu set perintah untuk AWS CodeBuild untuk berjalan di lingkungan build selama siklus hidup build (misalnya, menginstal dependensi build dan menguji dan mengkompilasi kode sumber Anda). Ada beberapa cara untuk menentukan perintah ini:

- Buat file spesifikasi build dan sertakan dengan kode sumber Anda. Dalam file ini, tentukan perintah yang ingin Anda jalankan di setiap fase siklus hidup build. Untuk informasi lain, lihat [Buat referensi spesifikasi untuk CodeBuild](#).
- Gunakan konsol CodeBuild untuk membuat proyek build. Masukkan perintah build, untuk Perintah build, masukkan perintah yang ingin Anda jalankan di `build` fase. Untuk informasi selengkapnya, lihat [Buat proyek build \(konsol\)](#).
- Gunakan konsol CodeBuild untuk mengubah pengaturan proyek build. Masukkan perintah build, untuk Perintah build, masukkan perintah yang ingin Anda jalankan di `build` fase. Untuk informasi selengkapnya, lihat [Mengubah setelan proyek build \(konsol\)](#).

- Menggunakan `AWS CLI` atau `AWSSDK` untuk membuat proyek build atau mengubah pengaturan proyek build. Referensi kode sumber yang berisi file `buildspec` dengan perintah Anda, atau tentukan string tunggal yang menyertakan isi file `buildspec` setara. Untuk informasi lebih lanjut, lihat [Membuat proyek pembangunan](#) atau [Mengubah pengaturan proyek pembangunan](#).
- Menggunakan `AWS CLI` atau `AWSSDK` untuk memulai build, menentukan file `buildspec` atau string tunggal yang mencakup isi file `buildspec` setara. Untuk informasi lebih lanjut, lihat deskripsi `buildspecOverride` nilai dalam [Jalankan build](#).

Anda dapat menentukan perintah Shell Command Language (sh). Dalam versi `buildspec 0.1`, CodeBuild menjalankan setiap perintah Shell dalam instance terpisah di lingkungan build. Ini berarti bahwa setiap perintah berjalan secara terpisah dari semua perintah lainnya. Oleh karena itu, secara default, Anda tidak dapat menjalankan satu perintah yang bergantung pada status perintah sebelumnya (misalnya, mengubah direktori atau pengaturan variabel lingkungan). Untuk mengatasi keterbatasan ini, kami sarankan Anda menggunakan versi `0.2`, yang memecahkan masalah ini. Jika Anda harus menggunakan versi `0.1`, kami merekomendasikan pendekatan berikut:

- Sertakan skrip shell dalam kode sumber Anda yang berisi perintah yang ingin Anda jalankan dalam satu contoh shell default. Misalnya, Anda dapat menyertakan file bernama `my-script.sh` dalam kode sumber Anda yang berisi perintah seperti `cd MyDir; mkdir -p mySubDir; cd mySubDir; pwd;`. Kemudian, dalam file `buildspec` Anda, tentukan perintahnya `./my-script.sh`.
- Dalam file `buildspec` Anda atau pada `Perintah build` pengaturan untuk `build` hanya fase, masukkan satu perintah yang mencakup semua perintah yang ingin Anda jalankan dalam satu contoh dari shell default (misalnya, `cd MyDir && mkdir -p mySubDir && cd mySubDir && pwd`).

Jika CodeBuild menemukan kesalahan, kesalahan mungkin lebih sulit untuk memecahkan masalah dibandingkan dengan menjalankan perintah tunggal dalam instance sendiri dari shell default.

Perintah yang dijalankan dalam gambar Windows Server Core menggunakan shell PowerShell.

Variabel lingkungan di lingkungan build

AWS CodeBuild menyediakan beberapa variabel lingkungan yang dapat Anda gunakan dalam perintah build Anda:

`AWS_DEFAULT_REGION`

AWS Wilayah tempat build berjalan (misalnya, `us-east-1`). Variabel lingkungan ini digunakan terutama oleh `AWS CLI`.

AWS_REGION

AWS Wilayah tempat build berjalan (misalnya, `us-east-1`). Variabel lingkungan ini digunakan terutama oleh AWS SDK.

CODEBUILD_BATCH_BUILD_IDENTIFIER

Identifikasi dari membangun dalam batch membangun. Ini ditentukan dalam `buildspec` batch. Untuk informasi selengkapnya, lihat [the section called "Batch buildspec"](#).

CODEBUILD_BUILD_ID

Amazon Resource Name (ARN) penerapan (misalnya, `arn:aws:codebuild:region-ID:account-ID:build/codebuild-demo-project:b1e6661e-e4f2-4156-9ab9-82a19EXAMPLE`).

CODEBUILD_BUILD_ID

CodeBuild ID build (misalnya, `codebuild-demo-project:b1e6661e-e4f2-4156-9ab9-82a19EXAMPLE`).

CODEBUILD_BUILD_IMAGE

CodeBuild Build image identifier (misalnya, `aws/codebuild/standard:2.0`).

CODEBUILD_BUILD_NUMBER

Membangun saat ini untuk proyek.

CODEBUILD_BUILD_SUCCEEDING

Apakah build saat ini berhasil. `0` jika build gagal, atau `1` jika build berhasil.

CODEBUILD_INITIATOR

Entitas yang memulai build. Jika CodePipeline mulai membangun, ini adalah nama pipa (misalnya, `codepipeline/my-demo-pipeline`). Jika pengguna memulai build, ini adalah nama pengguna (misalnya, `MyUserName`). Jika plugin Jenkins untuk CodeBuild memulai build, ini adalah `string CodeBuild-Jenkins-Plugin`.

CODEBUILD_KMS_KEY_ID

Pengenalan AWS KMS kunci yang CodeBuild digunakan untuk mengenkripsi artefak keluaran build (misalnya, `arn:aws:kms:region-ID:account-ID:key/key-ID` atau `alias/key-alias`).

CODEBUILD_LOG_PATH

Nama aliran log di CloudWatch Log untuk build.

CODEBUILD_PUBLIC_BUILD_URL

URL hasil build untuk build ini di situs web build publik. Variabel ini hanya disetel jika proyek build mengaktifkan build publik. Untuk informasi selengkapnya, lihat [Proyek pembangunan publik di AWS CodeBuild](#).

CODEBUILD_RESOLVED_SOURCE_VERSION

Pengenal versi kode sumber build. Isi tergantung pada repositori kode sumber:

CodeCommit, GitHub, ServerGitHub Perusahaan, dan Bitbucket

Variabel ini berisi ID komit.

CodePipeline

Variabel ini berisi revisi sumber yang disediakan oleh CodePipeline.

Jika CodePipeline tidak dapat menyelesaikan revisi sumber, seperti ketika sumber adalah bucket Amazon S3 yang tidak mengaktifkan versi, variabel lingkungan ini tidak disetel.

Amazon S3

Variabel ini tidak diatur.

Bila berlaku, `CODEBUILD_RESOLVED_SOURCE_VERSION` variabel hanya tersedia setelah `DOWNLOAD_SOURCE` fase.

CODEBUILD_SOURCE_REPO_URL

URL ke artefak masukan atau repositori kode sumber. Untuk Amazon S3, ini `s3://` diikuti dengan nama bucket dan jalur ke artefak input. Untuk CodeCommit dan GitHub, ini adalah URL klon repositori. Jika membangun berasal dari CodePipeline, variabel lingkungan ini mungkin kosong.


Untuk sumber sekunder, variabel lingkungan untuk URL repositori sumber sekunder adalah `CODEBUILD_SOURCE_REPO_URL_<sourceIdentifier>`, di mana `<sourceIdentifier>` mengidentifikasi sumber yang Anda buat.

CODEBUILD_SOURCE_VERSION

Format nilai tergantung pada repositori sumber.

- Untuk Amazon S3, ini adalah ID versi yang terkait dengan artefak input.
- Untuk CodeCommit, itu adalah ID komit atau nama cabang yang terkait dengan versi kode sumber yang akan dibangun.

- Untuk GitHub, GitHub Enterprise Server, dan Bitbucket adalah nama tanda yang terkait dengan versi kode sumber yang akan dibangun.

 Note

Untuk membangun GitHub atau GitHub Enterprise Server yang dipicu oleh acara permintaan tarik webhook, itu adalah `pull-request-number`.

Untuk sumber sekunder, variabel lingkungan untuk versi sumber sekunder adalah `CODEBUILD_SOURCE_VERSION_<sourceIdentifier>`, di mana `<sourceIdentifier>` mengidentifikasi sumber yang Anda buat. Untuk informasi selengkapnya, lihat [Beberapa sumber input dan sampel artefak keluaran](#).

CODEBUILD_SRC_DIR

Jalur direktori yang CodeBuild digunakan untuk build (misalnya, `/tmp/src123456789/src`).

Untuk sumber sekunder, variabel lingkungan untuk jalur direktori sumber sekunder adalah `CODEBUILD_SRC_DIR_<sourceIdentifier>`, di mana `<sourceIdentifier>` mengidentifikasi sumber yang Anda buat. Untuk informasi selengkapnya, lihat [Beberapa sumber input dan sampel artefak keluaran](#).

CODEBUILD_START_TIME

Waktu mulai build ditentukan sebagai stempel waktu Unix dalam milidetik.

CODEBUILD_WEBHOOK_ACTOR_ACCOUNT_ID

ID akun pengguna yang memicu acara webhook.

CODEBUILD_WEBHOOK_BASE_REF

Nama referensi dasar dari peristiwa webhook yang memicu build saat ini. Untuk pull request, ini adalah referensi cabang.

CODEBUILD_WEBHOOK_EVENT

Acara webhook yang memicu penerapan penerapan saat ini.

CODEBUILD_WEBHOOK_MERGE_COMMIT

Pengenal komit gabungan yang digunakan untuk build. Variabel ini diatur ketika permintaan tarik Bitbucket digabungkan dengan strategi squash dan cabang pull request ditutup. Dalam kasus

ini, commit pull request asli sudah tidak ada lagi, jadi variabel environment ini berisi identifier dari commit merge yang tergencet.

CODEBUILD_WEBHOOK_PREV_COMMIT

ID dari commit terbaru sebelum peristiwa push webhook yang memicu build saat ini.

CODEBUILD_WEBHOOK_HEAD_REF

Nama referensi kepala acara webhook yang memicu build saat ini. Ini bisa menjadi referensi cabang atau referensi tag.

CODEBUILD_WEBHOOK_TRIGGER

Menunjukkan peristiwa webhook yang memicu build. Variabel ini hanya tersedia untuk build yang dipicu oleh webhook. Nilai diurai dari payload yang dikirim ke CodeBuild oleh GitHub, GitHub Enterprise Server, atau Bitbucket. Format nilai tergantung pada jenis acara yang memicu build.

- Untuk build yang dipicu oleh pull request, itu adalah `pr/pull-request-number`.
- Untuk build yang dipicu dengan membuat cabang baru atau mendorong commit ke cabang, itu adalah `branch/branch-name`.
- Untuk membangun dipicu oleh mendorong tag ke repositori, itu adalah `tag/tag-name`.

RUMAH

Variabel lingkungan ini selalu diatur ke `/root`.

Anda juga dapat menyediakan lingkungan build dengan variabel lingkungan Anda sendiri. Untuk informasi lain, lihat topik berikut:

- [Gunakan CodePipeline dengan CodeBuild](#)
- [Membuat proyek pembangunan](#)
- [Mengubah pengaturan proyek pembangunan](#)
- [Jalankan build](#)
- [Referensi Buildspec](#)

Untuk mencantumkan semua variabel lingkungan yang tersedia di lingkungan build, Anda dapat menjalankan `printenv` perintah (untuk lingkungan build berbasis Linux) atau `"Get-ChildItem Env:"` (untuk lingkungan build berbasis Windows) selama build. Kecuali untuk yang sebelumnya terdaftar, variabel lingkungan yang dimulai dengan `CODEBUILD_` adalah untuk penggunaan CodeBuild internal. Mereka tidak boleh digunakan dalam perintah build Anda.

⚠ Important

Kami sangat mencegah penggunaan variabel lingkungan untuk menyimpan nilai sensitif, terutama AWS mengakses ID kunci. Variabel lingkungan dapat ditampilkan dalam teks biasa menggunakan alat-alat seperti CodeBuild konsol dan AWS CLI.

Kami menyarankan Anda menyimpan nilai sensitif di Amazon EC2 Systems Manager Parameter Store dan kemudian mengambilnya dari buildspec Anda. Untuk menyimpan nilai sensitif, lihat [Penyimpanan dan Panduan Parameter Manajer Sistem: Membuat dan menguji parameter String \(konsol\)](#) di Panduan Pengguna Amazon EC2 Systems Manager. Untuk mengambilnya, lihat `parameter-store` pemetaan di [Sintaks Buildspec](#).

Tugas latar belakang di lingkungan build

Anda dapat menjalankan tugas latar belakang di lingkungan build. Untuk melakukan ini, di buildspec Anda, gunakan `nohup` perintah untuk menjalankan perintah sebagai tugas di latar belakang, bahkan jika proses build keluar dari shell. Gunakan `disown` perintah untuk menghentikan tugas latar belakang yang sedang berjalan secara paksa.

Contoh:

- Mulai proses latar belakang dan tunggu sampai selesai nanti:

```
|  
nohup sleep 30 & echo $! > pidfile  
...  
wait $(cat pidfile)
```

- Mulai proses latar belakang dan jangan menunggu sampai selesai:

```
|  
nohup sleep 30 & disown $!
```

- Mulai proses latar belakang dan bunuh nanti:

```
|  
nohup sleep 30 & echo $! > pidfile  
...  
kill $(cat pidfile)
```

Jalankan build secara lokal denganAWS CodeBuildagen

Anda dapat menggunakanAWS CodeBuildagen untuk menjalankanCodeBuilddibangun di atas mesin lokal. Ada agen yang tersedia untuk platform x86_64 dan ARM.

Anda juga dapat berlangganan untuk menerima pemberitahuan ketika versi baru agen dipublikasikan.

Prasyarat

Sebelum Anda mulai, Anda perlu melakukan hal berikut:

- Instal Git di komputer lokal Anda.
- Instal dan atur[Docker](#)pada mesin lokal Anda.

Mengatur image build

Anda hanya perlu mengatur image build saat pertama kali menjalankan agen, atau saat gambar telah berubah.

Untuk mengatur image build

1. Jika Anda ingin menggunakan gambar Amazon Linux 2 yang dikurasi, Anda dapat menariknya dariCodeBuildrepositori Amazon ECR publik dihttps://gallery.ecr.aws/codebuild/amazonlinux2-x86_64-standarddengan perintah berikut:

```
$ docker pull public.ecr.aws/codebuild/amazonlinux2-x86_64-standard:4.0
```

Atau, jika Anda ingin menggunakan gambar Linux lain, lakukan langkah-langkah berikut:

- a. KloningCodeBuildrepo gambar:

```
$ git clone https://github.com/aws/aws-codebuild-docker-images.git
```

- b. Ubah ke direktori gambar. Untuk contoh ini, gunakanaws/codebuild/standard:5.0gambar:

```
$ cd aws-codebuild-docker-images/ubuntu/standard/5.0
```

- c. Bangun gambar. Ini akan memakan waktu beberapa menit.

```
$ docker build -t aws/codebuild/standard:5.0 .
```

2. Unduh CodeBuild agen.

Untuk mengunduh versi x86_64 agen, jalankan perintah berikut:

```
$ docker pull public.ecr.aws/codebuild/local-builds:latest
```

Untuk mengunduh versi ARM agen, jalankan perintah berikut:

```
$ docker pull public.ecr.aws/codebuild/local-builds:aarch64
```

3. YangCodeBuildagen tersedia dari <https://gallery.ecr.aws/codebuild/local-builds>.

Tanda tangan Secure Hash Algorithm (SHA) untuk versi x86_64 agen adalah:

```
sha256:fac17c6d6c3cb500f6e7975887de1e41d29a9e70a86d6f49f76a2beacfcf967e
```

Tanda tangan SHA untuk versi ARM agen adalah:

```
sha256:57a5dfda63be50edce13dea16dcd5e73e8d8559029658ba08b793c9a7adc68c7
```

Anda dapat menggunakan SHA untuk mengidentifikasi versi agen. Untuk melihat tanda tangan SHA agen, jalankan perintah berikut dan cari SHA di bawahRepoDigests:

```
$ docker inspect public.ecr.aws/codebuild/local-builds:latest
```

JalankanCodeBuildagen

Untuk menjalankanCodeBuildagen

1. Ubah ke direktori yang berisi sumber proyek build Anda.
2. Unduh [codebuild_build.sh](https://raw.githubusercontent.com/aws/aws-codebuild-docker-images/master/local_builds/codebuild_build.sh) naskah:

```
$ curl -O https://raw.githubusercontent.com/aws/aws-codebuild-docker-images/master/local_builds/codebuild_build.sh
$ chmod +x codebuild_build.sh
```


3. Jalankan `codebuild_build.sh` script dan menentukan gambar kontainer Anda dan direktori output.

Untuk menjalankan build x86_64, jalankan perintah berikut:

```
$ ./codebuild_build.sh -i <container-image> -a <output directory>
```

Untuk menjalankan build ARM, jalankan perintah berikut:

```
$ ./codebuild_build.sh -i <container-image> -a <output directory> -l  
public.ecr.aws/codebuild/local-builds:aarch64
```

Ganti `<container-image>` dengan nama gambar kontainer, seperti `aws/codebuild/standard:5.0` atau `public.ecr.aws/codebuild/amazonlinux2-x86_64-standard:4.0`.

Skrip meluncurkan image build dan menjalankan build pada proyek di direktori saat ini. Untuk menentukan lokasi proyek pembangunan, tambahkan `-s <build project directory>` pilihan untuk perintah script.

Menerima pemberitahuan untuk yang baru CodeBuild versi agen

Anda dapat berlangganan pemberitahuan Amazon SNS sehingga Anda akan diberitahu ketika versi baru AWS CodeBuild agen dilepaskan.

Untuk berlangganan CodeBuild Pemberitahuan Agen

1. Buka konsol Amazon SNS di <https://console.aws.amazon.com/sns/v3/home>.
2. Di bilah navigasi, jika belum dipilih, ubah AWS Wilayah ke AS Timur (Virginia N.). Anda harus memilih ini AWS Wilayah karena pemberitahuan Amazon SNS yang Anda langgani dibuat di Wilayah ini.
3. Di panel navigasi, pilih Subscriptions (Langganan).
4. Pilih Create subscription (Buat langganan).
5. Dalam Buat langganan, lakukan hal berikut:
 - a. Untuk Topik ARN, gunakan Amazon Resource Name (ARN) berikut:

```
arn:aws:sns:us-east-1:850632864840:AWS-CodeBuild-Local-Agent-Updates
```

- b. Untuk Protokol, pilih Surel atau SMS.
- c. Untuk Titik akhir, pilih di mana (email atau SMS) untuk menerima notifikasi. Masukkan email atau alamat atau nomor telepon, termasuk kode area.
- d. Pilih Buat langganan.
- e. Pilih Surel untuk menerima email yang meminta Anda mengonfirmasi langganan Anda. Ikuti petunjuk di email untuk menyelesaikan langganan Anda.

Jika Anda tidak ingin lagi menerima notifikasi ini, gunakan prosedur berikut untuk berhenti berlangganan.

Untuk berhenti berlangganan CodeBuild Pemberitahuan Agen

1. Buka konsol Amazon SNS di <https://console.aws.amazon.com/sns/v3/home>.
2. Di panel navigasi, pilih Berlangganan.
3. Pilih langganan dan dari Tindakan, pilih Hapus langganan. Ketika diminta untuk mengonfirmasi, pilih Hapus.

Gunakan AWS CodeBuild dengan Amazon Virtual Private Cloud

Biasanya, AWS CodeBuild tidak dapat mengakses sumber daya dalam VPC. Untuk mengaktifkan akses, Anda harus memberikan informasi konfigurasi khusus VPC tambahan di CodeBuild konfigurasi proyek. Ini termasuk ID VPC, ID subnet VPC, dan ID grup keamanan VPC. Build berkemampuan VPC kemudian dapat mengakses sumber daya di dalam VPC Anda. Untuk informasi selengkapnya tentang menyiapkan VPC di Amazon VPC, lihat [Panduan Pengguna Amazon VPC](#).

Topik

- [Kasus penggunaan](#)
- [Mengizinkan akses Amazon VPC di CodeBuild memproyeksikan](#)
- [Praktik terbaik untuk VPC](#)
- [Memecahkan masalah pengaturan VPC Anda](#)
- [Batasan VPC](#)
- [Gunakan titik akhir VPC](#)
- [AWS CloudFormation Templat VPC](#)
- [Gunakan AWS CodeBuild dengan server proksi](#)

Kasus penggunaan

Konektivitas VPC dari AWS CodeBuild membangun memungkinkan untuk:

- Jalankan pengujian integrasi dari build Anda terhadap data dalam database Amazon RDS yang terisolasi pada subnet pribadi.
- Kueri data di Amazon ElastiCache cluster langsung dari tes.
- Berinteraksi dengan layanan web internal yang dihosting di Amazon EC2, Amazon ECS, atau layanan yang menggunakan Elastic Load Balancing internal.
- Ambil dependensi dari repositori artefak internal yang di-host sendiri, seperti PyPI untuk Python, Maven for Java, dan npm untuk Node.js.
- Akses objek dalam bucket S3 yang dikonfigurasi untuk memungkinkan akses hanya melalui titik akhir Amazon VPC.

- Kueri layanan web eksternal yang memerlukan alamat IP tetap melalui alamat IP Elastic dari gateway NAT atau instance NAT yang terkait dengan subnet Anda.

Build Anda dapat mengakses sumber daya apa pun yang dihosting di VPC Anda.

Mengizinkan akses Amazon VPC diCodeBuildmemproyeksikan

Sertakan pengaturan ini dalam konfigurasi VPC Anda:

- UntukVPC, pilih ID VPC yangCodeBuildmenggunakan.
- UntukSubnet, pilih subnet pribadi dengan terjemahan NAT yang mencakup atau memiliki rute ke sumber daya yang digunakan olehCodeBuild.
- UntukGrup Keamanan, pilih grup keamanan yangCodeBuilddigunakan untuk memungkinkan akses ke sumber daya di VPC.

Untuk menggunakan konsol untuk membuat proyek pembangunan, lihat[Buat proyek build \(konsol\)](#). Saat Anda membuat atau mengubahCodeBuildproject, diVPC, pilih ID VPC, subnet, dan grup keamanan Anda.

Untuk menggunakanAWS CLIuntuk membuat proyek pembangunan, lihat[Buat proyek build \(AWS CLI\)](#). Jika Anda menggunakanAWS CLIbersamaCodeBuild, peran layanan yang digunakan olehCodeBuilduntuk berinteraksi dengan layanan atas nama pengguna IAM harus memiliki kebijakan terlampir. Untuk informasi, lihat [Izinkan CodeBuild akses ke AWS layanan yang diperlukan untuk membuat antarmuka jaringan VPC](#).

YangVPCConfigobjek harus mencakup AndaVPCId,securityGroupIds, dansubnet.

- **VPCId**: Diperlukan. ID VPC yangCodeBuildmenggunakan. Jalankan perintah ini untuk mendapatkan daftar semua ID VPC Amazon di Wilayah Anda:

```
aws ec2 describe-vpcs
```

- **subnet**: Diperlukan. ID subnet yang menyertakan sumber daya yang digunakan olehCodeBuild. Jalankan perintah ini mendapatkan ID ini:

```
aws ec2 describe-subnets --filters "Name=vpc-id,Values=<vpc-id>" --region us-east-1
```

Note

Ganti `us-east-1` dengan Wilayah Anda.

- ***securityGroupIds***: Diperlukan. ID grup keamanan yang digunakan oleh CodeBuild untuk memungkinkan akses ke sumber daya di VPC. Jalankan perintah ini untuk mendapatkan ID ini:

```
aws ec2 describe-security-groups --filters "Name=vpc-id,Values=<vpc-id>" --region us-east-1
```

Note

Ganti `us-east-1` dengan Wilayah Anda.

Praktik terbaik untuk VPC

Gunakan daftar periksa ini saat Anda menyiapkan VPC untuk digunakan CodeBuild.

- Siapkan VPC Anda dengan subnet publik dan pribadi, dan gateway NAT. Gateway NAT harus berada di subnet publik. Untuk informasi selengkapnya, lihat [VPC dengan subnet publik dan pribadi \(NAT\)](#) di Panduan Pengguna Amazon VPC.

⚠ Important

Anda memerlukan gateway NAT atau instance NAT untuk digunakan CodeBuild dengan VPC Anda sehingga CodeBuild dapat mencapai titik akhir publik (misalnya, untuk menjalankan perintah CLI saat menjalankan build). Anda tidak dapat menggunakan gateway internet alih-alih gateway NAT atau instans NAT karena CodeBuild tidak mendukung penetapan alamat IP Elastic ke antarmuka jaringan yang dibuatnya, dan penetapan otomatis alamat IP publik tidak didukung oleh Amazon EC2 untuk antarmuka jaringan apa pun yang dibuat di luar peluncuran instans Amazon EC2.

- Sertakan beberapa Availability Zone dengan VPC Anda.
- Pastikan grup keamanan Anda tidak memiliki lalu lintas masuk (masuknya) yang diizinkan ke build Anda. CodeBuild tidak memiliki persyaratan khusus untuk lalu lintas keluar, tetapi Anda

harus mengizinkan akses ke sumber daya Internet apa pun yang diperlukan untuk build Anda, seperti GitHub atau Amazon S3.

Untuk informasi lebih lanjut, lihat [Aturan grup keamanan](#) di dalam Panduan Pengguna Amazon VPC.

- Siapkan subnet terpisah untuk build Anda.
- Ketika Anda mengatur CodeBuild proyek untuk mengakses VPC Anda, pilih subnet pribadi saja.

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

Untuk informasi lebih lanjut tentang penggunaan AWS CloudFormation untuk mengkonfigurasi VPC untuk menggunakan CodeBuild fitur VPC, lihat [AWS CloudFormation Templat VPC](#).

Memecahkan masalah pengaturan VPC Anda

Gunakan informasi yang muncul dalam pesan kesalahan untuk membantu Anda mengidentifikasi, mendiagnosis, dan mengatasi masalah.

Berikut ini adalah beberapa panduan untuk membantu Anda saat memecahkan masalah umum CodeBuild Kesalahan VPC: Build does not have internet connectivity. Please check subnet network configuration.

1. [Pastikan gateway internet Anda terpasang ke VPC.](#)
2. [Pastikan bahwa tabel rute untuk subnet publik Anda mengarah ke gateway internet.](#)
3. [Pastikan ACL jaringan Anda memungkinkan lalu lintas mengalir.](#)
4. [Pastikan grup keamanan Anda mengizinkan lalu lintas mengalir.](#)
5. [Memecahkan masalah gateway NAT Anda.](#)
6. [Pastikan bahwa tabel rute untuk subnet pribadi mengarah ke gateway NAT.](#)
7. Pastikan bahwa peran layanan yang digunakan oleh CodeBuild untuk berinteraksi dengan layanan atas nama pengguna IAM memiliki izin di [kebijakan ini](#). Untuk informasi selengkapnya, lihat [Membuat peran CodeBuild layanan](#).

Jika CodeBuild hilang izin, Anda mungkin menerima kesalahan yang mengatakan, `Unexpected EC2 error: UnauthorizedOperation`. Kesalahan ini dapat terjadi jika CodeBuild tidak memiliki izin Amazon EC2 yang diperlukan untuk bekerja dengan VPC.

Batasan VPC

- Konektivitas VPC dari CodeBuild tidak didukung di Windows.
- Konektivitas VPC dari CodeBuild tidak didukung untuk VPC bersama.
- Jika Anda mengkonfigurasi VPC, [caching lokal](#) tidak didukung. Mulai 02/28/22, build VPC Anda akan memakan waktu lebih lama karena instans Amazon EC2 baru akan digunakan untuk setiap build.

Gunakan titik akhir VPC

Anda dapat meningkatkan keamanan build Anda dengan mengonfigurasi AWS CodeBuild untuk menggunakan antarmuka VPC endpoint. Endpoint antarmuka didukung oleh PrivateLink, teknologi yang dapat Anda gunakan untuk mengakses Amazon EC2 secara pribadi dan CodeBuild dengan menggunakan alamat IP pribadi. PrivateLink membatasi semua lalu lintas jaringan antara instans terkelola Anda, CodeBuild, dan Amazon EC2 ke jaringan Amazon. (Instans terkelola tidak memiliki akses ke internet.) Selain itu, Anda tidak memerlukan gateway internet, perangkat NAT, atau gateway pribadi virtual. Anda tidak diwajibkan untuk mengonfigurasi PrivateLink, tetapi sebaiknya direkomendasikan. Untuk informasi lebih lanjut tentang PrivateLink dan titik akhir VPC, lihat [Apa itu AWS PrivateLink?](#).

Sebelum Anda membuat titik akhir VPC

Sebelum Anda mengonfigurasi titik akhir VPC untuk AWS CodeBuild, waspadai batasan dan batasan berikut.

Note

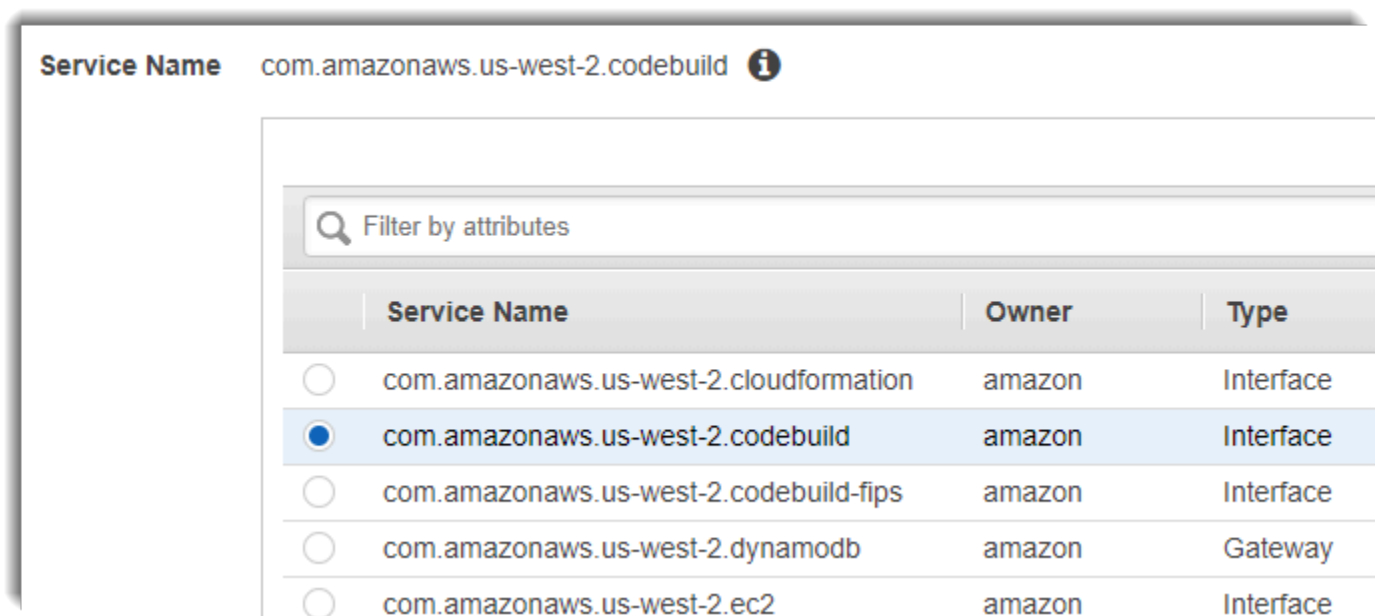
Gunakan [Gerbang NAT](#) jika Anda ingin menggunakan CodeBuild bersama AWS layanan yang tidak mendukung Amazon VPC PrivateLink koneksi.

- Titik akhir VPC mendukung DNS yang disediakan Amazon hanya melalui Amazon Route 53. Jika Anda ingin menggunakan DNS Anda sendiri, Anda dapat menggunakan penerusan DNS bersyarat. Untuk informasi lebih lanjut, lihat [Set opsi DHCP](#) di Panduan Pengguna Amazon VPC.
- Titik akhir VPC saat ini tidak mendukung permintaan lintas wilayah. Pastikan Anda membuat titik akhir Anda di tempat yang sama AWS Region sebagai bucket S3 apa pun yang menyimpan input dan output build Anda. Anda dapat menggunakan konsol Amazon S3 atau [get-bucket-location](#) perintah untuk menemukan lokasi ember Anda. Gunakan titik akhir Amazon S3

khusus Wilayah untuk mengakses bucket Anda (misalnya, `<bucket-name>.s3-us-west-2.amazonaws.com`). Untuk informasi selengkapnya tentang titik akhir khusus Wilayah untuk Amazon S3, lihat [Layanan Penyimpanan Sederhana Amazon](#) di Referensi Umum Amazon Web. Jika Anda menggunakan AWS CLI untuk membuat permintaan ke Amazon S3, menyetel Wilayah default Anda ke Wilayah yang sama tempat bucket Anda dibuat, atau gunakan `--region` parameter dalam permintaan Anda.

Membuat titik akhir VPC untuk CodeBuild

Ikuti instruksi di [Membuat titik akhir antarmuka](#) untuk membuat titik akhir `com.amazonaws.<region>.codebuild`. Ini adalah titik akhir VPC untuk AWS CodeBuild.



`<region>` mewakili pengidentifikasi region untuk sebuah AWS Region yang didukung oleh CodeBuild, seperti `us-east-2` untuk Region US East (Ohio). Untuk daftar yang didukung AWS Daerah, lihat [CodeBuild](#) di AWS Referensi Umum. Titik akhir diisi sebelumnya dengan Wilayah yang Anda tentukan saat Anda masuk AWS. Jika Anda mengubah Wilayah Anda, titik akhir VPC diperbarui sesuai dengan itu.

Membuat kebijakan titik akhir VPC untuk CodeBuild

Anda dapat membuat kebijakan untuk titik akhir Amazon VPC untuk AWS CodeBuild di mana Anda dapat menentukan:

- Principal yang dapat melakukan tindakan.

- Tindakan yang dapat dilakukan.
- Sumber daya yang dapat memiliki tindakan yang dilakukan pada mereka.

Contoh kebijakan berikut menetapkan bahwa semua prinsipal hanya dapat memulai dan melihat build untuk `project-name` proyek.

```
{
  "Statement": [
    {
      "Action": [
        "codebuild:ListBuildsForProject",
        "codebuild:StartBuild",
        "codebuild:BatchGetBuilds"
      ],
      "Effect": "Allow",
      "Resource": "arn:aws:codebuild:region-ID:account-ID:project/project-name",
      "Principal": "*"
    }
  ]
}
```

Untuk informasi lebih lanjut, lihat [Mengendalikan akses ke layanan dengan VPC endpoint](#) di Panduan Pengguna Amazon VPC.

AWS CloudFormationTemplat VPC

AWS CloudFormationmemungkinkan Anda untuk membuat dan menyediakanAWSpenyebaran infrastruktur dapat diduga dan berulang kali, dengan menggunakan file template untuk membuat dan menghapus koleksi sumber daya bersama-sama sebagai satu unit (atumpukan). Untuk informasi selengkapnya, lihat [Panduan Pengguna AWS CloudFormation](#).

Berikut ini adalahAWS CloudFormationTemplate YAML untuk mengkonfigurasi VPC untuk digunakanAWS CodeBuild. File ini juga tersedia di[samples.zip](#).

Description: This template deploys a VPC, with a pair of public and private subnets spread across two Availability Zones. It deploys an internet gateway, with a default route on the public subnets. It deploys a pair of NAT gateways (one in each AZ), and default routes for them in the private subnets.

Parameters:**EnvironmentName:**

Description: An environment name that is prefixed to resource names

Type: String

VpcCIDR:

Description: Please enter the IP range (CIDR notation) for this VPC

Type: String

Default: 10.192.0.0/16

PublicSubnet1CIDR:

Description: Please enter the IP range (CIDR notation) for the public subnet in the first Availability Zone

Type: String

Default: 10.192.10.0/24

PublicSubnet2CIDR:

Description: Please enter the IP range (CIDR notation) for the public subnet in the second Availability Zone

Type: String

Default: 10.192.11.0/24

PrivateSubnet1CIDR:

Description: Please enter the IP range (CIDR notation) for the private subnet in the first Availability Zone

Type: String

Default: 10.192.20.0/24

PrivateSubnet2CIDR:

Description: Please enter the IP range (CIDR notation) for the private subnet in the second Availability Zone

Type: String

Default: 10.192.21.0/24

Resources:**VPC:**

Type: AWS::EC2::VPC

Properties:

CidrBlock: !Ref VpcCIDR

EnableDnsSupport: true

EnableDnsHostnames: true

Tags:

- Key: Name

Value: !Ref EnvironmentName

```
InternetGateway:
  Type: AWS::EC2::InternetGateway
  Properties:
    Tags:
      - Key: Name
        Value: !Ref EnvironmentName

InternetGatewayAttachment:
  Type: AWS::EC2::VPCGatewayAttachment
  Properties:
    InternetGatewayId: !Ref InternetGateway
    VpcId: !Ref VPC

PublicSubnet1:
  Type: AWS::EC2::Subnet
  Properties:
    VpcId: !Ref VPC
    AvailabilityZone: !Select [ 0, !GetAZs '' ]
    CidrBlock: !Ref PublicSubnet1CIDR
    MapPublicIpOnLaunch: true
    Tags:
      - Key: Name
        Value: !Sub ${EnvironmentName} Public Subnet (AZ1)

PublicSubnet2:
  Type: AWS::EC2::Subnet
  Properties:
    VpcId: !Ref VPC
    AvailabilityZone: !Select [ 1, !GetAZs '' ]
    CidrBlock: !Ref PublicSubnet2CIDR
    MapPublicIpOnLaunch: true
    Tags:
      - Key: Name
        Value: !Sub ${EnvironmentName} Public Subnet (AZ2)

PrivateSubnet1:
  Type: AWS::EC2::Subnet
  Properties:
    VpcId: !Ref VPC
    AvailabilityZone: !Select [ 0, !GetAZs '' ]
    CidrBlock: !Ref PrivateSubnet1CIDR
    MapPublicIpOnLaunch: false
    Tags:
```

```
- Key: Name
  Value: !Sub ${EnvironmentName} Private Subnet (AZ1)
```

PrivateSubnet2:

```
Type: AWS::EC2::Subnet
```

Properties:

```
VpcId: !Ref VPC
```

```
AvailabilityZone: !Select [ 1, !GetAZs '' ]
```

```
CidrBlock: !Ref PrivateSubnet2CIDR
```

```
MapPublicIpOnLaunch: false
```

Tags:

```
- Key: Name
```

```
  Value: !Sub ${EnvironmentName} Private Subnet (AZ2)
```

NatGateway1EIP:

```
Type: AWS::EC2::EIP
```

```
DependsOn: InternetGatewayAttachment
```

Properties:

```
Domain: vpc
```

NatGateway2EIP:

```
Type: AWS::EC2::EIP
```

```
DependsOn: InternetGatewayAttachment
```

Properties:

```
Domain: vpc
```

NatGateway1:

```
Type: AWS::EC2::NatGateway
```

Properties:

```
AllocationId: !GetAtt NatGateway1EIP.AllocationId
```

```
SubnetId: !Ref PublicSubnet1
```

NatGateway2:

```
Type: AWS::EC2::NatGateway
```

Properties:

```
AllocationId: !GetAtt NatGateway2EIP.AllocationId
```

```
SubnetId: !Ref PublicSubnet2
```

PublicRouteTable:

```
Type: AWS::EC2::RouteTable
```

Properties:

```
VpcId: !Ref VPC
```

Tags:

```
- Key: Name
```

```
Value: !Sub ${EnvironmentName} Public Routes
```

DefaultPublicRoute:

```
Type: AWS::EC2::Route
DependsOn: InternetGatewayAttachment
Properties:
  RouteTableId: !Ref PublicRouteTable
  DestinationCidrBlock: 0.0.0.0/0
  GatewayId: !Ref InternetGateway
```

PublicSubnet1RouteTableAssociation:

```
Type: AWS::EC2::SubnetRouteTableAssociation
Properties:
  RouteTableId: !Ref PublicRouteTable
  SubnetId: !Ref PublicSubnet1
```

PublicSubnet2RouteTableAssociation:

```
Type: AWS::EC2::SubnetRouteTableAssociation
Properties:
  RouteTableId: !Ref PublicRouteTable
  SubnetId: !Ref PublicSubnet2
```

PrivateRouteTable1:

```
Type: AWS::EC2::RouteTable
Properties:
  VpcId: !Ref VPC
  Tags:
    - Key: Name
      Value: !Sub ${EnvironmentName} Private Routes (AZ1)
```

DefaultPrivateRoute1:

```
Type: AWS::EC2::Route
Properties:
  RouteTableId: !Ref PrivateRouteTable1
  DestinationCidrBlock: 0.0.0.0/0
  NatGatewayId: !Ref NatGateway1
```

PrivateSubnet1RouteTableAssociation:

```
Type: AWS::EC2::SubnetRouteTableAssociation
Properties:
  RouteTableId: !Ref PrivateRouteTable1
  SubnetId: !Ref PrivateSubnet1
```

```
PrivateRouteTable2:
  Type: AWS::EC2::RouteTable
  Properties:
    VpcId: !Ref VPC
    Tags:
      - Key: Name
        Value: !Sub ${EnvironmentName} Private Routes (AZ2)

DefaultPrivateRoute2:
  Type: AWS::EC2::Route
  Properties:
    RouteTableId: !Ref PrivateRouteTable2
    DestinationCidrBlock: 0.0.0.0/0
    NatGatewayId: !Ref NatGateway2

PrivateSubnet2RouteTableAssociation:
  Type: AWS::EC2::SubnetRouteTableAssociation
  Properties:
    RouteTableId: !Ref PrivateRouteTable2
    SubnetId: !Ref PrivateSubnet2

NoIngressSecurityGroup:
  Type: AWS::EC2::SecurityGroup
  Properties:
    GroupName: "no-ingress-sg"
    GroupDescription: "Security group with no ingress rule"
    VpcId: !Ref VPC

Outputs:
  VPC:
    Description: A reference to the created VPC
    Value: !Ref VPC

  PublicSubnets:
    Description: A list of the public subnets
    Value: !Join [ ",", [ !Ref PublicSubnet1, !Ref PublicSubnet2 ]]

  PrivateSubnets:
    Description: A list of the private subnets
    Value: !Join [ ",", [ !Ref PrivateSubnet1, !Ref PrivateSubnet2 ]]

  PublicSubnet1:
    Description: A reference to the public subnet in the 1st Availability Zone
    Value: !Ref PublicSubnet1
```

```
PublicSubnet2:
  Description: A reference to the public subnet in the 2nd Availability Zone
  Value: !Ref PublicSubnet2

PrivateSubnet1:
  Description: A reference to the private subnet in the 1st Availability Zone
  Value: !Ref PrivateSubnet1

PrivateSubnet2:
  Description: A reference to the private subnet in the 2nd Availability Zone
  Value: !Ref PrivateSubnet2

NoIngressSecurityGroup:
  Description: Security group with no ingress rule
  Value: !Ref NoIngressSecurityGroup
```

Gunakan AWS CodeBuild dengan server proksi

Anda dapat menggunakan AWS CodeBuild dengan server proxy untuk mengatur lalu lintas HTTP dan HTTPS ke dan dari internet. Untuk menjalankan CodeBuild dengan server proxy, Anda menginstal server proxy di subnet publik dan CodeBuild di subnet pribadi dalam VPC.

Ada dua kasus penggunaan utama untuk menjalankan CodeBuild di server proxy:

- Ini menghilangkan penggunaan gateway NAT atau instance NAT di VPC Anda.
- Ini memungkinkan Anda menentukan URL yang instansi dalam server proxy dapat mengakses dan URL yang server proxy menolak akses.

Anda dapat menggunakan CodeBuild dengan dua jenis server proxy. Untuk keduanya, server proxy berjalan di subnet publik dan CodeBuild berjalan di subnet pribadi.

- Proksi eksplisit: Jika Anda menggunakan server proxy eksplisit, Anda harus mengkonfigurasi `NO_PROXY`, `HTTP_PROXY`, dan `HTTPS_PROXY` variabel lingkungan di CodeBuild di tingkat proyek. Untuk informasi selengkapnya, lihat [Mengubah pengaturan proyek build di AWS CodeBuild](#) dan [Membuat proyek pembangunan di AWS CodeBuild](#).
- Proksi transparan: Jika Anda menggunakan server proxy transparan, tidak ada konfigurasi khusus yang diperlukan.

Topik

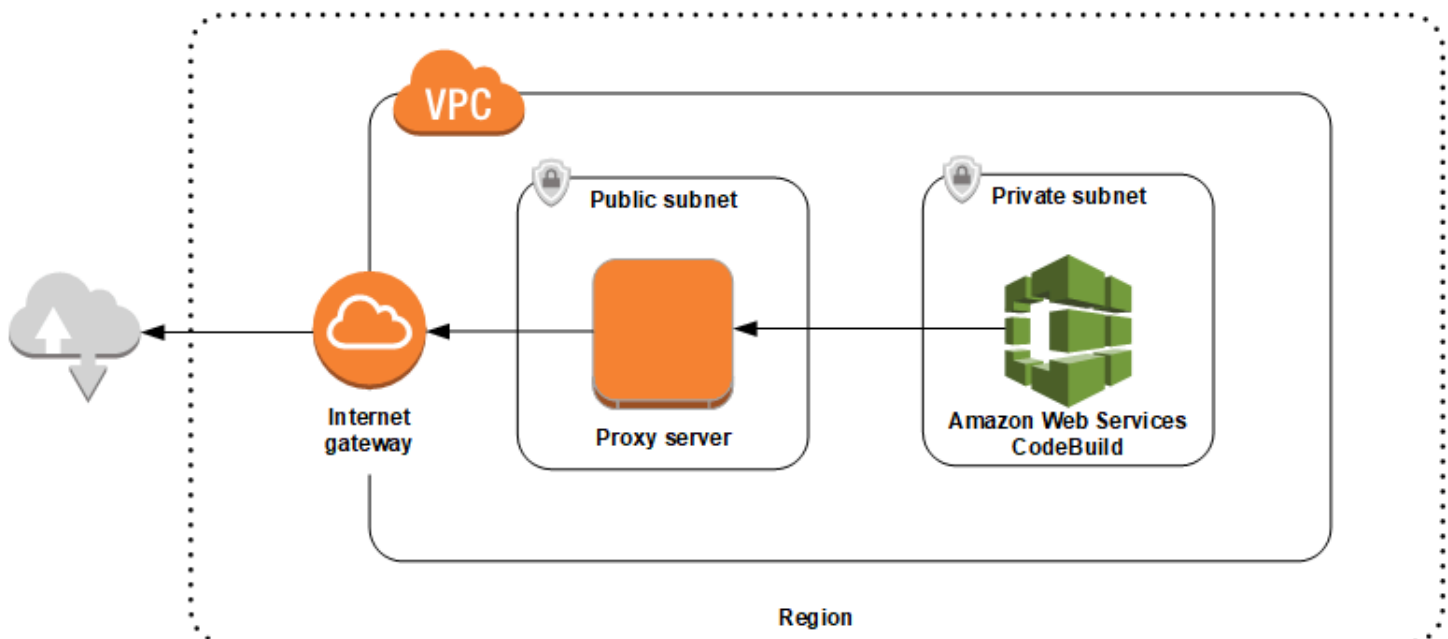
- [Komponen yang diperlukan untuk menjalankan CodeBuild di server proxy](#)
- [Jalankan CodeBuild di server proxy eksplisit](#)
- [Jalankan CodeBuild di server proxy transparan](#)
- [Menjalankan pengelola paket dan alat lain di server proxy](#)

Komponen yang diperlukan untuk menjalankan CodeBuild di server proxy

Anda perlu komponen-komponen ini untuk menjalankan AWS CodeBuild di server proxy transparan atau eksplisit:

- VPC.
- Satu subnet publik di VPC Anda untuk server proxy.
- Satu subnet privat di VPC Anda untuk CodeBuild.
- Gateway internet yang memungkinkan pergantian antara VPC dan internet.

Diagram berikut menunjukkan bagaimana komponen berinteraksi.



Mengatur VPC, subnet, dan gateway jaringan

Langkah-langkah berikut diperlukan untuk menjalankan AWS CodeBuild di server proxy transparan atau eksplisit.

1. Buat sebuah VPC. Untuk informasi, lihat [Membuat VPC](#) di Panduan Pengguna Amazon VPC.
2. Membuat dua subnet di VPC Anda. Salah satunya adalah subnet publik `Public Subnet` di mana server proxy Anda berjalan. Yang lainnya adalah subnet pribadi bernama `Private Subnet` di mana CodeBuild berjalan.

Untuk informasi, lihat [Membuat subnet di VPC](#).

3. Buat dan lampirkan gateway internet ke VPC Anda. Untuk informasi selengkapnya, lihat [Membuat dan melampirkan gateway internet](#).
4. Tambahkan aturan ke tabel rute default yang merutekan lalu lintas keluar dari VPC (0.0.0/0) ke gateway internet. Untuk informasi, lihat [Menambahkan dan menghapus rute dari tabel rute](#).
5. Tambahkan aturan ke grup keamanan default VPC Anda yang mengizinkan lalu lintas SSH (TCP 22) dari VPC Anda (0.0.0/0).
6. Ikuti petunjuknya [Meluncurkan menggunakan launch instance wizard](#) di Panduan Pengguna Amazon EC2 untuk meluncurkan instans Amazon Linux. Saat Anda menjalankan wizard, pilih opsi berikut:
 - Masuk `Pilih Jenis Instans`, pilih Amazon Linux Machine Image (AMI).
 - Masuk `Subnet`, pilih subnet publik yang Anda buat sebelumnya dalam topik ini. Jika Anda menggunakan nama yang disarankan, itu `Subnet publik`.
 - Masuk `Menetapkan IP Publik`, pilih `Aktifkan`.
 - Pada `Konfigurasi Kelompok Keamanan` halaman, untuk `Menetapkan grup keamanan`, pilih `Pilih grup keamanan yang sudah ada`. Selanjutnya, pilih grup keamanan default.
 - Setelah Anda memilih `Luncurkan`, pilih key pair yang sudah ada atau membuat.

Pilih pengaturan default untuk semua opsi lainnya.

7. Setelah instans EC2 Anda berjalan, nonaktifkan pemeriksaan sumber/tujuan. Untuk informasi, lihat [Menonaktifkan pemeriksaan Sumber/Tujuan](#) di Panduan Pengguna Amazon VPC.
8. Buat tabel rute di VPC Anda. Tambahkan aturan ke tabel rute yang mengarahkan lalu lintas yang ditakdirkan untuk internet ke server proxy Anda. Kaitkan tabel rute ini dengan subnet pribadi Anda. Hal ini diperlukan agar permintaan keluar dari instance di subnet pribadi Anda, di mana CodeBuild berjalan, selalu dialihkan melalui server proxy.

Instal dan konfigurasi server proxy

Ada banyak server proxy yang bisa dipilih. Server proxy open-source, Squid, digunakan di sini untuk menunjukkan bagaimana AWS CodeBuild berjalan di server proxy. Anda dapat menerapkan konsep yang sama ke server proxy lainnya.

Untuk menginstal Squid, gunakan repo yum dengan menjalankan perintah berikut:

```
sudo yum update -y
sudo yum install -y squid
```

Setelah Anda menginstal Squid, mengedit `nyasquid.conf` menggunakan petunjuk nanti dalam topik ini.

Mengkonfigurasi Squid untuk lalu lintas HTTPS

Untuk HTTPS, lalu lintas HTTP dikemas dalam koneksi Transport Layer Security (TLS). Squid menggunakan fitur yang disebut [SslPeekAndSplice](#) untuk mengambil Server Name Indication (SNI) dari inisiasi TLS yang berisi host internet yang diminta. Hal ini diperlukan agar Squid tidak perlu unencrypt lalu lintas HTTPS. Untuk mengaktifkan `SslPeekAndSplice`, Squid memerlukan sertifikat. Buat sertifikat ini menggunakan OpenSSL:

```
sudo mkdir /etc/squid/ssl
cd /etc/squid/ssl
sudo openssl genrsa -out squid.key 2048
sudo openssl req -new -key squid.key -out squid.csr -subj "/C=XX/ST=XX/L=squid/O=squid/CN=squid"
sudo openssl x509 -req -days 3650 -in squid.csr -signkey squid.key -out squid.crt
sudo cat squid.key squid.crt | sudo tee squid.pem
```

Note

Untuk HTTP, Squid tidak memerlukan konfigurasi. Dari semua pesan permintaan HTTP/1.1, dapat mengambil kolom header `host`, yang menentukan host internet yang sedang diminta.

Jalankan CodeBuild di server proxy eksplisit

Topik

- [Mengkonfigurasi Squid sebagai server proxy eksplisit](#)
- [Membuat proyek CodeBuild](#)
- [Contoh server proksisquid.conf](#)

Untuk menjalankan AWS CodeBuild di server proxy eksplisit, Anda harus mengkonfigurasi server proxy untuk mengizinkan atau menolak lalu lintas ke dan dari situs eksternal, dan kemudian mengkonfigurasi `HTTP_PROXY` dan `HTTPS_PROXY` variabel lingkungan.

Mengkonfigurasi Squid sebagai server proxy eksplisit

Untuk mengkonfigurasi server proxy Squid menjadi eksplisit, Anda harus membuat modifikasi berikut untuk `etc/squid/squid.conf` file:

- Hapus aturan daftar kontrol akses default berikut.

```
acl localnet src 10.0.0.0/8
acl localnet src 172.16.0.0/12
acl localnet src 192.168.0.0/16
acl localnet src fc00::/7
acl localnet src fe80::/10
```

Tambahkan yang berikut di tempat aturan ACL default yang Anda hapus. Baris pertama memungkinkan permintaan dari VPC Anda. Dua baris berikutnya memberikan akses server proxy Anda ke URL tujuan yang mungkin digunakan oleh AWS CodeBuild. Mengedit ekspresi reguler di baris terakhir untuk menentukan ember S3 atau repositori CodeCommit dalam sebuah AWS Wilayah. Misalnya:

- Jika sumber Anda adalah Amazon S3, gunakan perintah `acl download_src dstdomain .*s3\.us-west-1\.amazonaws\.com` untuk memberikan akses ke ember S3 di `us-west-1` Wilayah.
- Jika sumber Anda AWS CodeCommit, menggunakan `git-codecommit.<your-region>.amazonaws.com` untuk menambahkan AWS Wilayah ke daftar izinkan.

```
acl localnet src 10.1.0.0/16 #Only allow requests from within the VPC
acl allowed_sites dstdomain .github.com #Allows to download source from GitHub
acl allowed_sites dstdomain .bitbucket.com #Allows to download source from Bitbucket
acl download_src dstdomain .*\.amazonaws\.com #Allows to download source from
Amazon S3 or CodeCommit
```

- Ganti `http_access allow localnet` dengan hal berikut:

```
http_access allow localnet allowed_sites
http_access allow localnet download_src
```

- Jika Anda ingin mengunggah log dan artefak, lakukan salah satu hal berikut:

1. Sebelum `http_access deny all` pernyataan, masukkan pernyataan berikut. Mereka memungkinkan CodeBuild untuk mengakses CloudWatch dan Amazon S3. Akses ke CloudWatch diperlukan agar CodeBuild dapat membuat log CloudWatch. Akses ke Amazon S3 diperlukan untuk mengunggah artefak dan caching Amazon S3.

- ```
https_port 3130 cert=/etc/squid/ssl/squid.pem ssl-bump intercept
acl SSL_port port 443
http_access allow SSL_port
acl allowed_https_sites ssl::server_name .amazonaws.com
acl step1 at_step SslBump1
acl step2 at_step SslBump2
acl step3 at_step SslBump3
ssl_bump peek step1 all
ssl_bump peek step2 allowed_https_sites
ssl_bump splice step3 allowed_https_sites
ssl_bump terminate step2 all
```

- Setelah Anda menyimpan `squid.conf`, jalankan perintah berikut:

```
sudo iptables -t nat -A PREROUTING -p tcp --dport 443 -j REDIRECT --to-port 3130
sudo service squid restart
```

2. Tambahkan `proxy` ke file `buildspec` Anda. Untuk informasi selengkapnya, lihat [Sintaks Buildspec](#).

```
version: 0.2
proxy:
 upload-artifacts: yes
 logs: yes
phases:
 build:
 commands:
 - command
```

### Note

Jika Anda menerima kesalahan `timeout RequestError`, lihat [RequestError kesalahan batas waktu saat berjalan CodeBuild di server proxy](#).

Untuk informasi selengkapnya, lihat [Contoh server proksisquid.conf](#) dalam topik ini.

## Membuat proyek CodeBuild

Untuk menjalankan AWS CodeBuild dengan server proxy eksplisit Anda, atur `HTTP_PROXY` dan `HTTPS_PROXY` variabel lingkungan dengan alamat IP pribadi instans EC2 yang Anda buat untuk server proxy dan port 3128 di tingkat proyek. Alamat IP pribadi terlihat seperti `http://your-ec2-private-ip-address:3128`. Untuk informasi selengkapnya, lihat [Membuat proyek pembangunan di AWS CodeBuild](#) dan [Mengubah pengaturan proyek build di AWS CodeBuild](#).

Gunakan perintah berikut untuk menampilkan log akses proxy Squid:

```
sudo tail -f /var/log/squid/access.log
```

## Contoh server proksi `squid.conf` fail

Berikut ini adalah contoh `squid.conf` file yang dikonfigurasi untuk server proxy eksplisit.

```
acl localnet src 10.0.0.0/16 #Only allow requests from within the VPC
add all URLs to be whitelisted for download source and commands to be run in build
environment
acl allowed_sites dstdomain .github.com #Allows to download source from github
acl allowed_sites dstdomain .bitbucket.com #Allows to download source from bitbucket
acl allowed_sites dstdomain ppa.launchpad.net #Allows to run apt-get in build
environment
acl download_src dstdom_regex .*\.amazonaws\.com #Allows to download source from S3
or CodeCommit
acl SSL_ports port 443
acl Safe_ports port 80 # http
acl Safe_ports port 21 # ftp
acl Safe_ports port 443 # https
acl Safe_ports port 70 # gopher
acl Safe_ports port 210 # wais
acl Safe_ports port 1025-65535 # unregistered ports
acl Safe_ports port 280 # http-mgmt
acl Safe_ports port 488 # gss-http
acl Safe_ports port 591 # filemaker
acl Safe_ports port 777 # multiling http
acl CONNECT method CONNECT
#
Recommended minimum Access Permission configuration:
#
Deny requests to certain unsafe ports
```

```
http_access deny !Safe_ports
Deny CONNECT to other than secure SSL ports
http_access deny CONNECT !SSL_ports
Only allow cachemgr access from localhost
http_access allow localhost manager
http_access deny manager
We strongly recommend the following be uncommented to protect innocent
web applications running on the proxy server who think the only
one who can access services on "localhost" is a local user
#http_access deny to_localhost
#
INSERT YOUR OWN RULE(S) HERE TO ALLOW ACCESS FROM YOUR CLIENTS
#
Example rule allowing access from your local networks.
Adapt localnet in the ACL section to list your (internal) IP networks
from where browsing should be allowed
http_access allow localnet allowed_sites
http_access allow localnet download_src
http_access allow localhost
Add this for CodeBuild to access CWL end point, caching and upload artifacts S3
bucket end point
https_port 3130 cert=/etc/squid/ssl/squid.pem ssl-bump intercept
acl SSL_port port 443
http_access allow SSL_port
acl allowed_https_sites ssl::server_name .amazonaws.com
acl step1 at_step SslBump1
acl step2 at_step SslBump2
acl step3 at_step SslBump3
ssl_bump peek step1 all
ssl_bump peek step2 allowed_https_sites
ssl_bump splice step3 allowed_https_sites
ssl_bump terminate step2 all
And finally deny all other access to this proxy
http_access deny all
Squid normally listens to port 3128
http_port 3128
Uncomment and adjust the following to add a disk cache directory.
#cache_dir ufs /var/spool/squid 100 16 256
Leave coredumps in the first cache dir
coredump_dir /var/spool/squid
#
Add any of your own refresh_pattern entries above these.
#
refresh_pattern ^ftp: 1440 20% 10080
```

```
refresh_pattern ^gopher: 1440 0% 1440
refresh_pattern -i (/cgi-bin/|\?) 0 0% 0
refresh_pattern . 0 20% 4320
```

## Jalankan CodeBuild di server proxy transparan

Untuk menjalankan AWS CodeBuild di server proxy transparan, Anda harus mengkonfigurasi server proxy dengan akses ke situs web dan domain yang berinteraksi dengannya.

### Mengkonfigurasi Squid sebagai server proxy transparan

Untuk mengkonfigurasi server proxy agar transparan, Anda harus memberikannya akses ke domain dan situs web yang ingin Anda akses. Untuk menjalankan AWS CodeBuild dengan server proxy transparan, Anda harus memberikan akses ke `amazonaws.com`. Anda juga harus memberikan akses ke situs web lain yang digunakan CodeBuild. Ini bervariasi, tergantung pada bagaimana Anda membuat proyek CodeBuild Anda. Contoh website adalah mereka untuk repositori seperti GitHub, Bitbucket, Yum, dan Maven. Untuk memberikan akses Squid ke domain dan situs web tertentu, gunakan perintah yang serupa dengan berikut ini untuk memperbarui `squid.conf` file. Perintah sampel ini memberikan akses ke `amazonaws.com`, `github.com`, dan `bitbucket.com`. Anda dapat mengedit contoh ini untuk memberikan akses ke situs web lain.

```
cat | sudo tee /etc/squid/squid.conf #EOF
visible_hostname squid
#Handling HTTP requests
http_port 3129 intercept
acl allowed_http_sites dstdomain .amazonaws.com
#acl allowed_http_sites dstdomain domain_name [uncomment this line to add another
domain]
http_access allow allowed_http_sites
#Handling HTTPS requests
https_port 3130 cert=/etc/squid/ssl/squid.pem ssl-bump intercept
acl SSL_port port 443
http_access allow SSL_port
acl allowed_https_sites ssl::server_name .amazonaws.com
acl allowed_https_sites ssl::server_name .github.com
acl allowed_https_sites ssl::server_name .bitbucket.com
#acl allowed_https_sites ssl::server_name [uncomment this line to add another website]
acl step1 at_step SslBump1
acl step2 at_step SslBump2
acl step3 at_step SslBump3
ssl_bump peek step1 all
```

```
ssl_bump peek step2 allowed_https_sites
ssl_bump splice step3 allowed_https_sites
ssl_bump terminate step2 all
http_access deny all
EOF
```

Permintaan masuk dari instance di subnet pribadi harus dialihkan ke port Squid. Squid mendengarkan pada port 3129 untuk lalu lintas HTTP (bukan 80) dan 3130 untuk lalu lintas HTTPS (bukan 443). Menggunakan iptables perintah untuk rute lalu lintas:

```
sudo iptables -t nat -A PREROUTING -p tcp --dport 80 -j REDIRECT --to-port 3129
sudo iptables -t nat -A PREROUTING -p tcp --dport 443 -j REDIRECT --to-port 3130
sudo service iptables save
sudo service squid start
```

## Membuat proyek CodeBuild

Setelah Anda mengkonfigurasi server proxy Anda, Anda dapat menggunakannya dengan AWS CodeBuild dalam subnet pribadi tanpa konfigurasi lainnya. Setiap permintaan HTTP dan HTTPS berjalan melalui server proxy publik. Gunakan perintah berikut untuk menampilkan log akses proxy Squid:

```
sudo tail -f /var/log/squid/access.log
```

## Menjalankan pengelola paket dan alat lain di server proxy

Untuk menjalankan alat, seperti manajer paket, di server proxy

1. Tambahkan alat ke daftar izinkan di server proxy Anda dengan menambahkan pernyataan `kesquid.conf` file.
2. Tambahkan baris ke file `buildspec` Anda yang menunjuk ke titik akhir pribadi server proxy Anda.

Contoh-contoh berikut mendemonstrasikan cara melakukannya `apt-get`, `curl`, dan `maven`. Jika Anda menggunakan alat yang berbeda, prinsip yang sama berlaku. Menambahkannya ke daftar memungkinkan `disquid.conf` file dan tambahkan perintah ke file `buildspec` Anda untuk membuat CodeBuild menyadari endpoint server proxy Anda.



## Untuk menjalankan `apt-get` di server proksi

1. Tambahkan pernyataan berikut ke `squid.conf` untuk ditambahkan ke daftar memungkinkan di server proxy Anda. Tiga baris pertama memungkinkan `apt-get` untuk berjalan di lingkungan membangun.

```
acl allowed_sites dstdomain ppa.launchpad.net # Required for apt-get to run in the
build environment
acl apt_get dstdom_regex .*\.launchpad.net # Required for CodeBuild to run apt-get
in the build environment
acl apt_get dstdom_regex .*\.ubuntu.com # Required for CodeBuild to run apt-get
in the build environment
http_access allow localnet allowed_sites
http_access allow localnet apt_get
```

2. Tambahkan pernyataan berikut di file `buildspec` Anda sehingga `apt-get` perintah mencari konfigurasi proxy di `/etc/apt/apt.conf.d/00proxy`.

```
echo 'Acquire::http::Proxy "http://<private-ip-of-proxy-server>:3128";
Acquire::https::Proxy "http://<private-ip-of-proxy-server>:3128";
Acquire::ftp::Proxy "http://<private-ip-of-proxy-server>:3128";' > /etc/apt/
apt.conf.d/00proxy
```

## Untuk menjalankan `curl` di server proksi

1. Tambahkan yang berikut ini ke `squid.conf` untuk ditambahkan ke daftar memungkinkan di lingkungan build Anda.

```
acl allowed_sites dstdomain ppa.launchpad.net # Required to run apt-get in the
build environment
acl allowed_sites dstdomain google.com # Required for access to a website. This
example uses www.google.com.
http_access allow localnet allowed_sites
http_access allow localnet apt_get
```

2. Tambahkan pernyataan berikut di file `buildspec` Anda `curl` menggunakan server proxy pribadi untuk mengakses situs web yang Anda tambahkan ke `squid.conf`. Dalam contoh ini, situs `webgoogle.com`.

```
curl -x <private-ip-of-proxy-server>:3128 https://www.google.com
```

## Untuk menjalankan `maven` di server proksi

1. Tambahkan yang berikut ini ke `settings.xml` untuk menambahkan konfigurasi yang memungkinkan di lingkungan build Anda.

```
acl allowed_sites dstdomain ppa.launchpad.net # Required to run apt-get in the
build environment
acl maven dstdom_regex .*\.maven.org # Allows access to the maven repository in the
build environment
http_access allow localnet allowed_sites
http_access allow localnet maven
```

2. Tambahkan pernyataan berikut ke file `buildspec` Anda.

```
maven clean install -DproxySet=true -DproxyHost=<private-ip-of-proxy-server> -
DproxyPort=3128
```

# Bekerja dengan membangun proyek dan membangun diAWS CodeBuild

Untuk memulai, ikuti langkah-langkah di [Membuat proyek pembangunan](#), dan kemudian ikuti langkah-langkah di [Jalankan build](#). Untuk informasi lebih lanjut membangun proyek dan pembangunan, lihat topik berikut.

## Topik

- [Bekerja dengan membangun proyek](#)
- [Bekerja dengan membangun diAWS CodeBuild](#)

## Bekerja dengan membangun proyek

Proyek build mencakup informasi tentang cara menjalankan build, termasuk tempat mendapatkan kode sumber, lingkungan build mana yang akan digunakan, perintah build mana yang akan dijalankan, dan tempat menyimpan output build.

Anda dapat melakukan tugas-tugas ini saat bekerja dengan proyek build:

## Topik

- [Membuat proyek pembangunan diAWS CodeBuild](#)
- [Membuat aturan notifikasi](#)
- [Melihat daftar nama proyek build diAWS CodeBuild](#)
- [Lihat detail proyek build diAWS CodeBuild](#)
- [Membangun caching di AWS CodeBuild](#)
- [Bangun pemicu di AWS CodeBuild](#)
- [Menggunakan webhooks dengan AWS CodeBuild](#)
- [Mengubah pengaturan proyek build diAWS CodeBuild](#)
- [Menghapus proyek pembangunan diAWS CodeBuild](#)
- [Bekerja dengan proyek bersama](#)
- [Proyek penandaan diAWS CodeBuild](#)
- [Batch dibangun di AWS CodeBuild](#)
- [GitHubPelari aksi diAWS CodeBuild](#)

- [Proyek pembangunan publik diAWS CodeBuild](#)

## Membuat proyek pembangunan diAWS CodeBuild

Anda dapat menggunakanAWS CodeBuildkonsol,AWS CLI, atauAWSSDK untuk membuat proyek build.

### Prasyarat

Sebelum membuat proyek build, jawab pertanyaan-pertanyaan di[Rencanakan pembangunan](#).

### Topik

- [Buat proyek build \(konsol\)](#)
- [Buat proyek build \(AWS CLI\)](#)
- [Membuat proyek pembangunan \(AWSSDK\)](#)
- [Membuat proyek pembangunan \(AWS CloudFormation\)](#)

### Buat proyek build (konsol)

Buka AWS CodeBuild konsol di <https://console.aws.amazon.com/codesuite/codebuild/home>.

Jika halaman CodeBuild informasi ditampilkan, pilih Buat proyek build. Jika tidak, pada panel navigasi, perluas Build, pilih Build projects, lalu pilih Create build project.

Pilih Buat proyek build.

Isi bagian berikut. Setelah selesai, pilih Buat proyek build di bagian bawah halaman.

Bagian:

- [Konfigurasi proyek](#)
- [Sumber](#)
- [Environment](#)
- [Buildspec](#)
- [Konfigurasi Batch](#)
- [Artifacts](#)
- [Log](#)

## Konfigurasi proyek

### Nama proyek

Masukkan nama untuk proyek build ini. Membangun nama proyek harus unik di setiap AWS akun.

### Deskripsi

Masukkan deskripsi opsional proyek build untuk membantu pengguna lain memahami untuk apa proyek ini digunakan.

### Membangun lencana

(Opsional) Pilih Aktifkan lencana build untuk membuat status build proyek Anda terlihat dan dapat disematkan. Untuk informasi selengkapnya, lihat [Membangun sampel lencana](#).

#### Note

Build badge tidak berlaku jika penyedia sumber Anda adalah Amazon S3.

### Aktifkan batas build bersamaan

(Opsional) Jika Anda ingin membatasi jumlah build bersamaan untuk proyek ini, lakukan langkah-langkah berikut:

1. Pilih Batasi jumlah build bersamaan yang dapat dimulai proyek ini.
2. Dalam batas build bersamaan, masukkan jumlah maksimum build bersamaan yang diizinkan untuk project ini. Batas ini tidak boleh lebih besar dari batas build bersamaan yang ditetapkan untuk akun. Jika Anda mencoba memasukkan angka yang lebih besar dari batas akun, pesan kesalahan akan ditampilkan.

Pembuatan baru hanya dimulai jika jumlah pembuatan saat ini kurang dari atau sama dengan batas ini. Jika jumlah pembuatan saat ini memenuhi batas ini, pembuatan baru ditutup dan tidak dijalankan.

### Informasi tambahan

(Opsional) Untuk Tag, masukkan nama dan nilai tag apa pun yang Anda inginkan untuk digunakan oleh AWS layanan pendukung. Gunakan Tambah baris untuk menambahkan tag. Anda dapat menambahkan hingga 50 tanda.

## Sumber

### Penyedia sumber

Pilih jenis penyedia kode sumber. Gunakan daftar berikut untuk membuat pilihan yang sesuai untuk penyedia sumber Anda:

#### Note

CodeBuild tidak mendukung Bitbucket Server.

### Amazon S3

#### Bucket

Pilih nama bucket input yang berisi kode sumber.

#### Kunci objek S3 atau folder S3

Masukkan nama file ZIP atau path ke folder yang berisi kode sumber. Masukkan garis miring (/) untuk mengunduh semua yang ada di bucket S3.

#### Versi sumber

Masukkan ID versi objek yang mewakili build file input Anda. Untuk informasi selengkapnya, lihat [Contoh versi sumber dengan AWS CodeBuild](#).


### CodeCommit

#### Repositori

Pilih repositori yang ingin Anda gunakan.

#### Jenis referensi

Pilih Cabang, tag Git, atau ID Komit untuk menentukan versi kode sumber Anda. Untuk informasi selengkapnya, lihat [Contoh versi sumber dengan AWS CodeBuild](#).

 Note

Kami menyarankan Anda memilih nama cabang Git yang tidak terlihat seperti ID komit, seperti 811dd1ba1aba14473856cee38308caed7190c0d atau 5392f7. Ini membantu Anda menghindari tabrakan checkout Git dengan commit yang sebenarnya.

## Kedalaman klon Git

Pilih untuk membuat klon dangkal dengan riwayat terpotong ke jumlah komit yang ditentukan. Jika Anda ingin klon penuh, pilih Penuh.

## Submodul Git

Pilih Gunakan submodul Git jika Anda ingin menyertakan submodul Git dalam repositori Anda.

## Bitbucket


## Repositori

Pilih Connect menggunakan OAuth atau Connect with a Bitbucket app password dan ikuti petunjuk untuk menghubungkan (atau menyambung kembali) ke Bitbucket.

Pilih repositori publik atau repositori di akun Anda.

## Versi sumber

Masukkan cabang, ID komit, tag, atau referensi dan ID komit. Lihat informasi yang lebih lengkap di [Contoh versi sumber dengan AWS CodeBuild](#)

 Note

Kami menyarankan Anda memilih nama cabang Git yang tidak terlihat seperti ID komit, seperti 811dd1ba1aba14473856cee38308caed7190c0d atau 5392f7. Ini membantu Anda menghindari tabrakan checkout Git dengan commit yang sebenarnya.

## Kedalaman klon Git

Pilih kedalaman klon Git untuk membuat klon dangkal dengan riwayat terpotong ke jumlah komit yang ditentukan. Jika Anda ingin klon penuh, pilih Penuh.

## Submodul Git

Pilih Gunakan submodul Git jika Anda ingin menyertakan submodul Git dalam repositori Anda.

### Membangun status

Pilih Laporkan status build ke penyedia sumber saat build dimulai dan selesai jika ingin status awal dan penyelesaian build dilaporkan ke penyedia sumber.

Untuk dapat melaporkan status build ke penyedia sumber, pengguna yang terkait dengan penyedia sumber harus memiliki akses tulis ke repo. Jika pengguna tidak memiliki akses tulis, status build tidak dapat diperbarui. Untuk informasi selengkapnya, lihat [Akses penyedia sumber](#).

Untuk konteks Status, masukkan nilai yang akan digunakan untuk name parameter dalam status komit Bitbucket. Untuk informasi selengkapnya, lihat [build](#) dalam dokumentasi Bitbucket API.

Untuk URL Target, masukkan nilai yang akan digunakan untuk url parameter dalam status komit Bitbucket. Untuk informasi selengkapnya, lihat [build](#) dalam dokumentasi Bitbucket API.

Status build yang dipicu oleh webhook selalu dilaporkan ke penyedia sumber. Agar status build yang dimulai dari konsol atau panggilan API dilaporkan ke penyedia sumber, Anda harus memilih setelan ini.

Jika build proyek Anda dipicu oleh webhook, Anda harus mendorong komit baru ke repo agar perubahan pada pengaturan ini diterapkan.

Dalam peristiwa webhook sumber utama, pilih Rebuild setiap kali perubahan kode didorong ke repositori ini jika Anda CodeBuild ingin membangun kode sumber setiap kali perubahan kode didorong ke repositori ini. Untuk informasi selengkapnya tentang webhook dan grup filter, lihat [Acara webhook Bitbucket](#)

## GitHub

### Repositori

Pilih Connect menggunakan OAuth atau Connect dengan token akses GitHub pribadi dan ikuti petunjuk untuk menghubungkan (atau menyambung kembali) ke GitHub dan mengotorisasi akses ke AWS CodeBuild

Pilih repositori publik atau repositori di akun Anda.



## Versi sumber

Masukkan cabang, ID komit, tag, atau referensi dan ID komit. Lihat informasi yang lebih lengkap di [Contoh versi sumber dengan AWS CodeBuild](#)

### Note

Kami menyarankan Anda memilih nama cabang Git yang tidak terlihat seperti ID komit, seperti 811dd1ba1aba14473856cee38308caed7190c0d atau 5392f7. Ini membantu Anda menghindari tabrakan checkout Git dengan commit yang sebenarnya.

## Kedalaman klon Git

Pilih kedalaman klon Git untuk membuat klon dangkal dengan riwayat terpotong ke jumlah komit yang ditentukan. Jika Anda ingin klon penuh, pilih Penuh.

## Submodul Git

Pilih Gunakan submodul Git jika Anda ingin menyertakan submodul Git dalam repositori Anda.

## Membangun status

Pilih Laporkan status build ke penyedia sumber saat build dimulai dan selesai jika ingin status awal dan penyelesaian build dilaporkan ke penyedia sumber.

Untuk dapat melaporkan status build ke penyedia sumber, pengguna yang terkait dengan penyedia sumber harus memiliki akses tulis ke repo. Jika pengguna tidak memiliki akses tulis, status build tidak dapat diperbarui. Untuk informasi selengkapnya, lihat [Akses penyedia sumber](#).

Untuk konteks Status, masukkan nilai yang akan digunakan untuk context parameter dalam status GitHub komit. Untuk informasi selengkapnya, lihat [Membuat status komit](#) di panduan GitHub pengembang.

Untuk URL Target, masukkan nilai yang akan digunakan untuk target\_url parameter dalam status GitHub komit. Untuk informasi selengkapnya, lihat [Membuat status komit](#) di panduan GitHub pengembang.

Status build yang dipicu oleh webhook selalu dilaporkan ke penyedia sumber. Agar status build yang dimulai dari konsol atau panggilan API dilaporkan ke penyedia sumber, Anda harus memilih setelan ini.

Jika build proyek Anda dipicu oleh webhook, Anda harus mendorong komit baru ke repo agar perubahan pada pengaturan ini diterapkan.

Dalam peristiwa webhook sumber utama, pilih Rebuild setiap kali perubahan kode didorong ke repositori ini jika Anda CodeBuild ingin membangun kode sumber setiap kali perubahan kode didorong ke repositori ini. Untuk informasi selengkapnya tentang webhook dan grup filter, lihat [GitHub acara webhook](#)

## GitHub Enterprise Server

### GitHub Token akses pribadi perusahaan

Lihat [GitHub Contoh Server Perusahaan](#) untuk informasi tentang cara menyalin token akses pribadi ke clipboard Anda. Tempel token di bidang teks, lalu pilih Simpan Token.

#### Note

Anda hanya perlu memasukkan dan menyimpan token akses pribadi sekali. CodeBuild menggunakan token ini di semua proyek future.

### Versi sumber

Masukkan permintaan tarik, cabang, ID komit, tag, atau referensi, dan ID komit. Untuk informasi selengkapnya, lihat [Contoh versi sumber dengan AWS CodeBuild](#).

#### Note

Kami menyarankan Anda memilih nama cabang Git yang tidak terlihat seperti ID komit, seperti 811dd1ba1aba14473856cee38308caed7190c0d atau5392f7. Ini membantu Anda menghindari tabrakan checkout Git dengan commit yang sebenarnya.

### Kedalaman klon Git

Pilih kedalaman klon Git untuk membuat klon dangkal dengan riwayat terpotong ke jumlah komit yang ditentukan. Jika Anda ingin klon penuh, pilih Penuh.

### Submodul Git

Pilih Gunakan submodul Git jika Anda ingin menyertakan submodul Git dalam repositori Anda.

## Membangun status

Pilih Laporkan status build ke penyedia sumber saat build dimulai dan selesai jika ingin status awal dan penyelesaian build dilaporkan ke penyedia sumber.

Untuk dapat melaporkan status build ke penyedia sumber, pengguna yang terkait dengan penyedia sumber harus memiliki akses tulis ke repo. Jika pengguna tidak memiliki akses tulis, status build tidak dapat diperbarui. Untuk informasi selengkapnya, lihat [Akses penyedia sumber](#).

Untuk konteks Status, masukkan nilai yang akan digunakan untuk `context` parameter dalam status GitHub komit. Untuk informasi selengkapnya, lihat [Membuat status komit](#) di panduan GitHub pengembang.

Untuk URL Target, masukkan nilai yang akan digunakan untuk `target_url` parameter dalam status GitHub komit. Untuk informasi selengkapnya, lihat [Membuat status komit](#) di panduan GitHub pengembang.

Status build yang dipicu oleh webhook selalu dilaporkan ke penyedia sumber. Agar status build yang dimulai dari konsol atau panggilan API dilaporkan ke penyedia sumber, Anda harus memilih setelan ini.

Jika build proyek Anda dipicu oleh webhook, Anda harus mendorong komit baru ke repo agar perubahan pada pengaturan ini diterapkan.

## SSL tidak aman

Pilih Aktifkan SSL yang tidak aman untuk mengabaikan peringatan SSL saat menghubungkan ke repositori proyek Enterprise Anda GitHub .

Dalam peristiwa webhook sumber utama, pilih Rebuild setiap kali perubahan kode didorong ke repositori ini jika Anda CodeBuild ingin membangun kode sumber setiap kali perubahan kode didorong ke repositori ini. Untuk informasi selengkapnya tentang webhook dan grup filter, lihat [GitHub acara webhook](#)

## Environment

### Model penyedia

Lakukan salah satu dari cara berikut:

- Untuk menggunakan armada sesuai permintaan yang dikelola oleh AWS CodeBuild, pilih On-Demand. Dengan armada sesuai permintaan, CodeBuild menyediakan komputasi untuk build Anda. Mesin-mesin dihancurkan saat build selesai. Armada sesuai permintaan dikelola sepenuhnya, dan mencakup kemampuan penskalaan otomatis untuk menangani lonjakan permintaan.
- Untuk menggunakan armada kapasitas cadangan yang dikelola oleh AWS CodeBuild, pilih Kapasitas cadangan, lalu pilih nama Armada. Dengan armada kapasitas cadangan, Anda mengonfigurasi satu set instans khusus untuk lingkungan build Anda. Mesin ini tetap menganggur, siap untuk memproses build atau pengujian segera dan mengurangi durasi build. Dengan armada kapasitas cadangan, mesin Anda selalu berjalan dan akan terus mengeluarkan biaya selama disediakan.

Untuk informasi, lihat [Bekerja dengan kapasitas cadangan di AWS CodeBuild](#).

### Citra lingkungan

Lakukan salah satu dari cara berikut:

- Untuk menggunakan image Docker yang dikelola oleh AWS CodeBuild, pilih Managed image, lalu buat pilihan dari Sistem operasi, Runtime, Image, dan versi Image. Buat pilihan dari jenis Lingkungan jika tersedia.
- Untuk menggunakan gambar Docker lain, pilih Custom image. Untuk jenis Lingkungan, pilih ARM, Linux, GPU Linux, atau Windows. Jika Anda memilih Registri lain, untuk URL registri eksternal, masukkan nama dan tag gambar Docker di Docker Hub, menggunakan format *docker repository/docker image name*. Jika Anda memilih Amazon ECR, gunakan repositori Amazon ECR dan gambar Amazon ECR untuk memilih gambar Docker di akun Anda. AWS
- Untuk menggunakan image Docker pribadi, pilih Custom image. Untuk jenis Lingkungan, pilih ARM, Linux, GPU Linux, atau Windows. Untuk registri Gambar, pilih Registri lain, lalu masukkan ARN kredensial untuk gambar Docker pribadi Anda. Kredensialnya harus dibuat oleh Secrets Manager. Untuk informasi lebih lanjut, lihat [Apa itu AWS Secrets Manager?](#) di Panduan Pengguna AWS Secrets Manager.

#### Note

CodeBuild mengganti ENTRYPOINT untuk gambar Docker khusus.

## Hitung

Lakukan salah satu dari cara berikut:

- Untuk menggunakan komputasi EC2, pilih EC2. Komputasi EC2 menawarkan fleksibilitas yang dioptimalkan selama aksi berjalan.
- Untuk menggunakan komputasi Lambda, pilih Lambda. Lambda compute menawarkan kecepatan start-up yang dioptimalkan untuk build Anda. Lambda mendukung build yang lebih cepat karena latensi start-up yang lebih rendah. Lambda juga secara otomatis menskalakan, jadi build tidak menunggu dalam antrian untuk dijalankan. Untuk informasi, lihat [Bekerja dengan AWS Lambda komputasi di AWS CodeBuild](#).

## Peran layanan

Lakukan salah satu dari cara berikut:

- Jika Anda tidak memiliki peran CodeBuild layanan, pilih Peran layanan baru. Di Nama peran, masukkan nama untuk peran baru.
- Jika Anda memiliki peran CodeBuild layanan, pilih Peran layanan yang ada. Di Peran ARN, pilih peran layanan.

### Note

Saat menggunakan konsol untuk membuat proyek build, Anda dapat membuat peran CodeBuild layanan secara bersamaan. Secara default, peran hanya berfungsi dengan proyek build tersebut. Jika Anda menggunakan konsol untuk mengaitkan peran layanan ini dengan proyek build lain, peran akan diperbarui agar berfungsi dengan proyek build lainnya. Peran layanan dapat bekerja dengan hingga 10 proyek pembangunan.

## Konfigurasi tambahan

### Batas waktu

Tentukan nilai, antara 5 menit dan 8 jam, setelah itu CodeBuild menghentikan build jika tidak selesai. Jika jam dan menit dibiarkan kosong, nilai default 60 menit digunakan.

### Diistimewakan

(Opsional) Pilih Aktifkan bendera ini jika Anda ingin membuat gambar Docker atau ingin build Anda mendapatkan hak istimewa yang ditingkatkan hanya jika Anda berencana menggunakan

proyek build ini untuk membuat gambar Docker. Jika tidak, semua build terkait yang mencoba berinteraksi dengan daemon Docker gagal. Anda juga harus memulai daemon Docker sehingga build Anda dapat berinteraksi dengannya. Salah satu cara untuk melakukannya adalah dengan menginisialisasi daemon Docker di `install` fase spesifikasi build Anda dengan menjalankan perintah build berikut. Jangan jalankan perintah ini jika Anda memilih image lingkungan build yang disediakan oleh CodeBuild dukungan Docker.

#### Note

Secara default, daemon Docker diaktifkan untuk build non-VPC. Jika Anda ingin menggunakan kontainer Docker untuk build VPC, [lihat Runtime Privilege dan Kemampuan Linux di situs web Docker Docs dan](#) aktifkan mode istimewa. Juga, Windows tidak mendukung mode istimewa.

```
- nohup /usr/local/bin/dockerd --host=unix:///var/run/docker.sock --
host=tcp://127.0.0.1:2375 --storage-driver=overlay2 &
- timeout 15 sh -c "until docker info; do echo .; sleep 1; done"
```

## VPC

Jika Anda CodeBuild ingin bekerja dengan VPC Anda:

- Untuk VPC, pilih ID VPC yang digunakan. CodeBuild
- Untuk Subnet VPC, pilih subnet yang menyertakan sumber daya yang digunakan. CodeBuild
- Untuk grup Keamanan VPC, pilih grup keamanan yang CodeBuild digunakan untuk memungkinkan akses ke sumber daya di VPC.


Untuk informasi selengkapnya, lihat [Gunakan AWS CodeBuild dengan Amazon Virtual Private Cloud](#).

## Hitung

Pilih salah satu opsi yang tersedia.

## Variabel lingkungan

Masukkan nama dan nilai, lalu pilih jenis setiap variabel lingkungan untuk build yang akan digunakan.

 Note


CodeBuild menetapkan variabel lingkungan untuk AWS Wilayah Anda secara otomatis. Anda harus menyetel variabel lingkungan berikut jika Anda belum menambahkannya ke `buildspec.yml`:

- `AWS_ACCOUNT_ID`
- `IMAGE_REPO_NAME`
- `IMAGE_TAG`

Konsol dan AWS CLI pengguna dapat melihat variabel lingkungan. Jika Anda tidak memiliki kekhawatiran tentang visibilitas variabel lingkungan Anda, atur bidang Nama dan Nilai, lalu setel Type ke Plaintext.

Kami menyarankan Anda menyimpan variabel lingkungan dengan nilai sensitif, seperti ID kunci AWS akses, kunci akses AWS rahasia, atau kata sandi sebagai parameter di Amazon EC2 Systems Manager Parameter Store atau AWS Secrets Manager.

Jika Anda menggunakan Amazon EC2 Systems Manager Parameter Store, lalu untuk Type, pilih Parameter. Untuk Nama, masukkan pengenal CodeBuild untuk referensi. Untuk Nilai, masukkan nama parameter seperti yang disimpan di Amazon EC2 Systems Manager Parameter Store. Menggunakan parameter bernama `/CodeBuild/dockerLoginPassword` sebagai contoh, untuk Type, pilih Parameter. Untuk Nama, masukkan `LOGIN_PASSWORD`. Untuk Nilai, masukkan `/CodeBuild/dockerLoginPassword`.

 Important

Jika Anda menggunakan Amazon EC2 Systems Manager Parameter Store, sebaiknya Anda menyimpan parameter dengan nama parameter yang dimulai `/CodeBuild/` dengan (misalnya `/CodeBuild/dockerLoginPassword`). Anda dapat menggunakan CodeBuild konsol untuk membuat parameter di Amazon EC2 Systems Manager. Pilih Buat parameter, lalu ikuti instruksi di kotak dialog. (Di kotak dialog itu, untuk kunci KMS, Anda dapat menentukan ARN kunci di AWS KMS akun Anda. Amazon EC2 Systems Manager menggunakan kunci ini untuk mengenkripsi nilai parameter selama penyimpanan dan mendekripsi selama pengambilan.) Jika Anda menggunakan CodeBuild konsol untuk membuat parameter, konsol memulai nama parameter dengan `/CodeBuild/` saat sedang disimpan. Untuk informasi

selengkapnya, lihat [Systems Manager Parameter Store](#) dan [Systems Manager Parameter Store Console Walkthrough](#) di Panduan Pengguna Amazon EC2 Systems Manager.

Jika project build Anda mengacu pada parameter yang disimpan di Amazon EC2 Systems Manager Parameter Store, peran layanan project build harus mengizinkan tindakan `ssm:GetParameters` tersebut. Jika Anda memilih Peran layanan baru lebih awal, CodeBuild sertakan tindakan ini dalam peran layanan default untuk proyek build Anda. Namun, jika Anda memilih Peran layanan yang ada, Anda harus menyertakan tindakan ini ke peran layanan secara terpisah.

Jika project build Anda merujuk ke parameter yang disimpan di Amazon EC2 Systems Manager Parameter Store dengan nama parameter yang tidak dimulai/`CodeBuild/`, dan Anda memilih Peran layanan baru, Anda harus memperbarui peran layanan tersebut untuk mengizinkan akses ke nama parameter yang tidak dimulai/`CodeBuild/`. Ini karena peran layanan itu memungkinkan akses hanya ke nama parameter yang dimulai dengan/`CodeBuild/`.

Jika Anda memilih Peran layanan baru, peran layanan menyertakan izin untuk mendekripsi semua parameter di bawah `/CodeBuild/` namespace di Amazon EC2 Systems Manager Parameter Store.

Variabel lingkungan yang Anda tetapkan menggantikan variabel lingkungan yang ada. Misalnya, jika image Docker sudah berisi variabel lingkungan bernama `MY_VAR` dengan nilai `my_value`, dan Anda menetapkan variabel lingkungan bernama `MY_VAR` dengan nilai `other_value`, `my_value` maka diganti dengan `other_value`. Demikian pula, jika image Docker sudah berisi variabel lingkungan bernama `PATH` dengan nilai `/usr/local/sbin:/usr/local/bin`, dan Anda menetapkan variabel lingkungan bernama `PATH` dengan nilai `$PATH:/usr/share/ant/bin:/usr/local/sbin:/usr/local/bin` maka digantikan oleh nilai `$PATH:/usr/share/ant/bin` literal. Jangan mengatur variabel lingkungan apa pun dengan nama yang dimulai dengan `CODEBUILD_`. Awalan ini dicadangkan untuk penggunaan internal.

Jika variabel lingkungan dengan nama yang sama didefinisikan di beberapa tempat, nilainya ditentukan sebagai berikut:

- Nilai dalam panggilan operasi start build lebih diutamakan.
- Nilai dalam definisi proyek build akan diutamakan berikutnya.
- Nilai dalam deklarasi `buildspec` diutamakan paling rendah.



Jika Anda menggunakan Secrets Manager, untuk Type, pilih Secrets Manager. Untuk Nama, masukkan pengenalan CodeBuild untuk referensi. Untuk Nilai, masukkan `reference-key` menggunakan pola `secret-id:json-key:version-stage:version-id`. Untuk informasi, lihat [Secrets Manager reference-key in the buildspec file](#).

#### Important

Jika Anda menggunakan Secrets Manager, kami sarankan Anda menyimpan rahasia dengan nama yang dimulai dengan `/CodeBuild/` (misalnya, `/CodeBuild/dockerLoginPassword`). Untuk informasi lebih lanjut, lihat [Apa itu AWS Secrets Manager?](#) di Panduan Pengguna AWS Secrets Manager.

Jika proyek build Anda mengacu pada rahasia yang disimpan di Secrets Manager, peran layanan proyek build harus mengizinkan `secretsmanager:GetSecretValue` tindakan tersebut. Jika Anda memilih Peran layanan baru lebih awal, CodeBuild sertakan tindakan ini dalam peran layanan default untuk proyek build Anda. Namun, jika Anda memilih Peran layanan yang ada, Anda harus menyertakan tindakan ini ke peran layanan secara terpisah.

Jika proyek build mengacu pada rahasia yang disimpan di Secrets Manager dengan nama rahasia yang tidak dimulai `/CodeBuild/`, dan Anda memilih Peran layanan baru, Anda harus memperbarui peran layanan untuk mengizinkan akses ke nama rahasia yang tidak dimulai `/CodeBuild/`. Ini karena peran layanan memungkinkan akses hanya ke nama rahasia yang dimulai dengan `/CodeBuild/`.

Jika Anda memilih Peran layanan baru, peran layanan menyertakan izin untuk mendekripsi semua rahasia di bawah `/CodeBuild/` namespace di Secrets Manager.

## Buildspec

### Membangun spesifikasi

Lakukan salah satu dari cara berikut:

- Jika kode sumber Anda menyertakan file buildspec, pilih Gunakan file buildspec. Secara default, CodeBuild mencari file bernama `buildspec.yml` di direktori root kode sumber. Jika file buildspec Anda menggunakan nama atau lokasi yang berbeda, masukkan jalurnya dari root sumber dalam nama Buildspec (misalnya, `atau. buildspec-two.yml configuration/buildspec.yml`). Jika file buildspec ada di bucket S3, file tersebut harus berada di AWS

Region yang sama dengan project build Anda. Tentukan file buildspec menggunakan ARN (misalnya,). `arn:aws:s3:::<my-codebuild-sample2>/buildspec.yml`

- Jika kode sumber Anda tidak menyertakan file buildspec, atau jika Anda ingin menjalankan perintah build yang berbeda dari yang ditentukan untuk build fase dalam buildspec.yml file di direktori root kode sumber, pilih Sisipkan perintah build. Untuk perintah Build, masukkan perintah yang ingin Anda jalankan dalam build fase. Untuk beberapa perintah, pisahkan setiap perintah dengan && (misalnya, `mvn test && mvn package`). Untuk menjalankan perintah di fase lain, atau jika Anda memiliki daftar panjang perintah untuk build fase tersebut, tambahkan buildspec.yml file ke direktori root kode sumber, tambahkan perintah ke file, lalu pilih Gunakan buildspec.yml di direktori root kode sumber.

Lihat informasi yang lebih lengkap di [Referensi Buildspec](#).

## Konfigurasi Batch

Anda dapat menjalankan grup build sebagai satu operasi. Untuk informasi selengkapnya, lihat [Batch dibangun di AWS CodeBuild](#).

### Tentukan konfigurasi batch

Pilih untuk mengizinkan pembuatan batch dalam proyek ini.

### Peran layanan Batch

Menyediakan peran layanan untuk build batch.

Pilih salah satu cara berikut:

- Jika Anda tidak memiliki peran layanan batch, pilih Peran layanan baru. Di peran Layanan, masukkan nama untuk peran baru.
- Jika Anda memiliki peran layanan batch, pilih Peran layanan yang ada. Dalam peran Layanan, pilih peran layanan.

Batch build memperkenalkan peran keamanan baru dalam konfigurasi batch. Peran baru ini diperlukan karena CodeBuild harus dapat memanggil `StartBuild`, `StopBuild`, dan `RetryBuild` tindakan atas nama Anda untuk menjalankan build sebagai bagian dari batch. Pelanggan harus menggunakan peran baru, dan bukan peran yang sama yang mereka gunakan dalam build mereka, karena dua alasan:

- Memberikan peran `StartBuild`, `StopBuild`, dan `RetryBuild` izin akan memungkinkan satu build untuk memulai lebih banyak build melalui buildspec.

- CodeBuild build batch memberikan batasan yang membatasi jumlah build dan tipe komputasi yang dapat digunakan untuk build dalam batch. Jika peran build memiliki izin ini, kemungkinan build itu sendiri dapat melewati batasan ini.

### Jenis komputasi yang diizinkan untuk batch

Pilih jenis komputasi yang diizinkan untuk batch. Pilih semua yang berlaku.

### Build maksimum yang diizinkan dalam batch

Masukkan jumlah maksimum build yang diizinkan dalam batch. Jika batch melebihi batas ini, batch akan gagal.

### Batas waktu Batch

Masukkan jumlah waktu maksimum untuk menyelesaikan batch build.

### Gabungkan artefak

Pilih Gabungkan semua artefak dari batch ke satu lokasi agar semua artefak dari batch digabungkan menjadi satu lokasi.

### Mode laporan Batch

Pilih mode laporan status build yang diinginkan untuk build batch.

#### Note

Bidang ini hanya tersedia jika sumber proyek adalah Bitbucket, GitHub, atau GitHub Enterprise, dan Laporkan status build ke penyedia sumber saat build Anda mulai dan selesai dipilih di bawah Sumber.

### Build agregat

Pilih agar status untuk semua build dalam batch digabungkan menjadi satu laporan status.

### Membangun individu

Pilih agar status build untuk semua build dalam batch dilaporkan secara terpisah.

## Artifacts

### Jenis

Lakukan salah satu dari cara berikut:

- Jika Anda tidak ingin membuat artefak keluaran build, pilih Tidak ada artefak. Anda mungkin ingin melakukan ini jika Anda hanya menjalankan pengujian build atau Anda ingin mendorong image Docker ke repositori Amazon ECR.
- Untuk menyimpan output build dalam bucket S3, pilih Amazon S3, lalu lakukan hal berikut:
  - Jika Anda ingin menggunakan nama proyek Anda untuk file atau folder ZIP keluaran build, biarkan Nama kosong. Jika tidak, masukkan nama. (Jika Anda ingin mengeluarkan file ZIP, dan Anda ingin file ZIP memiliki ekstensi file, pastikan untuk memasukkannya setelah nama file ZIP.)
  - Pilih Aktifkan versi semantik jika Anda ingin nama yang ditentukan dalam file buildspec untuk mengganti nama apa pun yang ditentukan di konsol. Nama dalam file buildspec dihitung pada waktu pembuatan dan menggunakan bahasa perintah Shell. Misalnya, Anda dapat menambahkan tanggal dan waktu untuk nama artefak Anda sehingga selalu unik. Nama artefak unik mencegah artefak ditimpa. Untuk informasi selengkapnya, lihat [Sintaks Buildspec](#).
  - Untuk nama Bucket, pilih nama bucket keluaran.
  - Jika Anda memilih Insert build perintah sebelumnya dalam prosedur ini, maka untuk file Output, masukkan lokasi file dari build yang ingin Anda masukkan ke dalam file atau folder ZIP keluaran build. Untuk beberapa lokasi, pisahkan setiap lokasi dengan koma (misalnya, `appspec.yml, target/my-app.jar`). Untuk informasi lebih lanjut, lihat deskripsi files di [Sintaks Buildspec](#).
  - Jika Anda tidak ingin artefak build Anda dienkripsi, pilih Hapus enkripsi artefak.

Untuk setiap set artefak sekunder yang Anda inginkan:

1. Untuk pengenalan Artifact, masukkan nilai yang kurang dari 128 karakter dan hanya berisi karakter alfanumerik dan garis bawah.
2. Pilih Tambahkan artefak.
3. Ikuti langkah-langkah sebelumnya untuk mengonfigurasi artefak sekunder Anda.
4. Pilih Simpan artefak.

## Konfigurasi tambahan

### Kunci enkripsi

(Opsional) Lakukan salah satu langkah berikut:

- Untuk menggunakan Amazon S3 di akun Anda untuk mengenkripsi artefak keluaran build, biarkan kunci Enkripsi kosong. Kunci yang dikelola AWS Ini menjadi opsi default.
- Untuk menggunakan kunci yang dikelola pelanggan untuk mengenkripsi artefak keluaran build, di kunci Enkripsi, masukkan ARN kunci KMS. Gunakan format `arn:aws:kms:region-ID:account-ID:key/key-ID`.

## Jenis cache

Untuk jenis Cache, pilih salah satu dari berikut ini:

- Jika Anda tidak ingin menggunakan cache, pilih Tidak ada cache.
- Jika Anda ingin menggunakan cache Amazon S3, pilih Amazon S3, lalu lakukan hal berikut:
  - Untuk Bucket, pilih nama bucket S3 tempat cache disimpan.
  - (Opsional) Untuk awalan jalur Cache, masukkan awalan jalur Amazon S3. Nilai awalan jalur Cache mirip dengan nama direktori. Itu memungkinkan Anda untuk menyimpan cache di bawah direktori yang sama dalam ember.

### Important

Jangan menambahkan garis miring (/) ke ujung awalan jalur.

- Jika Anda ingin menggunakan cache lokal, pilih Lokal, lalu pilih satu atau beberapa mode cache lokal.

### Note

Mode cache lapisan Docker hanya tersedia untuk Linux. Jika Anda memilikinya, proyek Anda harus berjalan dalam mode istimewa.

Menggunakan cache menghemat banyak waktu pembuatan karena bagian lingkungan build yang dapat digunakan kembali disimpan dalam cache dan digunakan di seluruh build. Untuk informasi tentang menentukan cache dalam file buildspec, lihat [Sintaks Buildspec](#) Untuk informasi lebih lanjut tentang caching, lihat [Membangun caching di AWS CodeBuild](#).

## Log

Pilih log yang ingin Anda buat. Anda dapat membuat CloudWatch Log Amazon, log Amazon S3, atau keduanya.

## CloudWatch

Jika Anda ingin CloudWatch log Amazon Log:

CloudWatch log

Pilih CloudWatch log.

Nama grup

Masukkan nama grup CloudWatch log Amazon Logs Anda.

Nama aliran

Masukkan nama aliran CloudWatch log Amazon Logs Anda.

## S3

Jika Anda ingin log Amazon S3:

Log S3

Pilih log S3.

Bucket

Pilih nama bucket S3 untuk log Anda.

Awalan jalur

Masukkan awalan untuk log Anda.

Nonaktifkan enkripsi log S3

Pilih jika Anda tidak ingin log S3 Anda dienkripsi.

## Buat proyek build (AWS CLI)

Untuk informasi lebih lanjut tentang menggunakan AWS CLI with CodeBuild, lihat [Referensi baris perintah](#).

Untuk membuat proyek CodeBuild build menggunakan AWS CLI, Anda membuat struktur [Proyek](#) berformat JSON, mengisi struktur, dan memanggil [create-project](#) perintah untuk membuat proyek.

## Buat file JSON

Buat file JSON kerangka dengan [create-project](#) perintah, menggunakan opsi: `--generate-cli-skeleton`

```
aws codebuild create-project --generate-cli-skeleton > <json-file>
```

<json-file>Ini membuat file JSON dengan path dan nama file yang ditentukan oleh.

## Isi file JSON

Ubah data JSON sebagai berikut dan simpan hasil Anda.

```
{
 "name": "<project-name>",
 "description": "<description>",
 "source": {
 "type": "CODECOMMIT" | "CODEPIPELINE" | "GITHUB" | "GITHUB_ENTERPRISE" |
"BITBUCKET" | "S3" | "NO_SOURCE",
 "location": "<source-location>",
 "gitCloneDepth": "<git-clone-depth>",
 "buildspec": "<buildspec>",
 "InsecureSsl": "<insecure-ssl>",
 "reportBuildStatus": "<report-build-status>",
 "buildStatusConfig": {
 "context": "<context>",
 "targetUrl": "<target-url>"
 },
 "gitSubmodulesConfig": {
 "fetchSubmodules": "<fetch-submodules>"
 },
 "auth": {
 "type": "<auth-type>",
 "resource": "<auth-resource>"
 },
 "sourceIdentifier": "<source-identifier>"
 },
 "secondarySources": [
 {
 "type": "CODECOMMIT" | "CODEPIPELINE" | "GITHUB" | "GITHUB_ENTERPRISE" |
"BITBUCKET" | "S3" | "NO_SOURCE",
 "location": "<source-location>",
 "gitCloneDepth": "<git-clone-depth>",
```

```
 "buildspec": "<buildspec>",
 "InsecureSsl": "<insecure-ssl>",
 "reportBuildStatus": "<report-build-status>",
 "auth": {
 "type": "<auth-type>",
 "resource": "<auth-resource>"
 },
 "sourceIdentifier": "<source-identifier>"
 }
],
"secondarySourceVersions": [
 {
 "sourceIdentifier": "<secondary-source-identifier>",
 "sourceVersion": "<secondary-source-version>"
 }
],
"sourceVersion": "<source-version>",
"artifacts": {
 "type": "CODEPIPELINE" | "S3" | "NO_ARTIFACTS",
 "location": "<artifacts-location>",
 "path": "<artifacts-path>",
 "namespaceType": "<artifacts-namespacetype>",
 "name": "<artifacts-name>",
 "overrideArtifactName": "<override-artifact-name>",
 "packaging": "<artifacts-packaging>"
},
"secondaryArtifacts": [
 {
 "type": "CODEPIPELINE" | "S3" | "NO_ARTIFACTS",
 "location": "<secondary-artifact-location>",
 "path": "<secondary-artifact-path>",
 "namespaceType": "<secondary-artifact-namespaceType>",
 "name": "<secondary-artifact-name>",
 "packaging": "<secondary-artifact-packaging>",
 "artifactIdentifier": "<secondary-artifact-identifier>"
 }
],
"cache": {
 "type": "<cache-type>",
 "location": "<cache-location>",
 "mode": [
 "<cache-mode>"
]
},
```



```

"environment": {
 "type": "LINUX_CONTAINER" | "LINUX_GPU_CONTAINER" | "ARM_CONTAINER" |
"WINDOWS_SERVER_2019_CONTAINER",
 "image": "<image>",
 "computeType": "BUILD_GENERAL1_SMALL" | "BUILD_GENERAL1_MEDIUM" |
"BUILD_GENERAL1_LARGE" | "BUILD_GENERAL1_2XLARGE",
 "certificate": "<certificate>",
 "environmentVariables": [
 {
 "name": "<environmentVariable-name>",
 "value": "<environmentVariable-value>",
 "type": "<environmentVariable-type>"
 }
],
 "registryCredential": [
 {
 "credential": "<credential-arn-or-name>",
 "credentialProvider": "<credential-provider>"
 }
],
 "imagePullCredentialsType": "CODEBUILD" | "SERVICE_ROLE",
 "privilegedMode": "<privileged-mode>"
},
"serviceRole": "<service-role>",
"timeoutInMinutes": <timeout>,
"queuedTimeoutInMinutes": <queued-timeout>,
"encryptionKey": "<encryption-key>",
"tags": [
 {
 "key": "<tag-key>",
 "value": "<tag-value>"
 }
],
"vpcConfig": {
 "securityGroupIds": [
 "<security-group-id>"
],
 "subnets": [
 "<subnet-id>"
],
 "vpcId": "<vpc-id>"
},
"badgeEnabled": "<badge-enabled>",
"logsConfig": {

```

```

"cloudWatchLogs": {
 "status": "<cloudwatch-logs-status>",
 "groupName": "<group-name>",
 "streamName": "<stream-name>"
},
"s3Logs": {
 "status": "<s3-logs-status>",
 "location": "<s3-logs-location>",
 "encryptionDisabled": "<s3-logs-encryption-disabled>"
}
},
"fileSystemLocations": [
{
 "type": "EFS",
 "location": "<EFS-DNS-name-1>:/<directory-path>",
 "mountPoint": "<mount-point>",
 "identifier": "<efs-identifier>",
 "mountOptions": "<efs-mount-options>"
}
],
"buildBatchConfig": {
 "serviceRole": "<batch-service-role>",
 "combineArtifacts": <combine-artifacts>,
 "restrictions": {
 "maximumBuildsAllowed": <max-builds>,
 "computeTypesAllowed": [
 "<compute-type>"
]
 }
},
"timeoutInMins": <batch-timeout>,
"batchReportMode": "REPORT_AGGREGATED_BATCH" | "REPORT_INDIVIDUAL_BUILDS"
},
"concurrentBuildLimit": <concurrent-build-limit>
}

```

Ganti yang berikut ini:

name

Wajib. Nama untuk proyek pembangunan ini. Nama ini harus unik di semua proyek build di AWS akun Anda.

## deskripsi

Opsional. Deskripsi untuk proyek pembangunan ini.

## sumber

Wajib. [ProjectSource](#) Objek yang berisi informasi tentang pengaturan kode sumber proyek build ini. Setelah Anda menambahkan source objek, Anda dapat menambahkan hingga 12 sumber lagi menggunakan file. Pengaturan ini meliputi yang berikut:

## sumber/jenis

Wajib. Jenis repositori yang berisi kode sumber untuk membangun. Nilai yang valid meliputi:

- CODECOMMIT
- CODEPIPELINE
- GITHUB
- GITHUB\_ENTERPRISE
- BITBUCKET
- S3
- NO\_SOURCE

Jika Anda menggunakan NO\_SOURCE, buildspec tidak dapat berupa file karena proyek tidak memiliki sumber. Sebagai gantinya, Anda harus menggunakan buildspec atribut untuk menentukan string berformat YAML untuk buildspec Anda. Untuk informasi selengkapnya, lihat [Proyek tanpa sampel sumber](#).

## sumber/lokasi

Diperlukan kecuali Anda menyetel <source-type> ke CODEPIPELINE. Lokasi kode sumber untuk jenis repositori yang ditentukan.

- Untuk CodeCommit, URL kloning HTTPS ke repositori yang berisi kode sumber dan file buildspec (misalnya,). `https://git-codecommit.<region-id>.amazonaws.com/v1/repos/<repo-name>`
- Untuk Amazon S3, nama bucket input build, diikuti dengan path dan nama file ZIP yang berisi kode sumber dan buildspec. Sebagai contoh:
  - Untuk file ZIP yang terletak di root bucket input: `<bucket-name>/<object-name>.zip`.
  - Untuk file ZIP yang terletak di subfolder di bucket input: `<bucket-name>/<subfolder-path>/<object-name>.zip`.

- Untuk GitHub, URL kloning HTTPS ke repositori yang berisi kode sumber dan file buildspec. URL harus berisi github.com. Anda harus menghubungkan AWS akun Anda ke GitHub akun Anda. Untuk melakukan ini, gunakan CodeBuild konsol untuk membuat proyek build.
  1. Pada halaman GitHub Otorisasi aplikasi, di bagian Akses Organisasi, pilih Minta akses di sebelah setiap repositori yang CodeBuild ingin Anda akses di.
  2. Pilih Otorisasi aplikasi. (Setelah Anda terhubung ke GitHub akun Anda, Anda tidak perlu menyelesaikan pembuatan proyek build. Anda dapat menutup CodeBuild konsol.)
- Untuk GitHub Enterprise Server, HTTP atau HTTPS mengkloning URL ke repositori yang berisi kode sumber dan file buildspec. Anda juga harus menghubungkan AWS akun Anda ke akun GitHub Enterprise Server Anda. Untuk melakukan ini, gunakan CodeBuild konsol untuk membuat proyek build.
  1. Buat token akses pribadi di Server GitHub Perusahaan.
  2. Salin token ini ke clipboard Anda sehingga Anda dapat menggunakannya saat membuat proyek. CodeBuild Untuk informasi selengkapnya, lihat [Membuat token akses pribadi untuk baris perintah](#) di situs web GitHub Bantuan.
  3. Saat Anda menggunakan konsol untuk membuat CodeBuild proyek Anda, di Sumber, untuk penyedia Sumber, pilih GitHubEnterprise.
  4. Untuk Token Akses Pribadi, tempel token yang disalin ke clipboard Anda. Pilih Simpan Token. CodeBuild Akun Anda sekarang terhubung ke akun GitHub Enterprise Server Anda.
- Untuk Bitbucket, URL kloning HTTPS ke repositori yang berisi kode sumber dan file buildspec. URL harus berisi bitbucket.org. Anda juga harus menghubungkan AWS akun Anda ke akun Bitbucket Anda. Untuk melakukan ini, gunakan CodeBuild konsol untuk membuat proyek build.
  1. Ketika Anda menggunakan konsol untuk menghubungkan (atau menghubungkan kembali) dengan Bitbucket, pada halaman Mengonfirmasi akses ke akun Anda di Bitbucket, pilih Berikan akses. (Setelah Anda terhubung ke akun Bitbucket Anda, Anda tidak perlu menyelesaikan pembuatan proyek build. Anda dapat menutup CodeBuild konsol.)
- Untuk AWS CodePipeline, jangan tentukan `location` nilai untuk `source`. CodePipeline mengabaikan nilai ini karena saat Anda membuat pipeline CodePipeline, Anda menentukan lokasi kode sumber di tahap Sumber pipeline.

sumber/ gitCloneDepth

Opsional. Kedalaman riwayat yang akan diunduh. Nilai minimum adalah 0. Jika nilai ini adalah 0, lebih besar dari 25, atau tidak disediakan, maka riwayat penuh diunduh dengan setiap proyek build. Jika jenis sumber Anda adalah Amazon S3, nilai ini tidak didukung.

## sumber/buildspec

Opsional. Definisi spesifikasi build atau file yang akan digunakan. Jika nilai ini tidak disediakan atau diatur ke string kosong, kode sumber harus berisi `buildspec.yml` file dalam direktori root nya. Jika nilai ini disetel, itu bisa berupa definisi buildspec sebaris, jalur ke file buildspec alternatif relatif terhadap direktori root sumber utama Anda, atau jalur ke bucket S3. Bucket harus berada di AWS Wilayah yang sama dengan proyek pembangunan. Tentukan file buildspec menggunakan ARN (misalnya,). `arn:aws:s3:::<my-codebuild-sample2>/buildspec.yml` Untuk informasi selengkapnya, lihat [Nama file Buildspec dan lokasi penyimpanan](#).

## sumber/autentikasi

Jangan gunakan `.` Objek ini hanya digunakan oleh CodeBuild konsol.

## sumber/ reportBuildStatus

Menentukan apakah akan mengirim status awal dan penyelesaian build kepada penyedia sumber Anda. Jika Anda mengatur ini dengan penyedia sumber selain GitHub, GitHub Enterprise Server, atau Bitbucket, akan `invalidInputException` dilemparkan.

Untuk dapat melaporkan status build ke penyedia sumber, pengguna yang terkait dengan penyedia sumber harus memiliki akses tulis ke repo. Jika pengguna tidak memiliki akses tulis, status build tidak dapat diperbarui. Untuk informasi selengkapnya, lihat [Akses penyedia sumber](#).

## sumber/ buildStatusConfig

Berisi informasi yang menentukan cara proyek CodeBuild build melaporkan status build ke penyedia sumber. Opsi ini hanya digunakan ketika jenis sumbernya `GITHUB`, `GITHUB_ENTERPRISE`, atau `BITBUCKET`.

## sumber//konteks buildStatusConfig

Untuk sumber Bitbucket, parameter ini digunakan untuk name parameter dalam status komit Bitbucket. Untuk GitHub sumber, parameter ini digunakan untuk context parameter dalam status GitHub komit.

Misalnya, Anda dapat memiliki context berisi nomor build dan pemicu webhook menggunakan variabel CodeBuild lingkungan:

```
AWS CodeBuild sample-project Build #${CODEBUILD_BUILD_NUMBER} -
 ${CODEBUILD_WEBHOOK_TRIGGER}
```

Ini menghasilkan konteks yang muncul seperti ini untuk build #24 yang dipicu oleh peristiwa permintaan tarik webhook:

```
AWS CodeBuild sample-project Build #24 - pr/8
```

sumber//TargetUrl buildStatusConfig

Untuk sumber Bitbucket, parameter ini digunakan untuk `url` parameter dalam status komit Bitbucket. Untuk GitHub sumber, parameter ini digunakan untuk `target_url` parameter dalam status GitHub komit.

Misalnya, Anda dapat mengatur `targetUrl` ke `https://aws.amazon.com/codebuild/<path to build>` dan status komit akan ditautkan ke URL ini.

Anda juga dapat menyertakan variabel CodeBuild lingkungan `targetUrl` untuk menambahkan informasi tambahan ke URL. Misalnya, untuk menambahkan wilayah build ke URL, setel `targetUrl` ke:

```
"targetUrl": "https://aws.amazon.com/codebuild/<path to build>?region=$AWS_REGION"
```

Jika wilayah build adalah `us-east-2`, ini akan berkembang menjadi:

```
https://aws.amazon.com/codebuild/<path to build>?region=us-east-2
```

sumber/ gitSubmodulesConfig

Opsional. Informasi tentang konfigurasi submodul Git. Digunakan dengan CodeCommit, GitHub, Server GitHub Perusahaan, dan Bitbucket saja.

sumber//fetchSubmodules gitSubmodulesConfig

Setel `fetchSubmodules` ke `true` jika Anda ingin menyertakan submodul Git dalam repositori Anda. Submodul Git yang disertakan harus dikonfigurasi sebagai HTTPS.

sumber/ InsecureSsl

Opsional. Digunakan dengan GitHub Enterprise Server saja. Tetapkan nilai ini `true` untuk mengabaikan peringatan TLS saat menghubungkan ke repositori proyek GitHub Enterprise Server

Anda. Nilai default adalah `false`. `InsecureSsl` harus digunakan hanya untuk tujuan pengujian. Ini tidak boleh digunakan di lingkungan produksi.

#### `source/SourceIdentifier`

Pengidentifikasi yang ditentukan pengguna untuk sumber proyek. Opsional untuk sumber utama. Diperlukan untuk sumber sekunder.

#### `SecondarySources`

Opsional. Array [ProjectSource](#) objek yang berisi informasi tentang sumber sekunder untuk proyek build. Anda dapat menambahkan hingga 12 sumber sekunder. `secondarySources` objek menggunakan properti yang sama yang digunakan oleh objek. Dalam objek sumber sekunder, `sourceIdentifier` diperlukan.

#### `secondarySourceVersions`

Opsional. Susunan objek [ProjectSourceVersion](#). Jika `secondarySourceVersions` ditentukan pada level build, maka mereka lebih diutamakan daripada ini.

#### `SourceVersion`

Opsional. Versi input build yang akan dibangun untuk proyek ini. Jika tidak ditentukan, versi terbaru digunakan. Jika ditentukan, itu harus berupa salah satu dari:

- Untuk `CodeCommit`, ID komit, cabang, atau tag Git yang akan digunakan.
- Untuk `GitHub`, ID komit, ID permintaan tarik, nama cabang, atau nama tag yang sesuai dengan versi kode sumber yang ingin Anda buat. Jika ID permintaan tarik ditentukan, itu harus menggunakan format `pr/pull-request-ID` (misalnya `pr/25`). Jika nama cabang ditentukan, ID melakukan HEAD cabang digunakan. Jika tidak ditentukan, ID melakukan HEAD cabang default digunakan.
- Untuk `Bitbucket`, ID komit, nama cabang, atau nama tag yang sesuai dengan versi kode sumber yang ingin Anda buat. Jika nama cabang ditentukan, ID melakukan HEAD cabang digunakan. Jika tidak ditentukan, ID melakukan HEAD cabang default digunakan.
- Untuk `Amazon S3`, ID versi objek yang mewakili file ZIP input build yang akan digunakan.

Jika `sourceVersion` ditentukan pada tingkat pembuatan, maka versi tersebut akan mengambil alih `sourceVersion` ini (di tingkat proyek). Untuk informasi selengkapnya, lihat [Contoh versi sumber dengan AWS CodeBuild](#).

## artefak

Wajib. [ProjectArtifacts](#) Objek yang berisi informasi tentang pengaturan artefak keluaran proyek build ini. Setelah Anda menambahkan `artifacts` objek, Anda dapat menambahkan hingga 12 artefak lagi menggunakan. Pengaturan ini meliputi yang berikut:

### artefak/jenis

Wajib. Jenis artefak output build. Nilai yang valid adalah:

- `CODEPIPELINE`
- `NO_ARTIFACTS`
- `S3`

### artefak/lokasi

Hanya digunakan dengan tipe `S3` artefak. Tidak digunakan untuk jenis artefak lainnya.

Nama bucket keluaran yang Anda buat atau identifikasi dalam prasyarat.

### artefak/jalur

Hanya digunakan dengan tipe `S3` artefak. Tidak digunakan untuk jenis artefak lainnya.

Jalur di bucket output untuk menempatkan file ZIP atau folder. Jika Anda tidak menentukan nilai untuk `path`, CodeBuild gunakan `namespaceType` (jika ditentukan) dan `name` untuk menentukan jalur dan nama file atau folder ZIP keluaran build. Misalnya, jika Anda menentukan `MyPath` untuk `path` dan `MyArtifact.zip` untuk `name`, jalur dan nama akan menjadi `MyPath/MyArtifact.zip`.

### artefak/Namespacetype

Hanya digunakan dengan tipe `S3` artefak. Tidak digunakan untuk jenis artefak lainnya.

Namespace dari file atau folder ZIP keluaran build. Nilai yang valid mencakup `BUILD_ID` dan `NONE`. Gunakan `BUILD_ID` untuk menyisipkan ID build ke jalur file atau folder ZIP keluaran build. Jika tidak sesuai, gunakan `NONE`. Jika Anda tidak menentukan nilai untuk `namespaceType`, CodeBuild gunakan `path` (jika ditentukan) dan `name` untuk menentukan jalur dan nama file atau folder ZIP keluaran build. Misalnya, jika Anda menentukan `MyPath` untuk `path`, `BUILD_ID` untuk `namespaceType`, dan `MyArtifact.zip` untuk `name`, jalur dan nama akan menjadi `MyPath/build-ID/MyArtifact.zip`.



## artefak/nama

Hanya digunakan dengan tipe S3 artefak. Tidak digunakan untuk jenis artefak lainnya.

Nama file ZIP keluaran build atau folder di dalamnya `location`. Misalnya, jika Anda menentukan `MyPath` untuk path dan `MyArtifact.zip` untuk `name`, jalur dan nama akan menjadi `MyPath/MyArtifact.zip`.

## artefak/ overrideArtifactName

Hanya digunakan dengan tipe artefak S3. Tidak digunakan untuk jenis artefak lainnya.

Opsional. Jika disetel ke `true`, nama yang ditentukan dalam `artifacts` blok file `buildspec` akan diganti. Untuk informasi selengkapnya, lihat [Buat referensi spesifikasi untuk CodeBuild](#).

## artefak/kemasan

Hanya digunakan dengan tipe S3 artefak. Tidak digunakan untuk jenis artefak lainnya.

Opsional. Menentukan bagaimana untuk paket artefak. Nilai yang diizinkan adalah:

NONE

Buat folder yang berisi artefak build. Ini adalah nilai default.

ZIP

Buat file ZIP yang berisi artefak build.

## secondaryArtifacts

Opsional. Array [ProjectArtifacts](#) objek yang berisi informasi tentang pengaturan artefak sekunder untuk proyek build. Anda dapat menambahkan hingga 12 artefak sekunder. `secondaryArtifacts` Menggunakan banyak pengaturan yang sama yang digunakan oleh objek.

## cache

Wajib. [ProjectCache](#) Objek yang berisi informasi tentang pengaturan cache proyek build ini. Untuk informasi selengkapnya, lihat [Membangun caching](#).

## lingkungan

Wajib. [ProjectEnvironment](#) Objek yang berisi informasi tentang pengaturan lingkungan build proyek ini. Pengaturan ini meliputi:

## lingkungan/tipe

Wajib. Jenis lingkungan bangunan. Untuk informasi selengkapnya, lihat [mengetik](#) Referensi CodeBuild API.

## lingkungan/gambar

Wajib. Pengidentifikasi gambar Docker yang digunakan oleh lingkungan build ini. *Biasanya, identifier ini dinyatakan sebagai image-name: tag.* Misalnya, di repositori Docker yang CodeBuild digunakan untuk mengelola gambar Docker-nya, ini bisa jadi. `aws/codebuild/standard:5.0` Di Docker Hub, `maven:3.3.9-jdk-8`. Di Amazon ECR, `account-id.dkr.ecr.region-id.amazonaws.com/your-Amazon-ECR-repo-name:tag`. Untuk informasi selengkapnya, lihat [Gambar Docker disediakan oleh CodeBuild](#).

## lingkungan/ComputeType

Wajib. Menentukan sumber daya komputasi yang digunakan oleh lingkungan build ini. Untuk informasi selengkapnya, lihat [ComputeType](#) di Referensi API. CodeBuild

## lingkungan/sertifikat

Opsional. ARN bucket Amazon S3, awalan jalur, dan kunci objek yang berisi sertifikat yang disandikan PEM. Kunci objek dapat berupa file.pem atau file.zip yang berisi sertifikat yang dikodekan PEM. Misalnya, jika nama bucket Amazon S3 Anda, awalan jalur Anda adalah `<my-bucket>`, dan nama kunci objek Anda `<cert><certificate.pem>`, maka format yang dapat diterima untuk certificate are atau. `<my-bucket/cert/certificate.pem>`  
`arn:aws:s3:::<my-bucket/cert/certificate.pem>`

## lingkungan/lingkunganVariabel

Opsional. Array [EnvironmentVariable](#) objek yang berisi variabel lingkungan yang ingin Anda tentukan untuk lingkungan build ini. Setiap variabel lingkungan dinyatakan sebagai objek yang berisi `name`, `value`, dan `type` dari `name`, `value`, dan `type`.

Konsol dan AWS CLI pengguna dapat melihat semua variabel lingkungan. Jika Anda tidak memiliki kekhawatiran tentang visibilitas variabel lingkungan Anda, atur `name` dan `value`, dan atur `type` ke `PLAINTEXT`.

Kami menyarankan Anda menyimpan variabel lingkungan dengan nilai sensitif, seperti ID kunci AWS akses, kunci akses AWS rahasia, atau kata sandi, sebagai parameter di Amazon EC2 Systems Manager Parameter Store atau AWS Secrets Manager. Untuk `name`, untuk parameter yang disimpan itu, tetapkan pengenal CodeBuild untuk referensi.

Jika Anda menggunakan Amazon EC2 Systems Manager Parameter Store, `value` untuk, tetapkan nama parameter sebagai disimpan di Parameter Store. Atur `type` ke `PARAMETER_STORE`. Menggunakan parameter bernama `/CodeBuild/dockerLoginPassword` sebagai contoh, atur `name` ke `LOGIN_PASSWORD`. Atur `value` ke `/CodeBuild/dockerLoginPassword`. Atur `type` ke `PARAMETER_STORE`.

### Important

Jika Anda menggunakan Amazon EC2 Systems Manager Parameter Store, sebaiknya Anda menyimpan parameter dengan nama parameter yang dimulai `/CodeBuild/` dengan (misalnya `/CodeBuild/dockerLoginPassword`). Anda dapat menggunakan CodeBuild konsol untuk membuat parameter di Amazon EC2 Systems Manager. Pilih `Buat parameter`, lalu ikuti instruksi di kotak dialog. (Di kotak dialog itu, untuk kunci KMS, Anda dapat menentukan ARN kunci di AWS KMS akun Anda. Amazon EC2 Systems Manager menggunakan kunci ini untuk mengenkripsi nilai parameter selama penyimpanan dan mendekripsi selama pengambilan.) Jika Anda menggunakan CodeBuild konsol untuk membuat parameter, konsol memulai nama parameter dengan `/CodeBuild/` saat sedang disimpan. Untuk informasi selengkapnya, lihat [Systems Manager Parameter Store](#) dan [Systems Manager Parameter Store Console Walkthrough](#) di Panduan Pengguna Amazon EC2 Systems Manager.

Jika `project build` Anda mengacu pada parameter yang disimpan di Amazon EC2 Systems Manager Parameter Store, peran layanan `project build` harus mengizinkan tindakan `ssm:GetParameters` tersebut. Jika Anda memilih Peran layanan baru lebih awal, CodeBuild sertakan tindakan ini dalam peran layanan default untuk proyek build Anda. Namun, jika Anda memilih Peran layanan yang ada, Anda harus menyertakan tindakan ini ke peran layanan Anda secara terpisah.

Jika `project build` Anda merujuk ke parameter yang disimpan di Amazon EC2 Systems Manager Parameter Store dengan nama parameter yang tidak dimulai `/CodeBuild/`, dan Anda memilih Peran layanan baru, Anda harus memperbarui peran layanan tersebut untuk mengizinkan akses ke nama parameter yang tidak dimulai `/CodeBuild/`. Ini karena peran layanan itu memungkinkan akses hanya ke nama parameter yang dimulai dengan `/CodeBuild/`.

Jika Anda memilih Peran layanan baru, peran layanan menyertakan izin untuk mendekripsi semua parameter di bawah `/CodeBuild/` namespace di Amazon EC2 Systems Manager Parameter Store.

Variabel lingkungan yang Anda tetapkan menggantikan variabel lingkungan yang ada. Misalnya, jika image Docker sudah berisi variabel lingkungan bernama `MY_VAR` dengan

nilai `my_value`, dan Anda menetapkan variabel lingkungan bernama `MY_VAR` dengan nilai `other_value`, `my_value` maka diganti dengan `other_value`. Demikian pula, jika image Docker sudah berisi variabel lingkungan bernama `PATH` dengan nilai `/usr/local/sbin:/usr/local/bin`, dan Anda menetapkan variabel lingkungan bernama `PATH` dengan nilai `$PATH:/usr/share/ant/bin:/usr/local/sbin:/usr/local/bin` maka digantikan oleh nilai `$PATH:/usr/share/ant/bin` literal.

Jangan mengatur variabel lingkungan apa pun dengan nama yang dimulai dengan `CODEBUILD_`. Awalan ini dicadangkan untuk penggunaan internal.

Jika variabel lingkungan dengan nama yang sama didefinisikan di beberapa tempat, nilainya ditentukan sebagai berikut:

- Nilai dalam panggilan operasi start build lebih diutamakan.
- Nilai dalam definisi proyek build akan diutamakan berikutnya.
- Nilai dalam deklarasi `buildspec` diutamakan paling rendah.

Jika Anda menggunakan Secrets Manager, untuk `value`, tetapkan nama parameter sebagai disimpan di Secrets Manager. Atur `type` ke `SECRETS_MANAGER`. Menggunakan rahasia bernama `/CodeBuild/dockerLoginPassword` sebagai contoh, atur `name` ke `LOGIN_PASSWORD`. Atur `value` ke `/CodeBuild/dockerLoginPassword`. Atur `type` ke `SECRETS_MANAGER`.

#### Important

Jika Anda menggunakan Secrets Manager, kami sarankan Anda menyimpan rahasia dengan nama yang dimulai dengan `/CodeBuild/` (misalnya, `/CodeBuild/dockerLoginPassword`). Untuk informasi lebih lanjut, lihat [Apa itu AWS Secrets Manager?](#) di Panduan Pengguna AWS Secrets Manager.

Jika proyek build Anda mengacu pada rahasia yang disimpan di Secrets Manager, peran layanan proyek build harus mengizinkan `secretsmanager:GetSecretValue` tindakan tersebut. Jika Anda memilih Peran layanan baru lebih awal, CodeBuild sertakan tindakan ini dalam peran layanan default untuk proyek build Anda. Namun, jika Anda memilih Peran layanan yang ada, Anda harus menyertakan tindakan ini ke peran layanan Anda secara terpisah.

Jika proyek build mengacu pada rahasia yang disimpan di Secrets Manager dengan nama rahasia yang tidak dimulai dengan `/CodeBuild/`, dan Anda memilih Peran layanan baru, Anda harus memperbarui peran layanan untuk mengizinkan akses ke nama rahasia yang tidak

dimulai/`CodeBuild/`. Ini karena peran layanan memungkinkan akses hanya ke nama rahasia yang dimulai dengan/`CodeBuild/`.

Jika Anda memilih Peran layanan baru, peran layanan menyertakan izin untuk mendekripsi semua rahasia di bawah `/CodeBuild/` namespace di Secrets Manager.

## lingkungan/`registryCredential`

Opsional. [RegistryCredential](#) Objek yang menentukan kredensial yang menyediakan akses ke registri Docker pribadi.

### Lingkungan/`RegistryCredential/kredensi`

Menentukan ARN atau nama kredensial yang dibuat menggunakan AWS Managed Services. Anda dapat menggunakan nama kredensialnya hanya jika ada di Wilayah Anda saat ini.

### Lingkungan/`RegistryCredential/CredentialProvider`

Satu-satunya nilai yang valid adalah `SECRETS_MANAGER`.

Ketika ini diatur:

- `imagePullCredentials` harus diatur ke `SERVICE_ROLE`.
- Gambar tidak dapat berupa gambar yang dikuratori atau gambar Amazon ECR.

## lingkungan/`Jenis imagePullCredentials`

Opsional. Jenis kredensial yang CodeBuild digunakan untuk menarik gambar di build Anda. Ada dua nilai yang valid:

### `CODEBUILD`

`CODEBUILD` menentukan yang CodeBuild menggunakan kredensialnya sendiri. Anda harus mengedit kebijakan repositori Amazon ECR Anda untuk mempercayai prinsip layanan.

CodeBuild

### `SERVICE_ROLE`

Menentukan yang CodeBuild menggunakan peran layanan proyek build Anda.

Saat Anda menggunakan gambar registri lintas akun atau pribadi, Anda harus menggunakan `SERVICE_ROLE` kredensial. Saat Anda menggunakan gambar yang CodeBuild dikuratori, Anda harus menggunakan `CODEBUILD` kredensial.

## lingkungan/PrivilegedMode

Setel ke `true` hanya jika Anda berencana menggunakan proyek build ini untuk membangun image Docker. Jika tidak, semua build terkait yang mencoba berinteraksi dengan daemon Docker gagal. Anda juga harus memulai daemon Docker sehingga build Anda dapat berinteraksi dengannya. Salah satu cara untuk melakukannya adalah dengan menginisialisasi daemon Docker di `install` fase file `buildspec` Anda dengan menjalankan perintah build berikut. Jangan jalankan perintah ini jika Anda menentukan image lingkungan build yang disediakan oleh CodeBuild dengan dukungan Docker.

### Note

Secara default, daemon Docker diaktifkan untuk build non-VPC. Jika Anda ingin menggunakan kontainer Docker untuk build VPC, [lihat Runtime Privilege dan Kemampuan Linux di situs web Docker Docs](#) dan aktifkan mode istimewa. Juga, Windows tidak mendukung mode istimewa.

```
- nohup /usr/local/bin/dockerd --host=unix:///var/run/docker.sock --
host=tcp://127.0.0.1:2375 --storage-driver=overlay2 &
- timeout 15 sh -c "until docker info; do echo .; sleep 1; done"
```

## serviceRole

Wajib. ARN dari peran layanan CodeBuild digunakan untuk berinteraksi dengan layanan atas nama pengguna (misalnya, `arn:aws:iam::account-id:role/role-name`).

## timeoutInMinutes

Opsional. Jumlah menit, antara 5 hingga 480 (8 jam), setelah itu CodeBuild menghentikan build jika tidak selesai. Jika tidak ditentukan, default 60 digunakan. Untuk menentukan apakah dan kapan CodeBuild menghentikan build karena batas waktu, jalankan `batch-get-builds` perintah. Untuk menentukan apakah build telah berhenti, lihat di output untuk `buildStatus` nilai `FAILED`. Untuk menentukan kapan waktu build habis, lihat di output untuk `endTime` nilai yang terkait dengan `phaseStatus` nilai. `TIMED_OUT`

## queuedTimeoutInMenit

Opsional. Jumlah menit, antara 5 hingga 480 (8 jam), setelah itu CodeBuild menghentikan build jika masih antri. Jika tidak ditentukan, default 60 digunakan.

## encryptionKey

Opsional. Alias atau ARN yang digunakan CodeBuild oleh untuk mengenkripsi output build. AWS KMS key Jika Anda menentukan alias, gunakan format `arn:aws:kms:region-ID:account-ID:key/key-ID` atau, jika ada alias, gunakan formatnya. `alias/key-alias` Jika tidak ditentukan, kunci KMS yang AWS dikelola untuk Amazon S3 digunakan.

## tag

Opsional. Array objek [Tag](#) yang menyediakan tag yang ingin Anda kaitkan dengan proyek build ini. Anda dapat menentukan hingga 50 tag. Tag ini dapat digunakan oleh AWS layanan apa pun yang mendukung tag proyek CodeBuild build. Setiap tag dinyatakan sebagai objek dengan a key dan `value`.

## vpcConfig

Opsional. [VpcConfig](#) Objek yang berisi informasi informasi tentang konfigurasi VPC untuk proyek Anda. Untuk informasi selengkapnya, lihat [GunakanAWS CodeBuilddengan Amazon Virtual Private Cloud](#).

Properti ini meliputi:

## vpcl

Wajib. ID VPC yang CodeBuild menggunakan. Jalankan perintah ini untuk mendapatkan daftar semua ID VPC di Wilayah Anda:

```
aws ec2 describe-vpcs --region <region-ID>
```

## subnet

Wajib. Array ID subnet yang menyertakan sumber daya yang digunakan oleh CodeBuild. Jalankan perintah ini untuk mendapatkan ID ini:

```
aws ec2 describe-subnets --filters "Name=vpc-id,Values=<vpc-id>" --region <region-ID>
```

## securityGroupIds

Wajib. Array ID grup keamanan yang digunakan oleh CodeBuild untuk memungkinkan akses ke sumber daya di VPC. Jalankan perintah ini untuk mendapatkan ID ini:

```
aws ec2 describe-security-groups --filters "Name=vpc-id,Values=<vpc-id>" --<region-ID>
```

## badgeEnabled

Opsional. Menentukan apakah akan menyertakan lencana build dengan proyek Anda CodeBuild . Setel `true` untuk mengaktifkan lencana build, atau `false` lainnya. Untuk informasi selengkapnya, lihat [Buat sampel lencana dengan CodeBuild](#).

## LogsConfig

[LogsConfig](#) Objek yang berisi informasi tentang lokasi log build ini.

### LogsConfig/ cloudWatchLogs

[CloudWatchLogsConfig](#) Objek yang berisi informasi tentang mendorong log ke CloudWatch Log.

### LogsConfig/ S3logs

LogsConfig Objek [S3](#) yang berisi informasi tentang mendorong log ke Amazon S3.

## fileSystemLocations

Opsional. Array [ProjectFileSystemsLocation](#) objek yang berisi informasi tentang konfigurasi Amazon EFS Anda.

## buildBatchConfig

Opsional. `buildBatchConfig` Objek adalah [ProjectBuildBatchConfig](#) struktur yang berisi informasi konfigurasi build batch untuk proyek.

### buildBatchConfig/ServiceRole

Peran layanan ARN untuk proyek pembuatan batch.



## buildBatchConfig/Combineartifacts

Nilai Boolean yang menentukan apakah akan menggabungkan artefak build untuk batch build menjadi satu lokasi artefak.

## buildBatchConfig/batasan/ maximumBuildsAllowed

Jumlah maksimum build yang diizinkan.

## buildBatchConfig/batasan/ computeTypesAllowed

Array string yang menentukan jenis komputasi yang diizinkan untuk pembangunan batch. Lihat [Membangun tipe komputasi lingkungan](#) untuk nilai ini.

## buildBatchConfig/timeoutInMinutes

Jumlah waktu maksimum, dalam hitungan menit, pembuatan batch harus diselesaikan.

## buildBatchConfig/batchReportMode

Menentukan cara laporan status build dikirim ke penyedia sumber untuk build batch. Nilai yang valid meliputi:

REPORT\_AGGREGATED\_BATCH

(Default) Gabungkan semua status build ke dalam satu laporan status.

REPORT\_INDIVIDUAL\_BUILDS

Kirim laporan status terpisah untuk setiap build individu.

## concurrentBuildLimit

Jumlah maksimum membangun bersamaan yang diizinkan untuk proyek ini.

Pembuatan baru hanya dimulai jika jumlah pembuatan saat ini kurang dari atau sama dengan batas ini. Jika jumlah pembuatan saat ini memenuhi batas ini, pembuatan baru ditutup dan tidak dijalankan.

## Buat proyek

Untuk membuat proyek, jalankan [create-project](#) perintah lagi, meneruskan file JSON Anda:

```
aws codebuild create-project --cli-input-json file://<json-file>
```

Jika berhasil, representasi JSON dari objek [Project](#) muncul di output konsol. Lihat [Sintaks CreateProject Respons](#) untuk contoh data ini.

Kecuali untuk nama proyek build, Anda dapat mengubah pengaturan proyek build apa pun nanti. Untuk informasi selengkapnya, lihat [Mengubah setelan proyek pembangunan \(AWS CLI\)](#).

Untuk mulai menjalankan build, lihat [Jalankan build \(AWS CLI\)](#).

Jika kode sumber Anda disimpan dalam GitHub repositori, dan Anda CodeBuild ingin membangun kembali kode sumber setiap kali perubahan kode didorong ke repositori, lihat. [Mulai menjalankan build secara otomatis \(AWS CLI\)](#)

## Membuat proyek pembangunan (AWSSDK)

Untuk informasi tentang penggunaan AWS CodeBuild dengan AWSSDK, lihat [AWS Referensi SDK dan alat](#).

## Membuat proyek pembangunan (AWS CloudFormation)

Untuk informasi tentang penggunaan AWS CodeBuild bersama AWS CloudFormation, lihat [sangat AWS CloudFormation template untuk CodeBuild](#) di AWS CloudFormation Panduan Pengguna.

## Membuat aturan notifikasi

Anda dapat menggunakan aturan notifikasi untuk memberi tahu pengguna saat terjadi perubahan penting, seperti keberhasilan dan kegagalan build. Aturan notifikasi menentukan peristiwa dan topik Amazon SNS yang digunakan untuk mengirim notifikasi. Untuk informasi selengkapnya, lihat [Apa itu notifikasi?](#)

Anda dapat menggunakan konsol atau AWS CLI untuk membuat aturan notifikasi untuk AWS CodeBuild.

Untuk membuat aturan notifikasi (konsol)

1. Masuk ke AWS Management Console dan buka CodeBuild konsol di <https://console.aws.amazon.com/codebuild/>.
2. Pilih Build, pilih Build projects, lalu pilih build project tempat Anda ingin menambahkan notifikasi.
3. Pada halaman proyek build, pilih Beri tahu, lalu pilih Buat aturan notifikasi. Anda juga dapat membuka halaman Pengaturan untuk proyek build dan memilih Create notification rule.

4. Di Nama notifikasi, masukkan nama untuk aturan.
5. Di Jenis detail, pilih Dasar jika Anda hanya menginginkan informasi yang diberikan kepada Amazon yang EventBridge disertakan dalam notifikasi. Pilih Lengkap jika Anda ingin menyertakan informasi yang diberikan ke Amazon EventBridge dan informasi yang mungkin diberikan oleh CodeBuild atau pengelola notifikasi.

Untuk informasi selengkapnya, lihat [Memahami Konten Notifikasi dan Keamanan](#).

6. Di Peristiwa yang memicu notifikasi, pilih peristiwa yang ingin Anda kirimkan notifikasi. Untuk informasi selengkapnya, lihat [Peristiwa untuk Aturan Pemberitahuan tentang Proyek Pembuatan](#).
7. Di Target, lakukan salah satu langkah berikut:
  - Jika Anda telah mengonfigurasi sumber daya untuk digunakan dengan notifikasi, di Pilih tipe target, pilih salah satu dari AWS Chatbot (Slack) atau Topik SNS. Di Pilih target, pilih nama klien (untuk klien Slack yang dikonfigurasi di AWS Chatbot) atau Amazon Resource Name (ARN) dari topik Amazon SNS (untuk topik Amazon SNS yang sudah dikonfigurasi dengan kebijakan yang disyaratkan untuk notifikasi).
  - Jika Anda belum mengonfigurasi sumber daya untuk digunakan dengan notifikasi, pilih Buat target, lalu pilih Topik SNS. Berikan nama untuk topik setelah codestar-notifications-, lalu pilih Buat.

#### Note

- Jika Anda membuat topik Amazon SNS sebagai bagian dari membuat aturan notifikasi, kebijakan yang memungkinkan fitur notifikasi untuk mempublikasikan peristiwa untuk topik diterapkan untuk Anda. Menggunakan topik yang dibuat untuk aturan notifikasi membantu memastikan bahwa Anda hanya berlangganan pengguna yang ingin Anda terima notifikasinya tentang sumber daya ini.
- Anda tidak dapat membuat klien AWS Chatbot sebagai bagian dari pembuatan aturan notifikasi. Jika Anda memilih AWS Chatbot (Slack), Anda akan melihat tombol yang mengarahkan Anda untuk mengkonfigurasi klien di AWS Chatbot. Memilih opsi itu membuka konsol AWS Chatbot tersebut. Untuk informasi selengkapnya, lihat [Konfigurasi Integrasi Antara Pemberitahuan dan AWS Chatbot](#).
- Jika ingin menggunakan topik Amazon SNS yang ada sebagai target, Anda harus menambahkan kebijakan yang diperlukan untuk AWS CodeStar Pemberitahuan selain kebijakan lain yang mungkin ada untuk topik tersebut. Untuk informasi selengkapnya,

lihat [Konfigurasi Topik Amazon SNS untuk Notifikasi](#) dan [Memahami Konten Notifikasi dan Keamanan](#).

8. Untuk menyelesaikan pembuatan aturan, pilih Kirim.
9. Anda harus membuat pengguna berlangganan ke topik Amazon SNS untuk aturan tersebut sebelum mereka dapat menerima notifikasi. Untuk informasi selengkapnya, lihat [Buat Pengguna Berlangganan ke Topik Amazon SNS yang Merupakan Target](#). Anda juga dapat mengatur integrasi antara pemberitahuan dan AWS Chatbot untuk mengirim notifikasi ke ruang obrolan Amazon Chime. Untuk informasi selengkapnya, lihat [Konfigurasi Integrasi Antara Notifikasi dan AWS Chatbot](#).

### Membuat aturan notifikasi (AWS CLI)

1. Di terminal atau prompt perintah, jalankan perintah `create-notification rule` untuk menghasilkan kerangka JSON:

```
aws codestarnotifications create-notification-rule --generate-cli-skeleton
> rule.json
```

Anda dapat memberi nama file apa pun yang Anda inginkan. Dalam contoh ini, file diberi nama *rule.json*.

2. Buka file JSON dalam editor plaintext dan edit untuk memasukkan sumber daya, jenis peristiwa, dan target yang Anda inginkan untuk aturan tersebut. Contoh berikut menunjukkan aturan notifikasi bernama **MyNotificationRule** untuk proyek build bernama *MyBuildProject* dalam AWS account dengan ID *123456789012*. Notifikasi dikirim dengan tipe detail lengkap ke topik Amazon SNS bernama *codestar-notifications-MyNotificationTopic* saat build berhasil:

```
{
 "Name": "MyNotificationRule",
 "EventIds": [
 "codebuild-project-build-state-succeeded"
],
 "Resource": "arn:aws:codebuild:us-east-2:123456789012:MyBuildProject",
 "Targets": [
 {
 "TargetType": "SNS",
```

```
 "TargetAddress": "arn:aws:sns:us-east-2:123456789012:codestar-
notifications-MyNotificationTopic"
 }
],
 "Status": "ENABLED",
 "DetailType": "FULL"
}
```

Simpan file tersebut.

3. Menggunakan file yang baru saja Anda edit, di terminal atau baris perintah, jalankan `create-notification-rule` perintah lagi untuk membuat aturan notifikasi:

```
aws codestarnotifications create-notification-rule --cli-input-json
file://rule.json
```

4. Jika berhasil, perintah tersebut mengembalikan ARN aturan notifikasi, yang serupa dengan berikut ini:

```
{
 "Arn": "arn:aws:codestar-notifications:us-east-1:123456789012:notificationrule/
dc82df7a-EXAMPLE"
}
```

## Melihat daftar nama proyek build diAWS CodeBuild

Anda dapat menggunakanAWS CodeBuildkonsol,AWS CLI, atauAWSSDK untuk melihat daftar proyek build di CodeBuild.

### Topik

- [Melihat daftar nama proyek pembangunan \(konsol\)](#)
- [Melihat daftar nama proyek build \(AWS CLI\)](#)
- [Melihat daftar nama proyek build \(AWSSDKs\)](#)

### Melihat daftar nama proyek pembangunan (konsol)

Anda dapat melihat daftar proyek pembangunan di sebuahAWSWilayah di konsol. Informasi mencakup nama, penyedia sumber, repositori, status build terbaru, dan deskripsi, jika ada.

1. Buka AWS CodeBuild konsol di <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Di panel navigasi, pilih Proyek pembangunan.

#### Note

Secara default, hanya 10 proyek build terbaru yang ditampilkan. Untuk melihat proyek pembangunan lainnya, pilih ikon roda gigi, dan kemudian pilih nilai yang berbeda Proyek per halaman atau gunakan panah belakang dan depan.

## Melihat daftar nama proyek build (AWS CLI)

Jalankan perintah `list-projects`:

```
aws codebuild list-projects --sort-by sort-by --sort-order sort-order --next-token next-token
```

Di perintah sebelumnya, ganti placeholder berikut:

- ***sort-by***: String opsional digunakan untuk menunjukkan kriteria yang akan digunakan untuk daftar membangun nama proyek. Nilai yang valid meliputi:
  - **CREATED\_TIME**: Cantumkan nama proyek build berdasarkan kapan setiap proyek build dibuat.
  - **LAST\_MODIFIED\_TIME**: Cantumkan nama proyek build berdasarkan kapan informasi tentang setiap proyek build terakhir diubah.
  - **NAME**: Cantumkan nama proyek build berdasarkan nama masing-masing proyek build.
- ***sort-order***: string opsional digunakan untuk menunjukkan urutan di mana untuk daftar membangun proyek, berdasarkan ***sort-by***. Nilai yang valid mencakup **ASCENDING** dan **DESCENDING**.
- ***Token berikutnya***: String opsional. Selama menjalankan sebelumnya, jika ada lebih dari 100 item dalam daftar, hanya 100 item pertama yang dikembalikan, bersama dengan string unik yang disebut **Token berikutnya**. Untuk mendapatkan batch item berikutnya dalam daftar, jalankan perintah ini lagi, tambahkan token berikutnya ke panggilan. Untuk mendapatkan semua item dalam daftar, terus jalankan perintah ini dengan setiap token berikutnya berikutnya, sampai tidak ada lagi token berikutnya dikembalikan.

Misalnya, jika Anda menjalankan perintah ini:

```
aws codebuild list-projects --sort-by NAME --sort-order ASCENDING
```

Hasil yang mirip dengan berikut ini mungkin muncul di output:

```
{
 "nextToken": "Ci33ACF6...The full token has been omitted for brevity...U+AkMx8=",
 "projects": [
 "codebuild-demo-project",
 "codebuild-demo-project2",
 ... The full list of build project names has been omitted for brevity ...
 "codebuild-demo-project99"
]
}
```

Jika Anda menjalankan perintah ini lagi:

```
aws codebuild list-projects --sort-by NAME --sort-order ASCENDING --next-token
Ci33ACF6...The full token has been omitted for brevity...U+AkMx8=
```

Hasil yang mirip dengan berikut ini mungkin muncul di output:

```
{
 "projects": [
 "codebuild-demo-project100",
 "codebuild-demo-project101",
 ... The full list of build project names has been omitted for brevity ...
 "codebuild-demo-project122"
]
}
```

## Melihat daftar nama proyek build (AWSSDKs)

Untuk informasi lebih lanjut tentang penggunaan AWS CodeBuild dengan AWSSDK, lihat [AWS Referensi SDK dan alat](#).

## Lihat detail proyek build di AWS CodeBuild

Anda dapat menggunakan AWS CodeBuild konsol, AWS CLI, atau AWSSDKs untuk melihat detail proyek pembangunan di CodeBuild.

### Topik

- [Melihat detail proyek pembangunan \(konsol\)](#)
- [Lihat detail proyek build \(AWS CLI\)](#)
- [Lihat detail proyek build \(AWSSDKs\)](#)

## Melihat detail proyek pembangunan (konsol)

1. Buka AWS CodeBuild konsol di <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Di panel navigasi, pilih Proyek pembangunan.

### Note

Secara default, hanya 10 proyek build terbaru yang ditampilkan. Untuk melihat proyek pembangunan, pilih ikon roda gigi, dan kemudian pilih nilai yang berbeda Proyek per halaman atau gunakan panah belakang dan depan.

3. Dalam daftar proyek pembangunan, di Nama kolom, pilih link untuk membangun proyek.
4. Pada Proyek pembangunan: **nama proyek** halaman, pilih Detail pembangunan.

## Lihat detail proyek build (AWS CLI)

Jalankan perintah batch-get-projects:

```
aws codebuild batch-get-projects --names names
```

Di perintah sebelumnya, ganti placeholder berikut:

- **nama**: String yang diperlukan digunakan untuk menunjukkan satu atau lebih nama proyek build untuk melihat rincian tentang. Untuk menentukan lebih dari satu proyek pembangunan, pisahkan setiap nama proyek build dengan spasi. Anda dapat menentukan hingga 100 nama proyek pembangunan. Untuk mendapatkan daftar proyek pembangunan, lihat [Melihat daftar nama proyek build \(AWS CLI\)](#).

Misalnya, jika Anda menjalankan perintah ini:

```
aws codebuild batch-get-projects --names codebuild-demo-project codebuild-demo-project2
my-other-demo-project
```



Hasil yang mirip dengan berikut mungkin muncul dalam output. Elips (. . .) digunakan untuk mewakili data dihilangkan untuk singkatnya.

```
{
 "projectsNotFound": [
 "my-other-demo-project"
],
 "projects": [
 {
 ...
 "name": "codebuild-demo-project",
 ...
 },
 {
 ...
 "name": "codebuild-demo-project2",
 ...
 }
]
}
```

Dalam output sebelumnya, `projectsNotFound` array daftar setiap membangun nama proyek yang ditentukan, tetapi tidak ditemukan. Parameter `projects` array daftar rincian untuk setiap proyek membangun di mana informasi ditemukan. Membangun rincian proyek telah dihilangkan dari output sebelumnya untuk singkatnya. Untuk informasi lebih lanjut, lihat output [Buat proyek build \(AWS CLI\)](#).

Parameter `batch-get-projects` perintah tidak mendukung penyaringan untuk nilai-nilai properti tertentu, tetapi Anda dapat menulis script yang menghitung properti untuk proyek. Misalnya, skrip shell Linux berikut menghitung proyek di wilayah saat ini untuk akun saat ini, dan mencetak gambar yang digunakan oleh setiap proyek.

```
#!/usr/bin/sh

This script enumerates all of the projects for the current account
in the current region and prints out the image that each project is using.

imageName=""

function getImageName(){
 local environmentValues=(${1//$\t/ })
 imageName=${environmentValues[1]}
}
```

```
function processProjectInfo() {
 local projectInfo=$1

 while IFS=$'\t' read -r section value; do
 if [["$section" == *"ENVIRONMENT"*]]; then
 getImageName "$value"
 fi
 done <<< "$projectInfo"
}

Get the list of projects.
projectList=$(aws codebuild list-projects --output=text)

for projectName in $projectList
do
 if [["$projectName" != *"PROJECTS"*]]; then
 echo "====="

 # Get the detailed information for the project.
 projectInfo=$(aws codebuild batch-get-projects --output=text --names
"$projectName")

 processProjectInfo "$projectInfo"

 printf 'Project "%s" has image "%s"\n' "$projectName" "$imageName"
 fi
done
```

Untuk informasi lebih lanjut tentang penggunaan AWS CLI bersama AWS CodeBuild, lihat [Referensi baris perintah](#).

## Lihat detail proyek build (AWSSDKs)

Untuk informasi lebih lanjut tentang penggunaan AWS CodeBuild dengan AWSSDK, lihat [AWS Referensi SDK dan alat](#).

## Membangun caching di AWS CodeBuild

Anda dapat menghemat waktu ketika proyek Anda dibangun dengan menggunakan cache. Cache dapat menyimpan bagian yang dapat digunakan kembali dari lingkungan build Anda dan menggunakannya di beberapa build. Proyek build Anda dapat menggunakan salah satu dari dua

jenis caching: Amazon S3 atau lokal. Jika Anda menggunakan cache lokal, Anda harus memilih satu atau lebih dari tiga mode cache: cache sumber, cache lapisan Docker, dan cache khusus.

#### Note

Mode cache lapisan Docker hanya tersedia untuk lingkungan Linux. Jika Anda memilih mode ini, Anda harus menjalankan build Anda dalam mode istimewa. CodeBuild proyek yang diberikan mode istimewa memberikan akses kontainernya ke semua perangkat. Untuk informasi selengkapnya, lihat [hak istimewa Runtime dan kemampuan Linux](#) di situs web Docker Docs.

#### Topik

- [Caching Amazon S3](#)
- [Caching lokal](#)

## Caching Amazon S3

Caching Amazon S3 menyimpan cache di bucket Amazon S3 yang tersedia di beberapa host build. Ini adalah pilihan yang baik untuk artefak bangunan berukuran kecil hingga menengah yang lebih mahal untuk dibangun daripada diunduh. Ini bukan pilihan terbaik untuk artefak build besar karena dapat memakan waktu lama untuk ditransfer melalui jaringan Anda, yang dapat memengaruhi kinerja build. Ini juga bukan pilihan terbaik jika Anda menggunakan lapisan Docker.

## Caching lokal

Caching lokal menyimpan cache secara lokal pada host build yang hanya tersedia untuk host build tersebut. Ini adalah opsi yang baik untuk artefak build menengah hingga besar karena cache segera tersedia di host build. Ini bukan pilihan terbaik jika build Anda jarang. Ini berarti bahwa kinerja build tidak terpengaruh oleh waktu transfer jaringan.

Jika Anda memilih caching lokal, Anda harus memilih satu atau beberapa mode cache berikut:

- Mode cache sumber menyimpan metadata Git untuk sumber primer dan sekunder. Setelah cache dibuat, pembangunan berikutnya menarik perubahan antar pelaksanaan saja. Mode ini merupakan pilihan yang baik untuk proyek dengan direktori yang bersih dan berfungsi serta sumber yang merupakan repositori Git besar. Jika Anda memilih opsi ini dan proyek Anda tidak menggunakan

repositori Git (AWS CodeCommit,, Server GitHub Perusahaan GitHub, atau Bitbucket), opsi tersebut diabaikan.

- Mode cache lapisan Docker menyimpan lapisan Docker yang ada. Mode ini merupakan pilihan yang baik untuk proyek yang membangun atau menarik gambar Docker yang besar. Ia dapat mencegah masalah performa yang disebabkan oleh penarikan keluar gambar Docker yang besar dari jaringan.

#### Note

- Anda dapat menggunakan cache lapisan Docker di lingkungan Linux saja.
- Bendera `privileged` harus diatur agar proyek Anda memiliki izin Docker yang diperlukan.

Secara default, daemon Docker diaktifkan untuk build non-VPC. Jika Anda ingin menggunakan kontainer Docker untuk build VPC, [lihat Runtime Privilege dan Kemampuan Linux di situs web Docker Docs dan](#) aktifkan mode istimewa. Juga, Windows tidak mendukung mode istimewa.

- Anda harus mempertimbangkan implikasi keamanan sebelum Anda menggunakan cache lapisan Docker.

- Mode cache kustom menyimpan direktori yang Anda tentukan dalam file buildspec. Mode ini adalah pilihan yang baik jika skenario build Anda tidak cocok dengan salah satu dari dua mode cache lokal lainnya. Jika Anda menggunakan cache kustom:
  - Hanya direktori yang dapat ditentukan untuk caching. Anda tidak dapat menentukan file individu.
  - Symlink digunakan untuk mereferensikan direktori cache.
  - Direktori yang di-cache ditautkan ke pembangunan Anda sebelum mengunduh sumber proyeknya. Item yang di-cache mengganti item sumber jika memiliki nama yang sama. Direktori ditentukan menggunakan jalur cache di file buildspec. Untuk informasi selengkapnya, lihat [Sintaks Buildspec](#).
  - Hindari nama direktori yang sama di sumber dan di cache. Direktori cache lokal dapat mengganti, atau menghapus konten, direktori di repositori sumber yang memiliki nama yang sama.

**Note**

Caching lokal tidak didukung dengan jenis LINUX\_GPU\_CONTAINER lingkungan dan tipe BUILD\_GENERAL1\_2XLARGE komputasi. Untuk informasi selengkapnya, lihat [Bangun mode dan tipe komputasi lingkungan](#).

**Note**

Caching lokal tidak didukung saat Anda mengonfigurasi CodeBuild untuk bekerja dengan VPC. Untuk informasi lebih lanjut tentang menggunakan VPC dengan CodeBuild, lihat [Gunakan AWS CodeBuild dengan Amazon Virtual Private Cloud](#).

## Topik

- [Tentukan caching lokal \(CLI\)](#)
- [Tentukan caching lokal \(konsol\)](#)
- [Tentukan caching lokal \(\) AWS CloudFormation](#)

Anda dapat menggunakan AWS CLI, konsol, SDK, atau AWS CloudFormation untuk menentukan cache lokal.

## Tentukan caching lokal (CLI)

Anda dapat menggunakan `--cache` parameter di AWS CLI untuk menentukan masing-masing dari tiga jenis cache lokal.

- Untuk menentukan cache sumber:

```
--cache type=LOCAL,mode=[LOCAL_SOURCE_CACHE]
```

- Untuk menentukan cache layer Docker:

```
--cache type=LOCAL,mode=[LOCAL_DOCKER_LAYER_CACHE]
```

- Untuk menentukan cache kustom:

```
--cache type=LOCAL,mode=[LOCAL_CUSTOM_CACHE]
```

Untuk informasi selengkapnya, lihat [Buat proyek build \(AWS CLI\)](#).

Tentukan caching lokal (konsol)

Anda menentukan cache di bagian Artefak konsol. Untuk jenis Cache, pilih Amazon S3 atau Lokal. Jika Anda memilih Lokal, pilih satu atau lebih dari tiga opsi cache lokal.

**Cache type**

Local ▼

Select one or more local cache options.

**Docker layer cache**  
Caches existing Docker layers so they can be reused. Requires privileged mode.

**Source cache**  
Caches .git metadata so subsequent builds only pull the change in commits.

**Custom cache**  
Caches directories specified in the buildspec file.

Untuk informasi selengkapnya, lihat [Buat proyek build \(konsol\)](#).

Tentukan caching lokal () AWS CloudFormation

Jika Anda gunakan AWS CloudFormation untuk menentukan cache lokal, pada Cache properti, untukType, tentukanLOCAL. Contoh AWS CloudFormation kode berformat YAML berikut menentukan ketiga jenis cache lokal. Anda dapat menentukan kombinasi jenis apa pun. Jika Anda menggunakan cache layer Docker, di bawahEnvironment, Anda harus mengaturPrivilegedMode ke true dan Type keLINUX\_CONTAINER.

```
CodeBuildProject:
 Type: AWS::CodeBuild::Project
 Properties:
 Name: MyProject
 ServiceRole: <service-role>
 Artifacts:
 Type: S3
 Location: <bucket-name>
 Name: myArtifact
 EncryptionDisabled: true
 OverrideArtifactName: true
 Environment:
```

```
Type: LINUX_CONTAINER
ComputeType: BUILD_GENERAL1_SMALL
Image: aws/codebuild/standard:5.0
Certificate: <bucket/cert.zip>
PrivilegedMode must be true if you specify LOCAL_DOCKER_LAYER_CACHE
PrivilegedMode: true
Source:
 Type: GITHUB
 Location: <github-location>
 InsecureSsl: true
 GitCloneDepth: 1
 ReportBuildStatus: false
TimeoutInMinutes: 10
Cache:
 Type: LOCAL
 Modes: # You can specify one or more cache mode,
 - LOCAL_CUSTOM_CACHE
 - LOCAL_DOCKER_LAYER_CACHE
 - LOCAL_SOURCE_CACHE
```

#### Note

Secara default, daemon Docker diaktifkan untuk build non-VPC. Jika Anda ingin menggunakan kontainer Docker untuk build VPC, [lihat Runtime Privilege dan Kemampuan Linux di situs web Docker Docs dan](#) aktifkan mode istimewa. Juga, Windows tidak mendukung mode istimewa.

Lihat informasi yang lebih lengkap di [Membuat proyek pembangunan \(AWS CloudFormation\)](#).

## Bangun pemicu di AWS CodeBuild

### Topik

- [Buat AWS CodeBuild pemicu](#)
- [Edit AWS CodeBuild pemicu](#)

## Buat AWS CodeBuild pemicu

### Buat AWS CodeBuild pemicu (konsol)

Anda dapat membuat pemicu pada proyek untuk menjadwalkan build sekali setiap jam, hari, atau minggu. Anda juga dapat membuat pemicu menggunakan aturan khusus dengan ekspresi CloudWatch cron Amazon. Misalnya, menggunakan ekspresi cron, Anda dapat menjadwalkan build pada waktu tertentu setiap hari kerja.

#### Note

Tidak mungkin memulai build batch dari pemicu build, EventBridge peristiwa Amazon, atau AWS Step Functions tugas.

### Untuk membuat pemicu

1. Buka AWS CodeBuild konsol di <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Di panel navigasi, pilih Membangun proyek.
3. Pilih tautan untuk proyek build yang ingin Anda tambahkan pemicunya, lalu pilih tab Pemicu build.

#### Note

Secara default, 100 proyek build terbaru ditampilkan. Untuk melihat proyek build lainnya, pilih ikon roda gigi, lalu pilih nilai berbeda untuk Proyek per halaman atau gunakan panah mundur dan maju.

4. Pilih Buat pemicu.
5. Masukkan nama di nama Pemicu.
6. Dari daftar drop-down Frekuensi, pilih frekuensi untuk pemicu Anda. Jika Anda ingin membuat frekuensi menggunakan ekspresi cron, pilih Kustom.
7. Tentukan parameter untuk frekuensi pemicu Anda. Anda dapat memasukkan beberapa karakter pertama pilihan Anda di kotak teks untuk memfilter item menu tarik-turun.



 Note

Jam dan menit mulai berbasis nol. Menit awal adalah angka antara nol dan 59. Jam awal adalah angka antara nol dan 23. Misalnya, pemicu harian yang dimulai setiap hari pada pukul 12:15 P.M. memiliki jam mulai 12 dan menit mulai 15. Pemicu harian yang dimulai setiap hari pada tengah malam memiliki jam awal nol dan menit awal nol. Pemicu harian yang dimulai setiap hari pada pukul 11:59 malam memiliki jam mulai 23 dan menit mulai 59.

| Frekuensi | Parameter yang Diperlukan              | Detail                                                                                                                    |
|-----------|----------------------------------------|---------------------------------------------------------------------------------------------------------------------------|
| Per Jam   | Mulai menit                            | Gunakan menu tarik-turun menit Mulai.                                                                                     |
| Harian    | Mulai menit<br>Jam mulai               | Gunakan menu tarik-turun menit Mulai.<br><br>Gunakan menu drop-down Start hour.                                           |
| Mingguan  | Mulai menit<br>Jam mulai<br>Mulai hari | Gunakan menu tarik-turun menit Mulai.<br><br>Gunakan menu drop-down Start hour.<br><br>Gunakan menu drop-down Mulai hari. |

| Frekuensi | Parameter yang Diperlukan | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|-----------|---------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Kustom    | Ekspresi cron             | Masukkan ekspresi cron dalam ekspresi Cron. Ekspresi cron memiliki enam bidang wajib yang dipisahkan oleh spasi putih. Bidang menentukan nilai awal untuk menit, jam, hari bulan, bulan, hari dalam seminggu, dan tahun. Anda dapat menggunakan wildcard untuk menentukan rentang, nilai tambahan, dan lainnya. Misalnya, ekspresi cron <b>0 9 ? * MON-FRI</b> * menjadwalkan build setiap hari kerja pada pukul 9:00 .M. Untuk informasi selengkapnya, lihat <a href="#">Ekspresi Cron di Panduan Pengguna Acara Amazon. CloudWatch</a> |

8. Pilih Aktifkan pemicu ini.
9. (Opsional) Perluas bagian Lanjutan. Dalam versi Sumber, ketik versi sumber Anda.
  - Untuk Amazon S3, masukkan ID versi yang sesuai dengan versi artefak input yang ingin Anda buat. Jika versi Sumber dibiarkan kosong, versi terbaru digunakan.
  - Untuk AWS CodeCommit, ketik ID komit. Jika versi Sumber dibiarkan kosong, ID komit HEAD cabang default digunakan.
  - Untuk GitHub atau GitHub Enterprise, ketikkan ID komit, ID permintaan tarik, nama cabang, atau nama tag yang sesuai dengan versi kode sumber yang ingin Anda buat. Jika Anda menentukan ID permintaan tarik, itu harus menggunakan format `p1/pull-request-ID` (misalnya, p1/25). Jika Anda menentukan nama cabang, ID komit HEAD cabang digunakan. Jika versi Source kosong, ID komit HEAD cabang default digunakan.

- Untuk Bitbucket, ketik ID komit, nama cabang, atau nama tag yang sesuai dengan versi kode sumber yang ingin Anda buat. Jika Anda menentukan nama cabang, ID komit HEAD cabang digunakan. Jika versi Source kosong, ID komit HEAD cabang default digunakan.
10. (Opsional) Tentukan batas waktu antara 5 menit dan 480 menit (8 jam). Nilai ini menentukan berapa lama AWS CodeBuild mencoba membangun sebelum berhenti. Jika Jam dan Menit dibiarkan kosong, nilai batas waktu default yang ditentukan dalam proyek digunakan.
  11. Pilih Buat pemicu.

## Buat AWS CodeBuild pemicu secara terprogram

CodeBuild menggunakan EventBridge aturan Amazon untuk pemicu build. Anda dapat menggunakan EventBridge API untuk membuat pemicu build secara terprogram untuk proyek Anda. CodeBuild Lihat [Referensi Amazon EventBridge API](#) untuk informasi selengkapnya.

## Edit AWS CodeBuild pemicu

### Edit AWS CodeBuild pemicu (konsol)

Anda dapat mengedit pemicu pada proyek untuk menjadwalkan build setiap jam, hari, atau minggu sekali. Anda juga dapat mengedit pemicu untuk menggunakan aturan kustom dengan ekspresi CloudWatch cron Amazon. Misalnya, menggunakan ekspresi cron, Anda dapat menjadwalkan build pada waktu tertentu setiap hari kerja. Untuk informasi tentang membuat pemicu, lihat [Buat AWS CodeBuild pemicu](#).

### Untuk mengedit pemicu


1. Buka AWS CodeBuild konsol di <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Di panel navigasi, pilih Membangun proyek.
3. Pilih link untuk proyek build yang ingin diubah, lalu pilih tab Build triggers.

#### Note

Secara default, 100 proyek build terbaru ditampilkan. Untuk melihat proyek build lainnya, pilih ikon roda gigi, lalu pilih nilai berbeda untuk Proyek per halaman atau gunakan panah mundur dan maju.

4. Pilih tombol radio di sebelah pemicu yang ingin Anda ubah, lalu pilih Edit.

5. Dari daftar drop-down Frekuensi, pilih frekuensi untuk pemicu Anda. Jika Anda ingin membuat frekuensi menggunakan ekspresi cron, pilih Kustom.
6. Tentukan parameter untuk frekuensi pemicu Anda. Anda dapat memasukkan beberapa karakter pertama pilihan Anda di kotak teks untuk memfilter item menu tarik-turun.

 Note

Jam dan menit mulai berbasis nol. Menit awal adalah angka antara nol dan 59. Jam awal adalah angka antara nol dan 23. Misalnya, pemicu harian yang dimulai setiap hari pada pukul 12:15 P.M. memiliki jam mulai 12 dan menit mulai 15. Pemicu harian yang dimulai setiap hari pada tengah malam memiliki jam awal nol dan menit awal nol. Pemicu harian yang dimulai setiap hari pada pukul 11:59 malam memiliki jam mulai 23 dan menit mulai 59.

| Frekuensi | Parameter yang Diperlukan              | Detail                                                                                                                    |
|-----------|----------------------------------------|---------------------------------------------------------------------------------------------------------------------------|
| Per Jam   | Mulai menit                            | Gunakan menu tarik-turun menit Mulai.                                                                                     |
| Harian    | Mulai menit<br>Jam mulai               | Gunakan menu tarik-turun menit Mulai.<br><br>Gunakan menu drop-down Start hour.                                           |
| Mingguan  | Mulai menit<br>Jam mulai<br>Mulai hari | Gunakan menu tarik-turun menit Mulai.<br><br>Gunakan menu drop-down Start hour.<br><br>Gunakan menu drop-down Mulai hari. |
| Kustom    | Ekspresi cron                          | Masukkan ekspresi cron dalam ekspresi Cron. Ekspresi cron memiliki                                                        |

| Frekuensi | Parameter yang Diperlukan | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|-----------|---------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|           |                           | enam bidang wajib yang dipisahkan oleh spasi putih. Bidang menentukan nilai awal untuk menit, jam, hari bulan, bulan, hari dalam seminggu, dan tahun. Anda dapat menggunakan wildcard untuk menentukan rentang, nilai tambahan, dan lainnya. Misalnya, ekspresi cron <b>0 9 ? * MON-FRI</b> * menjadwalkan build setiap hari kerja pada pukul 9:00 .M. Untuk informasi selengkapnya, lihat <a href="#">Ekspresi Cron di Panduan Pengguna Acara</a> Amazon. CloudWatch |

## 7. Pilih Aktifkan pemicu ini.

### Note

Anda dapat menggunakan CloudWatch konsol Amazon di <https://console.aws.amazon.com/cloudwatch/> untuk mengedit versi sumber, batas waktu, dan opsi lain yang tidak tersedia di AWS CodeBuild.

## Edit AWS CodeBuild pemicu secara terprogram

CodeBuild menggunakan EventBridge aturan Amazon untuk pemicu build. Anda dapat menggunakan EventBridge API untuk mengedit pemicu build secara terprogram untuk proyek Anda. CodeBuild Lihat [Referensi Amazon EventBridge API](#) untuk informasi selengkapnya.

## Menggunakan webhooks dengan AWS CodeBuild

AWS CodeBuild mendukung integrasi webhook dengan GitHub, GitHub Enterprise Server, dan Bitbucket.

Topik

- [Praktik terbaik untuk menggunakan webhook dengan AWS CodeBuild](#)
- [Acara webhook Bitbucket](#)
- [GitHub acara webhook](#)

### Praktik terbaik untuk menggunakan webhook dengan AWS CodeBuild

Untuk proyek yang menggunakan repositori publik untuk mengatur webhook, kami merekomendasikan opsi berikut:

ACTOR\_ACCOUNT\_IDFilter pengaturan

Tambahkan ACTOR\_ACCOUNT\_ID filter ke grup filter webhook proyek Anda untuk menentukan pengguna mana yang dapat memicu build. Setiap acara webhook yang dikirimkan CodeBuild dilengkapi dengan informasi pengirim yang menentukan pengenalan aktor. CodeBuild akan memfilter webhook berdasarkan pola ekspresi reguler yang disediakan dalam filter. Anda dapat menentukan pengguna tertentu yang diizinkan untuk memicu build dengan filter ini. Lihat informasi yang lebih lengkap di [GitHub acara webhook](#) dan [Acara webhook Bitbucket](#).

FILE\_PATHFilter pengaturan

Tambahkan FILE\_PATH filter ke grup filter webhook proyek Anda untuk menyertakan atau mengecualikan file yang dapat memicu build saat diubah. Misalnya, Anda dapat menolak permintaan build untuk perubahan pada `buildspec.yml` file menggunakan pola ekspresi reguler seperti `^buildspec.yml$`, bersama dengan `excludeMatchedPattern` properti. Lihat informasi yang lebih lengkap di [GitHub acara webhook](#) dan [Acara webhook Bitbucket](#).

Cakupan izin untuk peran IAM build Anda

Build yang dipicu oleh webhook menggunakan peran layanan IAM yang ditentukan dalam proyek. Sebaiknya setel izin dalam peran layanan ke set izin minimum yang diperlukan untuk menjalankan build. Misalnya, dalam skenario pengujian dan penerapan, buat satu proyek untuk pengujian dan proyek lain untuk penerapan. Proyek pengujian menerima build webhook dari repositori, tetapi tidak memberikan izin menulis ke sumber daya Anda. Proyek penerapan

menyediakan izin tulis ke sumber daya Anda, dan filter webhook dikonfigurasi untuk hanya mengizinkan pengguna tepercaya memicu build.

## Menggunakan buildspec tersimpan sebaris atau Amazon S3

Jika Anda menentukan buildspec inline dalam project itu sendiri, atau menyimpan file buildspec di bucket Amazon S3, file buildspec hanya dapat dilihat oleh pemilik proyek. Ini mencegah permintaan tarik membuat perubahan kode ke file buildspec dan memicu build yang tidak diinginkan. Untuk informasi selengkapnya, lihat [ProjectSource.buildspec](#) di Referensi API. CodeBuild

## Acara webhook Bitbucket

Anda dapat menggunakan grup filter webhook untuk menentukan peristiwa webhook Bitbucket mana yang memicu build. Misalnya, Anda dapat menentukan bahwa build hanya dipicu untuk perubahan pada cabang tertentu.

Anda dapat membuat satu atau beberapa grup filter webhook untuk menentukan peristiwa webhook mana yang memicu build. Build dipicu jika grup filter apa pun mengevaluasi true, yang terjadi ketika semua filter dalam grup dievaluasi menjadi true. Saat Anda membuat grup filter, Anda menentukan:

### Sebuah acara

Untuk Bitbucket, Anda dapat memilih satu atau beberapa acara berikut:

- PUSH
- PULL\_REQUEST\_CREATED
- PULL\_REQUEST\_UPDATED
- PULL\_REQUEST\_MERGED
- PULL\_REQUEST\_CLOSED

Jenis acara webhook ada di header di X-Event-Key lapangan. Tabel berikut menunjukkan bagaimana nilai X-Event-Key header memetakan ke jenis acara.

### Note

Anda harus mengaktifkan merged acara dalam pengaturan webhook Bitbucket Anda jika Anda membuat grup filter webhook yang menggunakan jenis acara. PULL\_REQUEST\_MERGED Anda juga harus mengaktifkan declined acara dalam

pengaturan webhook Bitbucket Anda jika Anda membuat grup filter webhook yang menggunakan jenis acara. `PULL_REQUEST_CLOSED`

| X-Event-Key Nilai header           | Jenis peristiwa                   |
|------------------------------------|-----------------------------------|
| <code>repo:push</code>             | <code>PUSH</code>                 |
| <code>pullrequest:created</code>   | <code>PULL_REQUEST_CREATED</code> |
| <code>pullrequest:updated</code>   | <code>PULL_REQUEST_UPDATED</code> |
| <code>pullrequest:fulfilled</code> | <code>PULL_REQUEST_MERGED</code>  |
| <code>pullrequest:rejected</code>  | <code>PULL_REQUEST_CLOSED</code>  |

Untuk `PULL_REQUEST_MERGED`, jika permintaan tarik digabungkan dengan strategi squash dan cabang permintaan tarik ditutup, komit permintaan tarik asli tidak ada lagi. Dalam hal ini, variabel `CODEBUILD_WEBHOOK_MERGE_COMMIT` lingkungan berisi pengidentifikasi komit gabungan yang terjepit.

### Satu atau lebih filter opsional

Gunakan ekspresi reguler untuk menentukan filter. Agar peristiwa memicu build, setiap filter dalam grup yang terkait dengannya harus mengevaluasi ke `true`.

`ACTOR_ACCOUNT_ID(ACTOR_ID` di konsol)

Peristiwa webhook memicu build saat ID akun Bitbucket cocok dengan pola ekspresi reguler. Nilai ini muncul di `account_id` properti `actor` objek di payload filter webhook.

### `HEAD_REF`

Peristiwa webhook memicu build saat referensi kepala cocok dengan pola ekspresi reguler (misalnya, `refs/heads/branch-name` dan `refs/tags/tag-name`). `HEAD_REF` filter mengevaluasi nama referensi Git untuk cabang atau tag. Nama cabang atau tag muncul di `name` bidang `new` objek di `push` objek payload webhook. Untuk peristiwa permintaan tarik, nama cabang muncul di `name` bidang di `branch` objek `source` objek di payload webhook.



## BASE\_REF

Peristiwa webhook memicu build saat referensi dasar cocok dengan pola ekspresi reguler. `BASE_REF`Filter hanya berfungsi dengan peristiwa permintaan tarik (misalnya, `refs/heads/branch-name`). `BASE_REF`Filter mengevaluasi nama referensi Git untuk cabang. Nama cabang muncul di `name` bidang `branch` objek di `destination` objek di payload webhook.

## FILE\_PATH

Webhook memicu build saat jalur file yang diubah cocok dengan pola ekspresi reguler.

## COMMIT\_MESSAGE

Webhook memicu build saat pesan head commit cocok dengan pola ekspresi reguler.

### Note

Anda dapat menemukan payload webhook di pengaturan webhook repositori Bitbucket Anda.

## Topik

- [Filter acara webhook Bitbucket \(konsol\)](#)
- [Filter peristiwa webhook Bitbucket \(SDK\)](#)
- [Filter acara webhook Bitbucket \(AWS CloudFormation\)](#)

## Filter acara webhook Bitbucket (konsol)

Untuk menggunakan AWS Management Console untuk memfilter acara webhook:

1. Pilih Rebuild setiap kali perubahan kode didorong ke repositori ini ketika Anda membuat proyek Anda.
2. Dari jenis Acara, pilih satu atau beberapa acara.
3. Untuk memfilter saat peristiwa memicu build, di bawah Memulai build dalam kondisi ini, tambahkan satu atau beberapa filter opsional.
4. Untuk memfilter saat peristiwa tidak dipicu, di bawah Jangan memulai build dalam kondisi ini, tambahkan satu atau beberapa filter opsional.
5. Pilih Tambahkan grup filter untuk menambahkan grup filter lain.

Untuk informasi selengkapnya, lihat [Buat proyek build \(konsol\)](#) dan [WebhookFilter](#) di Referensi AWS CodeBuild API.

Dalam contoh ini, grup filter webhook memicu build untuk permintaan tarik saja:

### Filter group 1

Remove filter group

#### Event type

Add one or more webhook event filter groups to specify which events trigger a new build. If you do not add a webhook event filter group, then a new build is triggered every time a code change is pushed to your repository.

PULL\_REQUEST\_CREATED ✕

PULL\_REQUEST\_UPDATED ✕

PULL\_REQUEST\_MERGED ✕

PULL\_REQUEST\_CLOSED ✕

► Start a build under these conditions - *optional*

► Don't start a build under these conditions - *optional*

Menggunakan contoh dua grup filter, build dipicu ketika salah satu atau keduanya mengevaluasi ke true:

- Grup filter pertama menentukan permintaan tarik yang dibuat atau diperbarui pada cabang dengan nama referensi Git yang cocok dengan ekspresi reguler `^refs/heads/main$` dan referensi kepala yang cocok `^refs/heads/branch1!`.
- Grup filter kedua menentukan permintaan push pada cabang dengan nama referensi Git yang cocok dengan ekspresi `^refs/heads/branch1$` reguler.

### Webhook event filter group 1

**Event type**  
Add one or more a webhook event filter groups to specify which events trigger a new build. If you do not add a webhook event filter group, then a new build is triggered every time a code change is pushed to your repository.

▼ **Start a build under these conditions**

| ACTOR_ID - optional  | HEAD_REF - optional                                | BASE_REF - optional                             | FILE_PATH - optional |
|----------------------|----------------------------------------------------|-------------------------------------------------|----------------------|
| <input type="text"/> | <input type="text" value="^refs/heads/branch1\$"/> | <input type="text" value="^refs/heads/main\$"/> | <input type="text"/> |

COMMIT\_MESSAGE - optional

▶ **Don't start a build under these conditions**

### Webhook event filter group 2

**Event type**  
Add one or more a webhook event filter groups to specify which events trigger a new build. If you do not add a webhook event filter group, then a new build is triggered every time a code change is pushed to your repository.

▼ **Start a build under these conditions**

| ACTOR_ID - optional  | HEAD_REF - optional                                | BASE_REF - optional  | FILE_PATH - optional |
|----------------------|----------------------------------------------------|----------------------|----------------------|
| <input type="text"/> | <input type="text" value="^refs/heads/branch1\$"/> | <input type="text"/> | <input type="text"/> |

COMMIT\_MESSAGE - optional

▶ **Don't start a build under these conditions**

Dalam contoh ini, grup filter webhook memicu build untuk semua permintaan kecuali peristiwa tag.

### Filter group 1 Remove filter group

**Event type**  
Add one or more webhook event filter groups to specify which events trigger a new build. If you do not add a webhook event filter group, then a new build is triggered every time a code change is pushed to your repository.

PUSH X

PULL\_REQUEST\_CREATED X

PULL\_REQUEST\_UPDATED X

PULL\_REQUEST\_MERGED X

PULL\_REQUEST\_CLOSED X

▶ Start a build under these conditions - optional

▼ Don't start a build under these conditions - optional Add filter

---

#### Filter 1

Type

HEAD\_REF▼

Pattern

^refs/tags/.\*

Dalam contoh ini, grup filter webhook memicu build hanya jika file dengan nama yang cocok dengan ekspresi `^buildspec.*` reguler berubah.

## Webhook event filter group 1

## Event type



▼ Start a build under these conditions

ACTOR\_ID - optional

HEAD\_REF - optional

BASE\_REF - optional

FILE\_PATH - optional

COMMIT\_MESSAGE - optional

► Don't start a build under these conditions

Dalam contoh ini, grup filter webhook memicu build hanya ketika file diubah `src` atau `test` folder.

### Webhook event filter group 1

## Event type

Add one or more webhook event filter groups to specify which events trigger a new build. If you do not add a webhook event filter group, then a new build is triggered every time a code change is pushed to your repository.



▼ Start a build under these conditions

ACTOR\_ID - optional

HEAD\_REF - optional

BASE\_REF - optional

FILE\_PATH - optional

COMMIT\_MESSAGE - optional

► Don't start a build under these conditions

Dalam contoh ini, grup filter webhook memicu build hanya jika perubahan dilakukan oleh pengguna Bitbucket yang tidak memiliki ID akun yang cocok dengan ekspresi reguler. `actor-account-id`

**Note**

Untuk informasi tentang cara menemukan ID akun Bitbucket Anda, lihat <https://api.bitbucket.org/2.0/users/{user-name}>, di mana *nama pengguna* adalah *nama pengguna* Bitbucket Anda.

**Filter group 1**[Remove filter group](#)**Event type**

Add one or more webhook event filter groups to specify which events trigger a new build. If you do not add a webhook event filter group, then a new build is triggered every time a code change is pushed to your repository.

PUSH ✕

PULL\_REQUEST\_CREATED ✕

PULL\_REQUEST\_UPDATED ✕

PULL\_REQUEST\_MERGED ✕

PULL\_REQUEST\_CLOSED ✕

**▼ Start a build under these conditions - optional**[Add filter](#)**Filter 2****Type**

ACTOR\_ACCOUNT\_ID

**Pattern**

actor-account-id

Dalam contoh ini, grup filter webhook memicu build untuk acara push saat pesan head commit cocok dengan ekspresi reguler. `\[CodeBuild\]`

## Webhook event filter group 1

## Event type


**▼ Start a build under these conditions**

ACTOR\_ID - optional

HEAD\_REF - optional

BASE\_REF - optional

FILE\_PATH - optional

COMMIT\_MESSAGE - optional

**► Don't start a build under these conditions**

## Filter peristiwa webhook Bitbucket (SDK)

Untuk menggunakan AWS CodeBuild SDK untuk memfilter peristiwa webhook, gunakan `filterGroups` bidang dalam sintaks permintaan metode `CreateWebhook` atau `UpdateWebhook` API. Untuk informasi selengkapnya, lihat [WebhookFilter](#) di dalam Referensi API CodeBuild .

Untuk membuat filter webhook yang memicu build hanya untuk permintaan tarik, masukkan yang berikut ini ke dalam sintaks permintaan:

```
"filterGroups": [
 [
 {
 "type": "EVENT",
 "pattern": "PULL_REQUEST_CREATED, PULL_REQUEST_UPDATED, PULL_REQUEST_MERGED,
PULL_REQUEST_CLOSED"
 }
]
]
```

Untuk membuat filter webhook yang memicu build hanya untuk cabang tertentu, gunakan `pattern` parameter untuk menentukan ekspresi reguler untuk memfilter nama cabang. Menggunakan contoh dua grup filter, build dipicu ketika salah satu atau keduanya mengevaluasi ke `true`:

- Grup filter pertama menentukan permintaan tarik yang dibuat atau diperbarui pada cabang dengan nama referensi Git yang cocok dengan ekspresi reguler `^refs/heads/main$` dan referensi kepala yang cocok `^refs/heads/myBranch$`.
- Grup filter kedua menentukan permintaan push pada cabang dengan nama referensi Git yang cocok dengan ekspresi `^refs/heads/myBranch$` reguler.

```
"filterGroups": [
 [
 {
 "type": "EVENT",
 "pattern": "PULL_REQUEST_CREATED, PULL_REQUEST_UPDATED, PULL_REQUEST_CLOSED"
 },
 {
 "type": "HEAD_REF",
 "pattern": "^refs/heads/myBranch$"
 },
 {
 "type": "BASE_REF",
 "pattern": "^refs/heads/main$"
 }
],
 [
 {
 "type": "EVENT",
 "pattern": "PUSH"
 },
 {
 "type": "HEAD_REF",
 "pattern": "^refs/heads/myBranch$"
 }
]
]
```

Anda dapat menggunakan `excludeMatchedPattern` parameter untuk menentukan peristiwa mana yang tidak memicu build. Dalam contoh ini, build dipicu untuk semua permintaan kecuali peristiwa tag.

```
"filterGroups": [
 [
 {
 "type": "EVENT",
```



```

 "pattern": "PUSH, PULL_REQUEST_CREATED, PULL_REQUEST_UPDATED,
PULL_REQUEST_MERGED, PULL_REQUEST_CLOSED"
 },
 {
 "type": "HEAD_REF",
 "pattern": "^refs/tags/.*",
 "excludeMatchedPattern": true
 }
]
]

```

Anda dapat membuat filter yang memicu build hanya jika perubahan dilakukan oleh pengguna Bitbucket dengan ID akun. `actor-account-id`

### Note

Untuk informasi tentang cara menemukan ID akun Bitbucket Anda, lihat [https://api.bitbucket.org/2.0/users/ \*user-name\*](https://api.bitbucket.org/2.0/users/user-name), di mana *nama pengguna* adalah *nama pengguna* Bitbucket Anda.

```

"filterGroups": [
 [
 {
 "type": "EVENT",
 "pattern": "PUSH, PULL_REQUEST_CREATED, PULL_REQUEST_UPDATED,
PULL_REQUEST_MERGED, PULL_REQUEST_CLOSED"
 },
 {
 "type": "ACTOR_ACCOUNT_ID",
 "pattern": "actor-account-id"
 }
]
]

```

Anda dapat membuat filter yang memicu build hanya jika file dengan nama yang cocok dengan ekspresi reguler dalam `pattern` argumen berubah. Dalam contoh ini, grup filter menetapkan bahwa build dipicu hanya ketika file dengan nama yang cocok dengan ekspresi reguler `^buildspec.*` berubah.

```

"filterGroups": [

```

```
[
 {
 "type": "EVENT",
 "pattern": "PUSH"
 },
 {
 "type": "FILE_PATH",
 "pattern": "^buildspec.*"
 }
]
```

Dalam contoh ini, grup filter menetapkan bahwa build dipicu hanya ketika file diubah `src` atau `test` folder.

```
"filterGroups": [
 [
 {
 "type": "EVENT",
 "pattern": "PUSH"
 },
 {
 "type": "FILE_PATH",
 "pattern": "^src/.+|^test/.+"
 }
]
]
```

Anda dapat membuat filter yang memicu build hanya jika pesan head commit cocok dengan ekspresi reguler dalam argumen pola. Dalam contoh ini, grup filter menetapkan bahwa build dipicu hanya jika pesan head commit dari peristiwa push cocok dengan ekspresi `\[CodeBuild\]` reguler.

```
"filterGroups": [
 [
 {
 "type": "EVENT",
 "pattern": "PUSH"
 },
 {
 "type": "COMMIT_MESSAGE",
 "pattern": "\[CodeBuild\]"
 }
]
]
```

```

]
]

```

## Filter acara webhook Bitbucket ()AWS CloudFormation

Untuk menggunakan AWS CloudFormation template untuk memfilter peristiwa webhook, gunakan `FilterGroups` properti AWS CodeBuild proyek. Bagian AWS CloudFormation template yang diformat YAML berikut membuat dua grup filter. Bersama-sama, mereka memicu pembuatan ketika salah satu atau keduanya mengevaluasi ke BETUL:

- Grup filter pertama menentukan permintaan tarik dibuat atau diperbarui pada cabang dengan nama referensi Git yang cocok dengan ekspresi reguler `^refs/heads/main$` oleh pengguna Bitbucket yang tidak memiliki ID akun. 12345
- Grup filter kedua menentukan permintaan push dibuat pada cabang dengan nama referensi Git yang cocok dengan ekspresi `^refs/heads/.*` reguler.
- Grup filter ketiga menentukan permintaan push dengan pesan head commit yang cocok dengan ekspresi `\[CodeBuild\]` reguler.

```

CodeBuildProject:
 Type: AWS::CodeBuild::Project
 Properties:
 Name: MyProject
 ServiceRole: service-role
 Artifacts:
 Type: NO_ARTIFACTS
 Environment:
 Type: LINUX_CONTAINER
 ComputeType: BUILD_GENERAL1_SMALL
 Image: aws/codebuild/standard:5.0
 Source:
 Type: BITBUCKET
 Location: source-location
 Triggers:
 Webhook: true
 FilterGroups:
 - - Type: EVENT
 Pattern: PULL_REQUEST_CREATED,PULL_REQUEST_UPDATED
 - Type: BASE_REF
 Pattern: ^refs/heads/main$
 ExcludeMatchedPattern: false

```

```
- Type: ACTOR_ACCOUNT_ID
 Pattern: 12345
 ExcludeMatchedPattern: true
- - Type: EVENT
 Pattern: PUSH
- Type: HEAD_REF
 Pattern: ^refs/heads/.+
- Type: FILE_PATH
 Pattern: READ_ME
 ExcludeMatchedPattern: true
- - Type: EVENT
 Pattern: PUSH
- Type: COMMIT_MESSAGE
 Pattern: \[CodeBuild\]
- Type: FILE_PATH
 Pattern: ^src/.+|^test/.+
```

## GitHub acara webhook


Anda dapat menggunakan grup filter webhook untuk menentukan peristiwa GitHub webhook mana yang memicu build. Misalnya, Anda dapat menentukan bahwa build hanya dipicu untuk perubahan pada cabang tertentu.

Anda dapat membuat satu atau beberapa grup filter webhook untuk menentukan peristiwa webhook mana yang memicu build. Build dipicu jika grup filter apa pun mengevaluasi true, yang terjadi ketika semua filter dalam grup dievaluasi menjadi true. Saat Anda membuat grup filter, Anda menentukan:

### Sebuah acara

Untuk GitHub, Anda dapat memilih satu atau beberapa acara berikut: PUSH, PULL\_REQUEST\_CREATED, PULL\_REQUEST\_UPDATED, PULL\_REQUEST\_REOPENED, PULL\_REQUEST\_MERGED, PULL\_REQUEST\_CLOSED, dan PRERELEASED. Jenis acara webhook ada di X-GitHub-Event header di payload webhook. Di X-GitHub-Event header, Anda mungkin melihat pull\_request atau push. Untuk peristiwa permintaan tarik, jenisnya ada di action bidang payload peristiwa webhook. Tabel berikut menunjukkan bagaimana nilai X-GitHub-Event header dan nilai action bidang payload permintaan tarik webhook memetakan ke jenis acara yang tersedia.

| <b>X-GitHub-Event</b> Nilai header | Nilai <b>action</b> payload acara Webhook | Jenis peristiwa       |
|------------------------------------|-------------------------------------------|-----------------------|
| pull_request                       | opened                                    | PULL_REQUEST_CREATED  |
| pull_request                       | reopened                                  | PULL_REQUEST_REOPENED |
| pull_request                       | synchronize                               | PULL_REQUEST_UPDATED  |
| pull_request                       | closed dan merged lapangan adalah true    | PULL_REQUEST_MERGED   |
| pull_request                       | closed dan merged lapangan adalah false   | PULL_REQUEST_CLOSED   |
| push                               | T/A                                       | PUSH                  |
| release                            | dirilis                                   | RELEASED              |
| release                            | prarilis                                  | PRERELEASED           |

 Note

Jenis PULL\_REQUEST\_REOPENED acara dapat digunakan dengan GitHub dan GitHub Enterprise Server saja. Jenis RELEASED dan PRERELEASED acara GitHub hanya dapat digunakan dengan.

Satu atau lebih filter opsional

Gunakan ekspresi reguler untuk menentukan filter. Agar peristiwa memicu build, setiap filter dalam grup yang terkait dengannya harus mengevaluasi ke true.

`ACTOR_ACCOUNT_ID(ACTOR_ID di konsol)`

Peristiwa webhook memicu build saat ID akun Server GitHub atau GitHub Enterprise cocok dengan pola ekspresi reguler. Nilai ini ditemukan di `id` properti `sender` objek di payload webhook.

## HEAD\_REF

Peristiwa webhook memicu build saat referensi kepala cocok dengan pola ekspresi reguler (misalnya, `refs/heads/branch-name` atau `refs/tags/tag-name`). Untuk acara push, nama referensi ditemukan di `ref` properti di payload webhook. Untuk peristiwa permintaan tarik, nama cabang ditemukan di `ref` properti head objek di payload webhook.

## BASE\_REF

Peristiwa webhook memicu build saat referensi dasar cocok dengan pola ekspresi reguler (misalnya, `refs/heads/branch-name`). `BASE_REF` filter hanya dapat digunakan dengan peristiwa permintaan tarik. Nama cabang ditemukan di `ref` properti base objek di payload webhook.

## FILE\_PATH

Webhook memicu build saat jalur file yang diubah cocok dengan pola ekspresi reguler. `FILE_PATH` filter dapat digunakan dengan acara GitHub push and pull request dan GitHub Enterprise Server push events. Itu tidak dapat digunakan dengan peristiwa permintaan tarik Server GitHub Perusahaan.

## COMMIT\_MESSAGE

Webhook memicu build saat pesan head commit cocok dengan pola ekspresi reguler. `COMMIT_MESSAGE` filter dapat digunakan dengan acara GitHub push and pull request dan GitHub Enterprise Server push events. Itu tidak dapat digunakan dengan peristiwa permintaan tarik Server GitHub Perusahaan.

## TAG\_NAME

Webhook memicu build saat nama tag rilis cocok dengan pola ekspresi reguler. `TAG_NAME` filter dapat digunakan dengan peristiwa permintaan yang GitHub dirilis dan prarilis.

## RELEASE\_NAME

Webhook memicu build saat nama rilis cocok dengan pola ekspresi reguler. `RELEASE_NAME` filter dapat digunakan dengan peristiwa permintaan yang GitHub dirilis dan prarilis.

### Note

Anda dapat menemukan payload webhook di pengaturan webhook repositori Anda. [GitHub](#)

## Topik

- [Filter acara GitHub webhook \(konsol\)](#)
- [Filter acara GitHub webhook \(SDK\)](#)
- [Filter acara GitHub webhook \(AWS CloudFormation\)](#)

### Filter acara GitHub webhook (konsol)

Di Acara webhook sumber utama, pilih yang berikut ini. Bagian ini hanya tersedia ketika Anda memilih Repositori di GitHub akun saya untuk repositori sumber.

1. Pilih Rebuild setiap kali perubahan kode didorong ke repositori ini ketika Anda membuat proyek Anda.
2. Dari jenis Acara, pilih satu atau beberapa acara.
3. Untuk memfilter saat peristiwa memicu build, di bawah Memulai build dalam kondisi ini, tambahkan satu atau beberapa filter opsional.
4. Untuk memfilter saat peristiwa tidak dipicu, di bawah Jangan memulai build dalam kondisi ini, tambahkan satu atau beberapa filter opsional.
5. Pilih Tambahkan grup filter untuk menambahkan grup filter lain, jika diperlukan.

Untuk informasi selengkapnya, lihat [Buat proyek build \(konsol\)](#) dan [WebhookFilter](#) di Referensi AWS CodeBuild API.

Dalam contoh ini, grup filter webhook memicu build untuk permintaan tarik saja:

### Filter group 1 Remove filter group

**Event type**  
Add one or more webhook event filter groups to specify which events trigger a new build. If you do not add a webhook event filter group, then a new build is triggered every time a code change is pushed to your repository.

PULL\_REQUEST\_CREATED ✕

PULL\_REQUEST\_UPDATED ✕

PULL\_REQUEST\_REOPENED ✕

PULL\_REQUEST\_MERGED ✕

PULL\_REQUEST\_CLOSED ✕

▶ **Start a build under these conditions - optional**

▶ **Don't start a build under these conditions - optional**

Menggunakan contoh dua grup filter webhook, build dipicu ketika salah satu atau keduanya mengevaluasi ke true:

- Grup filter pertama menentukan permintaan tarik yang dibuat, diperbarui, atau dibuka kembali pada cabang dengan nama referensi Git yang cocok dengan ekspresi reguler `^refs/heads/main$` dan referensi kepala yang cocok. `^refs/heads/branch1$`
- Grup filter kedua menentukan permintaan push pada cabang dengan nama referensi Git yang cocok dengan ekspresi `^refs/heads/branch1$` reguler.



### Webhook event filter group 1

**Event type**  
Add one or more a webhook event filter groups to specify which events trigger a new build. If you do not add a webhook event filter group, then a new build is triggered every time a code change is pushed to your repository.

▼ **Start a build under these conditions**

|                            |                                                    |                                                 |                             |
|----------------------------|----------------------------------------------------|-------------------------------------------------|-----------------------------|
| <b>ACTOR_ID - optional</b> | <b>HEAD_REF - optional</b>                         | <b>BASE_REF - optional</b>                      | <b>FILE_PATH - optional</b> |
| <input type="text"/>       | <input type="text" value="^refs/heads/branch1\$"/> | <input type="text" value="^refs/heads/main\$"/> | <input type="text"/>        |

**COMMIT\_MESSAGE - optional**

► **Don't start a build under these conditions**

### Webhook event filter group 2

**Event type**  
Add one or more a webhook event filter groups to specify which events trigger a new build. If you do not add a webhook event filter group, then a new build is triggered every time a code change is pushed to your repository.

▼ **Start a build under these conditions**

|                            |                                                    |                            |                             |
|----------------------------|----------------------------------------------------|----------------------------|-----------------------------|
| <b>ACTOR_ID - optional</b> | <b>HEAD_REF - optional</b>                         | <b>BASE_REF - optional</b> | <b>FILE_PATH - optional</b> |
| <input type="text"/>       | <input type="text" value="^refs/heads/branch1\$"/> | <input type="text"/>       | <input type="text"/>        |

**COMMIT\_MESSAGE - optional**

► **Don't start a build under these conditions**

Dalam contoh ini, grup filter webhook memicu build untuk semua permintaan kecuali peristiwa tag.

## Filter group 1

[Remove filter group](#)

### Event type

Add one or more webhook event filter groups to specify which events trigger a new build. If you do not add a webhook event filter group, then a new build is triggered every time a code change is pushed to your repository.

[PUSH X](#)[PULL\\_REQUEST\\_CREATED X](#)[PULL\\_REQUEST\\_UPDATED X](#)[PULL\\_REQUEST\\_REOPENED X](#)[PULL\\_REQUEST\\_MERGED X](#)[PULL\\_REQUEST\\_CLOSED X](#)

► Start a build under these conditions - *optional*

▼ Don't start a build under these conditions - *optional*

[Add filter](#)

### Filter 1

#### Type

#### Pattern

Dalam contoh ini, grup filter webhook memicu build hanya jika file dengan nama yang cocok dengan ekspresi `^buildspec.*` reguler berubah.

## Webhook event filter group 1

## Event type



▼ Start a build under these conditions

ACTOR\_ID - optional

HEAD\_REF - optional

BASE\_REF - optional

FILE\_PATH - optional

COMMIT\_MESSAGE - optional

► Don't start a build under these conditions

Dalam contoh ini, grup filter webhook memicu build hanya ketika file diubah `src` atau `test` folder.

### Webhook event filter group 1

## Event type

Add one or more webhook event filter groups to specify which events trigger a new build. If you do not add a webhook event filter group, then a new build is triggered every time a code change is pushed to your repository.



▼ Start a build under these conditions

ACTOR\_ID - optional

HEAD\_REF - optional

BASE\_REF - optional

FILE\_PATH - optional

COMMIT\_MESSAGE - optional

► Don't start a build under these conditions

Dalam contoh ini, grup filter webhook memicu build hanya jika perubahan dilakukan oleh pengguna Server GitHub Perusahaan tertentu GitHub atau dengan ID akun yang cocok dengan ekspresi reguler. `actor-account-id`

**Note**

Untuk informasi tentang cara menemukan ID GitHub akun Anda, lihat [https://api.github.com/users/nama\\_pengguna](https://api.github.com/users/nama_pengguna), di mana *nama\_pengguna* adalah *nama\_pengguna* Anda. GitHub

**Filter group 1**[Remove filter group](#)**Event type**

Add one or more webhook event filter groups to specify which events trigger a new build. If you do not add a webhook event filter group, then a new build is triggered every time a code change is pushed to your repository.

[PUSH](#) ✕[PULL\\_REQUEST\\_CREATED](#) ✕[PULL\\_REQUEST\\_UPDATED](#) ✕[PULL\\_REQUEST\\_REOPENED](#) ✕[PULL\\_REQUEST\\_MERGED](#) ✕[PULL\\_REQUEST\\_CLOSED](#) ✕

▼ **Start a build under these conditions - optional**

[Add filter](#)**Filter 2****Type****Pattern**[Remove](#)

► **Don't start a build under these conditions - optional**

Dalam contoh ini, grup filter webhook memicu build untuk acara push saat pesan head commit cocok dengan ekspresi reguler. `\[CodeBuild\]`

## Webhook event filter group 1

## Event type



▼ Start a build under these conditions

ACTOR\_ID - optional

HEAD\_REF - optional

BASE\_REF - optional

FILE\_PATH - optional

COMMIT\_MESSAGE - optional

► Don't start a build under these conditions

## Filter acara GitHub webhook (SDK)

Untuk menggunakan AWS CodeBuild SDK untuk memfilter peristiwa webhook, gunakan `filterGroups` bidang dalam sintaks permintaan metode `CreateWebhook` atau `UpdateWebhook` API. Untuk informasi selengkapnya, lihat [WebhookFilter](#) di dalam Referensi API CodeBuild .

Untuk membuat filter webhook yang memicu build hanya untuk permintaan tarik, masukkan yang berikut ini ke dalam sintaks permintaan:

```
"filterGroups": [
 [
 {
 "type": "EVENT",
 "pattern": "PULL_REQUEST_CREATED, PULL_REQUEST_UPDATED,
PULL_REQUEST_REOPENED, PULL_REQUEST_MERGED, PULL_REQUEST_CLOSED"
 }
]
]
```

Untuk membuat filter webhook yang memicu build hanya untuk cabang tertentu, gunakan `pattern` parameter untuk menentukan ekspresi reguler untuk memfilter nama cabang. Menggunakan contoh dua grup filter, build dipicu ketika salah satu atau keduanya mengevaluasi ke `true`:

- Grup filter pertama menentukan permintaan tarik yang dibuat, diperbarui, atau dibuka kembali pada cabang dengan nama referensi Git yang cocok dengan ekspresi reguler `^refs/heads/main$` dan referensi kepala yang cocok. `^refs/heads/myBranch$`
- Grup filter kedua menentukan permintaan push pada cabang dengan nama referensi Git yang cocok dengan ekspresi `^refs/heads/myBranch$` reguler.

```
"filterGroups": [
 [
 {
 "type": "EVENT",
 "pattern": "PULL_REQUEST_CREATED, PULL_REQUEST_UPDATED,
PULL_REQUEST_REOPENED"
 },
 {
 "type": "HEAD_REF",
 "pattern": "^refs/heads/myBranch$"
 },
 {
 "type": "BASE_REF",
 "pattern": "^refs/heads/main$"
 }
],
 [
 {
 "type": "EVENT",
 "pattern": "PUSH"
 },
 {
 "type": "HEAD_REF",
 "pattern": "^refs/heads/myBranch$"
 }
]
]
```

Anda dapat menggunakan `excludeMatchedPattern` parameter untuk menentukan peristiwa mana yang tidak memicu build. Misalnya, dalam contoh ini build dipicu untuk semua permintaan kecuali peristiwa tag.

```
"filterGroups": [
 [
 {
```

```

 "type": "EVENT",
 "pattern": "PUSH, PULL_REQUEST_CREATED, PULL_REQUEST_UPDATED,
PULL_REQUEST_REOPENED, PULL_REQUEST_MERGED, PULL_REQUEST_CLOSED"
 },
 {
 "type": "HEAD_REF",
 "pattern": "^refs/tags/.*",
 "excludeMatchedPattern": true
 }
]
]

```

Anda dapat membuat filter yang memicu build hanya jika file dengan nama yang cocok dengan ekspresi reguler dalam `pattern` argumen berubah. Dalam contoh ini, grup filter menetapkan bahwa build dipicu hanya ketika file dengan nama yang cocok dengan ekspresi reguler `^buildspec.*` berubah.

```

"filterGroups": [
 [
 {
 "type": "EVENT",
 "pattern": "PUSH"
 },
 {
 "type": "FILE_PATH",
 "pattern": "^buildspec.*"
 }
]
]

```

Dalam contoh ini, grup filter menetapkan bahwa build dipicu hanya ketika file diubah `src` atau `test` folder.

```

"filterGroups": [
 [
 {
 "type": "EVENT",
 "pattern": "PUSH"
 },
 {
 "type": "FILE_PATH",
 "pattern": "^src/.+|^test/.+"
 }
]
]

```

```

 }
]
]

```

Anda dapat membuat filter yang memicu build hanya jika perubahan dilakukan oleh pengguna Server GitHub Perusahaan tertentu GitHub atau dengan ID `actor-account-id` akun.

### Note

Untuk informasi tentang cara menemukan ID GitHub akun Anda, lihat [https://api.github.com/users/nama\\_pengguna](https://api.github.com/users/nama_pengguna), di mana *nama\_pengguna* adalah *nama\_pengguna* Anda. GitHub

```

"filterGroups": [
 [
 {
 "type": "EVENT",
 "pattern": "PUSH, PULL_REQUEST_CREATED, PULL_REQUEST_UPDATED,
PULL_REQUEST_REOPENED, PULL_REQUEST_MERGED, PULL_REQUEST_CLOSED"
 },
 {
 "type": "ACTOR_ACCOUNT_ID",
 "pattern": "actor-account-id"
 }
]
]

```

Anda dapat membuat filter yang memicu build hanya jika pesan head commit cocok dengan ekspresi reguler dalam argumen pola. Dalam contoh ini, grup filter menetapkan bahwa build dipicu hanya jika pesan head commit dari peristiwa push cocok dengan ekspresi `\[CodeBuild\]` reguler.

```

"filterGroups": [
 [
 {
 "type": "EVENT",
 "pattern": "PUSH"
 },
 {
 "type": "COMMIT_MESSAGE",

```



```

 "pattern": "\[CodeBuild\]"
 }
]
]

```

## Filter acara GitHub webhook ()AWS CloudFormation

Untuk menggunakan AWS CloudFormation template untuk memfilter peristiwa webhook, gunakan FilterGroups properti AWS CodeBuild proyek. Bagian AWS CloudFormation template yang diformat YAML berikut membuat dua grup filter. Bersama-sama, mereka memicu pembuatan ketika salah satu atau keduanya mengevaluasi ke BETUL:

- Grup filter pertama menentukan permintaan tarik dibuat atau diperbarui pada cabang dengan nama referensi Git yang cocok dengan ekspresi reguler `^refs/heads/main$` oleh GitHub pengguna yang tidak memiliki ID 12345 akun.
- Grup filter kedua menentukan permintaan push dibuat pada file dengan nama yang cocok dengan ekspresi reguler `README` di cabang dengan nama referensi Git yang cocok dengan ekspresi reguler `^refs/heads/.*`.
- Grup filter ketiga menentukan permintaan push dengan pesan head commit yang cocok dengan ekspresi `\[CodeBuild\]` reguler.

```

CodeBuildProject:
 Type: AWS::CodeBuild::Project
 Properties:
 Name: MyProject
 ServiceRole: service-role
 Artifacts:
 Type: NO_ARTIFACTS
 Environment:
 Type: LINUX_CONTAINER
 ComputeType: BUILD_GENERAL1_SMALL
 Image: aws/codebuild/standard:5.0
 Source:
 Type: GITHUB
 Location: source-location
 Triggers:
 Webhook: true
 FilterGroups:
 - - Type: EVENT
 Pattern: PULL_REQUEST_CREATED,PULL_REQUEST_UPDATED

```

```
- Type: BASE_REF
 Pattern: ^refs/heads/main$
 ExcludeMatchedPattern: false
- Type: ACTOR_ACCOUNT_ID
 Pattern: 12345
 ExcludeMatchedPattern: true
- - Type: EVENT
 Pattern: PUSH
- Type: HEAD_REF
 Pattern: ^refs/heads/.+
- Type: FILE_PATH
 Pattern: READ_ME
 ExcludeMatchedPattern: true
- - Type: EVENT
 Pattern: PUSH
- Type: COMMIT_MESSAGE
 Pattern: \[CodeBuild\]
- Type: FILE_PATH
 Pattern: ^src/.+|^test/.+
```

## Mengubah pengaturan proyek build diAWS CodeBuild

Anda dapat menggunakanAWS CodeBuildkonsol,AWS CLI, atauAWSSDK untuk mengubah pengaturan proyek build.

Jika Anda menambahkan pelaporan pengujian ke proyek build, pastikan peran IAM Anda memiliki izin yang dijelaskan dalam[Bekerja dengan izin laporan pengujian](#).

### Topik

- [Mengubah setelan proyek build \(konsol\)](#)
- [Mengubah setelan proyek pembangunan \(AWS CLI\)](#)
- [Mengubah pengaturan proyek build \(AWSSDK\)](#)

### Mengubah setelan proyek build (konsol)

Untuk mengubah pengaturan untuk proyek build, lakukan prosedur berikut:

1. Buka AWS CodeBuild konsol di <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Di panel navigasi, pilih Membangun proyek.

### 3. Lakukan salah satu dari cara berikut:

- Pilih tautan untuk proyek build yang ingin Anda ubah, lalu pilih Detail build.
- Pilih tombol di samping proyek build yang ingin diubah, pilih Lihat detail, lalu pilih Build details.

Anda dapat memodifikasi bagian berikut:

#### Bagian-bagian

- [Konfigurasi proyek](#)
- [Sumber](#)
- [Environment](#)
- [Buildspec](#)
- [Konfigurasi Batch](#)
- [Artifacts](#)
- [Log](#)

#### Konfigurasi proyek

Di bagian Konfigurasi proyek, pilih Edit. Ketika perubahan Anda selesai, pilih Perbarui konfigurasi untuk menyimpan konfigurasi baru.

Anda dapat memodifikasi properti berikut.

#### Deskripsi

Masukkan deskripsi opsional proyek build untuk membantu pengguna lain memahami untuk apa proyek ini digunakan.

#### Membangun lencana

Pilih Aktifkan lencana build untuk membuat status build project Anda terlihat dan dapat disematkan. Untuk informasi selengkapnya, lihat [Membangun sampel lencana](#).

#### Note

Build badge tidak berlaku jika penyedia sumber Anda adalah Amazon S3.

## Aktifkan batas build bersamaan

Jika Anda ingin membatasi jumlah build bersamaan untuk proyek ini, lakukan langkah-langkah berikut:

1. Pilih Batasi jumlah build bersamaan yang dapat dimulai proyek ini.
2. Dalam batas build bersamaan, masukkan jumlah maksimum build bersamaan yang diizinkan untuk project ini. Batas ini tidak boleh lebih besar dari batas build bersamaan yang ditetapkan untuk akun. Jika Anda mencoba memasukkan angka yang lebih besar dari batas akun, pesan kesalahan akan ditampilkan.

Pembuatan baru hanya dimulai jika jumlah pembuatan saat ini kurang dari atau sama dengan batas ini. Jika jumlah pembuatan saat ini memenuhi batas ini, pembuatan baru ditutup dan tidak dijalankan.

## Aktifkan akses build publik

Untuk membuat hasil build project Anda tersedia untuk umum, termasuk pengguna tanpa akses ke AWS akun, pilih Aktifkan akses build publik dan konfirmasi bahwa Anda ingin membuat hasil build menjadi publik. Properti berikut digunakan untuk proyek pembangunan publik:

### Peran layanan pembangunan publik

Pilih Peran layanan baru jika ingin CodeBuild membuat peran layanan baru untuk Anda, atau Peran layanan yang ada jika Anda ingin menggunakan peran layanan yang ada.

Peran layanan build publik memungkinkan CodeBuild untuk membaca CloudWatch Log dan mengunduh artefak Amazon S3 untuk build proyek. Ini diperlukan untuk membuat log dan artefak bangunan proyek tersedia untuk umum.

### Peran layanan

Masukkan nama peran layanan baru atau peran layanan yang ada.

Untuk membuat hasil build proyek Anda pribadi, hapus Aktifkan akses build publik.

Untuk informasi selengkapnya, lihat [Proyek pembangunan publik diAWS CodeBuild](#).

#### Warning

Hal-hal berikut harus diingat saat membuat hasil build proyek Anda menjadi publik:

- Semua hasil build proyek, log, dan artefak, termasuk build yang dijalankan saat proyek bersifat pribadi, tersedia untuk umum.
- Semua log build dan artefak tersedia untuk umum. Variabel lingkungan, kode sumber, dan informasi sensitif lainnya mungkin telah dihasilkan ke log dan artefak build. Anda harus berhati-hati tentang informasi apa yang dihasilkan ke log build. Beberapa praktik terbaik adalah:
  - Jangan menyimpan nilai sensitif, terutama ID kunci AWS akses dan kunci akses rahasia, dalam variabel lingkungan. Kami menyarankan Anda menggunakan Amazon EC2 Systems Manager Parameter Store AWS Secrets Manager atau untuk menyimpan nilai sensitif.
  - Ikuti [Praktik terbaik untuk menggunakan webhooks](#) untuk membatasi entitas mana yang dapat memicu build, dan jangan menyimpan buildspec dalam proyek itu sendiri, untuk memastikan bahwa webhook Anda seaman mungkin.
  - Pengguna jahat dapat menggunakan build publik untuk mendistribusikan artefak berbahaya. Kami menyarankan agar administrator proyek meninjau semua permintaan tarik untuk memverifikasi bahwa permintaan tarik adalah perubahan yang sah. Kami juga menyarankan Anda memvalidasi artefak apa pun dengan checksum mereka untuk memastikan bahwa artefak yang benar sedang diunduh.

## Informasi tambahan

Untuk Tag, masukkan nama dan nilai tag apa pun yang ingin digunakan AWS layanan pendukung. Gunakan Tambah baris untuk menambahkan tag. Anda dapat menambahkan hingga 50 tanda.

## Sumber

Di bagian Sumber, pilih Edit. Ketika perubahan Anda selesai, pilih Perbarui konfigurasi untuk menyimpan konfigurasi baru.

Anda dapat memodifikasi properti berikut:

## Penyedia sumber

Pilih jenis penyedia kode sumber. Gunakan daftar berikut untuk membuat pilihan yang sesuai untuk penyedia sumber Anda:

**Note**

CodeBuild tidak mendukung Bitbucket Server.

## Amazon S3

### Bucket

Pilih nama bucket input yang berisi kode sumber.

### Kunci objek S3 atau folder S3

Masukkan nama file ZIP atau path ke folder yang berisi kode sumber. Masukkan garis miring (/) untuk mengunduh semua yang ada di bucket S3.

### Versi sumber

Masukkan ID versi objek yang mewakili build file input Anda. Untuk informasi selengkapnya, lihat [Contoh versi sumber dengan AWS CodeBuild](#).

## CodeCommit

### Repositori

Pilih repositori yang ingin Anda gunakan.

### Jenis referensi

Pilih Cabang, tag Git, atau ID Komit untuk menentukan versi kode sumber Anda. Untuk informasi selengkapnya, lihat [Contoh versi sumber dengan AWS CodeBuild](#).

**Note**

Kami menyarankan Anda memilih nama cabang Git yang tidak terlihat seperti ID komit, seperti 811dd1ba1aba14473856cee38308caed7190c0d atau 5392f7. Ini membantu Anda menghindari tabrakan checkout Git dengan commit yang sebenarnya.

### Kedalaman klon Git

Pilih untuk membuat klon dangkal dengan riwayat terpotong ke jumlah komit yang ditentukan. Jika Anda ingin klon penuh, pilih Penuh.

## Submodul Git

Pilih Gunakan submodul Git jika Anda ingin menyertakan submodul Git dalam repositori Anda.

## Bitbucket

### Repositori

Pilih Connect menggunakan OAuth atau Connect with a Bitbucket app password dan ikuti petunjuk untuk menghubungkan (atau menyambung kembali) ke Bitbucket.

Pilih repositori publik atau repositori di akun Anda.

### Versi sumber

Masukkan cabang, ID komit, tag, atau referensi, dan ID komit. Lihat informasi yang lebih lengkap di [Contoh versi sumber dengan AWS CodeBuild](#)

#### Note

Kami menyarankan Anda memilih nama cabang Git yang tidak terlihat seperti ID komit, seperti 811dd1ba1aba14473856cee38308caed7190c0d atau 5392f7. Ini membantu Anda menghindari tabrakan checkout Git dengan commit yang sebenarnya.

## Kedalaman klon Git

Pilih kedalaman klon Git untuk membuat klon dangkal dengan riwayat terpotong ke jumlah komit yang ditentukan. Jika Anda ingin klon penuh, pilih Penuh.

## Submodul Git

Pilih Gunakan submodul Git jika Anda ingin menyertakan submodul Git dalam repositori Anda.

## Membangun status

Pilih Laporkan status build ke penyedia sumber saat build dimulai dan selesai jika Anda ingin status awal dan penyelesaian build dilaporkan ke penyedia sumber.

Untuk dapat melaporkan status build ke penyedia sumber, pengguna yang terkait dengan penyedia sumber harus memiliki akses tulis ke repo. Jika pengguna tidak memiliki akses tulis, status build tidak dapat diperbarui. Untuk informasi selengkapnya, lihat [Akses penyedia sumber](#).

Untuk konteks Status, masukkan nilai yang akan digunakan untuk name parameter dalam status komit Bitbucket. Untuk informasi selengkapnya, lihat [build](#) dalam dokumentasi Bitbucket API.

Untuk URL Target, masukkan nilai yang akan digunakan untuk url parameter dalam status komit Bitbucket. Untuk informasi selengkapnya, lihat [build](#) dalam dokumentasi Bitbucket API.

Status build yang dipicu oleh webhook selalu dilaporkan ke penyedia sumber. Agar status build yang dimulai dari konsol atau panggilan API dilaporkan ke penyedia sumber, Anda harus memilih setelan ini.

Jika build proyek Anda dipicu oleh webhook, Anda harus mendorong komit baru ke repo agar perubahan pada pengaturan ini diterapkan.

Dalam peristiwa webhook sumber utama, pilih Rebuild setiap kali perubahan kode didorong ke repositori ini jika Anda CodeBuild ingin membangun kode sumber setiap kali perubahan kode didorong ke repositori ini. Untuk informasi selengkapnya tentang webhook dan grup filter, lihat [Acara webhook Bitbucket](#)

## GitHub

### Repositori

Pilih Connect menggunakan OAuth atau Connect dengan token akses GitHub pribadi dan ikuti petunjuk untuk menghubungkan (atau menyambung kembali) ke GitHub dan mengotorisasi akses ke AWS CodeBuild

Pilih repositori publik atau repositori di akun Anda.

### Versi sumber

Masukkan cabang, ID komit, tag, atau referensi, dan ID komit. Lihat informasi yang lebih lengkap di [Contoh versi sumber dengan AWS CodeBuild](#)

#### Note

Kami menyarankan Anda memilih nama cabang Git yang tidak terlihat seperti ID komit, seperti 811dd1ba1aba14473856cee38308caed7190c0d atau 5392f7. Ini membantu Anda menghindari tabrakan checkout Git dengan commit yang sebenarnya.



## Kedalaman klon Git

Pilih kedalaman klon Git untuk membuat klon dangkal dengan riwayat terpotong ke jumlah komit yang ditentukan. Jika Anda ingin klon penuh, pilih Penuh.

## Submodul Git

Pilih Gunakan submodul Git jika Anda ingin menyertakan submodul Git dalam repositori Anda.

## Membangun status

Pilih Laporkan status build ke penyedia sumber saat build dimulai dan selesai jika Anda ingin status awal dan penyelesaian build dilaporkan ke penyedia sumber.

Untuk dapat melaporkan status build ke penyedia sumber, pengguna yang terkait dengan penyedia sumber harus memiliki akses tulis ke repo. Jika pengguna tidak memiliki akses tulis, status build tidak dapat diperbarui. Untuk informasi selengkapnya, lihat [Akses penyedia sumber](#).

Untuk konteks Status, masukkan nilai yang akan digunakan untuk `context` parameter dalam status GitHub komit. Untuk informasi selengkapnya, lihat [Membuat status komit](#) di panduan GitHub pengembang.

Untuk URL Target, masukkan nilai yang akan digunakan untuk `target_url` parameter dalam status GitHub komit. Untuk informasi selengkapnya, lihat [Membuat status komit](#) di panduan GitHub pengembang.

Status build yang dipicu oleh webhook selalu dilaporkan ke penyedia sumber. Agar status build yang dimulai dari konsol atau panggilan API dilaporkan ke penyedia sumber, Anda harus memilih setelan ini.

Jika build proyek Anda dipicu oleh webhook, Anda harus mendorong komit baru ke repo agar perubahan pada pengaturan ini diterapkan.

Dalam peristiwa webhook sumber utama, pilih Rebuild setiap kali perubahan kode didorong ke repositori ini jika Anda CodeBuild ingin membangun kode sumber setiap kali perubahan kode didorong ke repositori ini. Untuk informasi selengkapnya tentang webhook dan grup filter, lihat [GitHub acara webhook](#)

## GitHub Enterprise Server

### GitHub Token akses pribadi perusahaan

Lihat [GitHub Contoh Server Perusahaan](#) untuk informasi tentang cara menyalin token akses pribadi ke clipboard Anda. Tempel token di bidang teks, lalu pilih Simpan Token.

#### Note

Anda hanya perlu memasukkan dan menyimpan token akses pribadi sekali. CodeBuild menggunakan token ini di semua proyek future.

### Versi sumber

Masukkan permintaan tarik, cabang, ID komit, tag, atau referensi, dan ID komit. Untuk informasi selengkapnya, lihat [Contoh versi sumber dengan AWS CodeBuild](#).

#### Note

Kami menyarankan Anda memilih nama cabang Git yang tidak terlihat seperti ID komit, seperti 811dd1ba1aba14473856cee38308caed7190c0d atau 5392f7. Ini membantu Anda menghindari tabrakan checkout Git dengan commit yang sebenarnya.

### Kedalaman klon Git

Pilih kedalaman klon Git untuk membuat klon dangkal dengan riwayat terpotong ke jumlah komit yang ditentukan. Jika Anda ingin klon penuh, pilih Penuh.

### Submodul Git

Pilih Gunakan submodul Git jika Anda ingin menyertakan submodul Git dalam repositori Anda.

### Membangun status

Pilih Laporkan status build ke penyedia sumber saat build dimulai dan selesai jika Anda ingin status awal dan penyelesaian build dilaporkan ke penyedia sumber.

Untuk dapat melaporkan status build ke penyedia sumber, pengguna yang terkait dengan penyedia sumber harus memiliki akses tulis ke repo. Jika pengguna tidak memiliki akses

tulis, status build tidak dapat diperbarui. Untuk informasi selengkapnya, lihat [Akses penyedia sumber](#).

Untuk konteks Status, masukkan nilai yang akan digunakan untuk context parameter dalam status GitHub komit. Untuk informasi selengkapnya, lihat [Membuat status komit](#) di panduan GitHub pengembang.

Untuk URL Target, masukkan nilai yang akan digunakan untuk target\_url parameter dalam status GitHub komit. Untuk informasi selengkapnya, lihat [Membuat status komit](#) di panduan GitHub pengembang.

Status build yang dipicu oleh webhook selalu dilaporkan ke penyedia sumber. Agar status build yang dimulai dari konsol atau panggilan API dilaporkan ke penyedia sumber, Anda harus memilih setelan ini.

Jika build proyek Anda dipicu oleh webhook, Anda harus mendorong komit baru ke repo agar perubahan pada pengaturan ini diterapkan.

## SSL tidak aman

Pilih Aktifkan SSL yang tidak aman untuk mengabaikan peringatan SSL saat menghubungkan ke repositori proyek Enterprise Anda GitHub .

Dalam peristiwa webhook sumber utama, pilih Rebuild setiap kali perubahan kode didorong ke repositori ini jika Anda CodeBuild ingin membangun kode sumber setiap kali perubahan kode didorong ke repositori ini. Untuk informasi selengkapnya tentang webhook dan grup filter, lihat [GitHub acara webhook](#)

## Environment

Di bagian Lingkungan, pilih Edit. Ketika perubahan Anda selesai, pilih Perbarui konfigurasi untuk menyimpan konfigurasi baru.

Anda dapat memodifikasi properti berikut:

## Model penyediaan

Untuk mengubah model penyediaan, pilih Ubah model penyediaan dan lakukan salah satu hal berikut:

- Untuk menggunakan armada sesuai permintaan yang dikelola oleh AWS CodeBuild, pilih On-Demand. Dengan armada sesuai permintaan, CodeBuild menyediakan komputasi untuk build Anda. Mesin-mesin dihancurkan saat build selesai. Armada sesuai permintaan dikelola sepenuhnya, dan mencakup kemampuan penskalaan otomatis untuk menangani lonjakan permintaan.
- Untuk menggunakan armada kapasitas cadangan yang dikelola oleh AWS CodeBuild, pilih Kapasitas cadangan, lalu pilih nama Armada. Dengan armada kapasitas cadangan, Anda mengonfigurasi satu set instans khusus untuk lingkungan build Anda. Mesin ini tetap menganggur, siap untuk memproses build atau pengujian segera dan mengurangi durasi build. Dengan armada kapasitas cadangan, mesin Anda selalu berjalan dan akan terus mengeluarkan biaya selama disediakan.

Untuk informasi, lihat [Bekerja dengan kapasitas cadangan di AWS CodeBuild](#).

### Citra lingkungan

Untuk mengubah image build, pilih Override image dan lakukan salah satu hal berikut:

- Untuk menggunakan image Docker yang dikelola oleh AWS CodeBuild, pilih Managed image, lalu buat pilihan dari Sistem operasi, Runtime, Image, dan versi Image. Buat pilihan dari jenis Lingkungan jika tersedia.
- Untuk menggunakan gambar Docker lain, pilih Custom image. Untuk jenis Lingkungan, pilih ARM, Linux, GPU Linux, atau Windows. Jika Anda memilih Registri lain, untuk URL registri eksternal, masukkan nama dan tag gambar Docker di Docker Hub, menggunakan format *docker repository/docker image name*. Jika Anda memilih Amazon ECR, gunakan repositori Amazon ECR dan gambar Amazon ECR untuk memilih gambar Docker di akun Anda. AWS
- Untuk menggunakan image Docker pribadi, pilih Custom image. Untuk jenis Lingkungan, pilih ARM, Linux, GPU Linux, atau Windows. Untuk registri Gambar, pilih Registri lain, lalu masukkan ARN kredensial untuk gambar Docker pribadi Anda. Kredensialnya harus dibuat oleh Secrets Manager. Untuk informasi lebih lanjut, lihat [Apa itu AWS Secrets Manager?](#) di Panduan Pengguna AWS Secrets Manager.

#### Note

CodeBuild mengganti ENTRYPOINT untuk gambar Docker khusus.

## Peran layanan

Lakukan salah satu dari cara berikut:

- Jika Anda tidak memiliki peran CodeBuild layanan, pilih Peran layanan baru. Di Nama peran, masukkan nama untuk peran baru.
- Jika Anda memiliki peran CodeBuild layanan, pilih Peran layanan yang ada. Di Peran ARN, pilih peran layanan.

### Note

Saat menggunakan konsol untuk membuat proyek build, Anda dapat membuat peran CodeBuild layanan secara bersamaan. Secara default, peran hanya berfungsi dengan proyek build tersebut. Jika Anda menggunakan konsol untuk mengaitkan peran layanan ini dengan proyek build lain, peran akan diperbarui agar berfungsi dengan proyek build lainnya. Peran layanan dapat bekerja dengan hingga 10 proyek pembangunan.

## Konfigurasi tambahan

### Batas waktu

Tentukan nilai, antara 5 menit dan 8 jam, setelah itu CodeBuild menghentikan build jika tidak selesai. Jika jam dan menit dibiarkan kosong, nilai default 60 menit digunakan.

### Diistimewakan

Pilih Aktifkan bendera ini jika Anda ingin membuat gambar Docker atau ingin build Anda mendapatkan hak istimewa yang lebih tinggi. Hanya jika Anda berencana menggunakan proyek build ini untuk membangun gambar Docker. Jika tidak, semua build terkait yang mencoba berinteraksi dengan daemon Docker gagal. Anda juga harus memulai daemon Docker sehingga build Anda dapat berinteraksi dengannya. Salah satu cara untuk melakukannya adalah dengan menginisialisasi daemon Docker di `install` fase spesifikasi build Anda dengan menjalankan perintah build berikut. Jangan jalankan perintah ini jika Anda memilih image lingkungan build yang disediakan oleh CodeBuild dengan dukungan Docker.

### Note

Secara default, daemon Docker diaktifkan untuk build non-VPC. Jika Anda ingin menggunakan kontainer Docker untuk build VPC, [lihat Runtime Privilege dan](#)

[Kemampuan Linux di situs web Docker Docs dan](#) aktifkan mode istimewa. Juga, Windows tidak mendukung mode istimewa.

```
- nohup /usr/local/bin/dockerd --host=unix:///var/run/docker.sock --
host=tcp://127.0.0.1:2375 --storage-driver=overlay2 &
- timeout 15 sh -c "until docker info; do echo .; sleep 1; done"
```

## VPC

Jika Anda CodeBuild ingin bekerja dengan VPC Anda:

- Untuk VPC, pilih ID VPC yang digunakan. CodeBuild
- Untuk Subnet VPC, pilih subnet yang menyertakan sumber daya yang digunakan. CodeBuild
- Untuk grup Keamanan VPC, pilih grup keamanan yang CodeBuild digunakan untuk memungkinkan akses ke sumber daya di VPC.

Untuk informasi selengkapnya, lihat [GunakanAWS CodeBuilddengan Amazon Virtual Private Cloud](#).

## Hitung

Pilih salah satu opsi yang tersedia.

## Variabel lingkungan

Masukkan nama dan nilai, lalu pilih jenis setiap variabel lingkungan untuk build yang akan digunakan.

### Note

CodeBuild menetapkan variabel lingkungan untuk AWS Wilayah Anda secara otomatis. Anda harus menyetel variabel lingkungan berikut jika Anda belum menambahkannya ke buildspec.ymlmu:

- AWS\_ACCOUNT\_ID
- IMAGE\_REPO\_NAME
- IMAGE\_TAG

Konsol dan AWS CLI pengguna dapat melihat variabel lingkungan. Jika Anda tidak memiliki kekhawatiran tentang visibilitas variabel lingkungan Anda, atur bidang Nama dan Nilai, lalu setel Type ke Plaintext.

Kami menyarankan Anda menyimpan variabel lingkungan dengan nilai sensitif, seperti ID kunci AWS akses, kunci akses AWS rahasia, atau kata sandi sebagai parameter di Amazon EC2 Systems Manager Parameter Store atau AWS Secrets Manager.

Jika Anda menggunakan Amazon EC2 Systems Manager Parameter Store, lalu untuk Type, pilih Parameter. Untuk Nama, masukkan pengenal CodeBuild untuk referensi. Untuk Nilai, masukkan nama parameter seperti yang disimpan di Amazon EC2 Systems Manager Parameter Store. Menggunakan parameter bernama `/CodeBuild/dockerLoginPassword` sebagai contoh, untuk Type, pilih Parameter. Untuk Nama, masukkan `LOGIN_PASSWORD`. Untuk Nilai, masukkan `/CodeBuild/dockerLoginPassword`.

#### Important

Jika Anda menggunakan Amazon EC2 Systems Manager Parameter Store, sebaiknya Anda menyimpan parameter dengan nama parameter yang dimulai `/CodeBuild/` dengan (misalnya `/CodeBuild/dockerLoginPassword`). Anda dapat menggunakan CodeBuild konsol untuk membuat parameter di Amazon EC2 Systems Manager. Pilih Buat parameter, lalu ikuti instruksi di kotak dialog. (Di kotak dialog itu, untuk kunci KMS, Anda dapat menentukan ARN kunci di AWS KMS akun Anda. Amazon EC2 Systems Manager menggunakan kunci ini untuk mengenkripsi nilai parameter selama penyimpanan dan mendekripsi selama pengambilan.) Jika Anda menggunakan CodeBuild konsol untuk membuat parameter, konsol memulai nama parameter dengan `/CodeBuild/` saat sedang disimpan. Untuk informasi selengkapnya, lihat [Systems Manager Parameter Store](#) dan [Systems Manager Parameter Store Console Walkthrough](#) di Panduan Pengguna Amazon EC2 Systems Manager.

Jika project build Anda mengacu pada parameter yang disimpan di Amazon EC2 Systems Manager Parameter Store, peran layanan project build harus mengizinkan tindakan `ssm:GetParameters` tersebut. Jika Anda memilih Peran layanan baru lebih awal, CodeBuild sertakan tindakan ini dalam peran layanan default untuk proyek build Anda. Namun, jika Anda memilih Peran layanan yang ada, Anda harus menyertakan tindakan ini ke peran layanan Anda secara terpisah.

Jika project build Anda merujuk ke parameter yang disimpan di Amazon EC2 Systems Manager Parameter Store dengan nama parameter yang tidak dimulai/

CodeBuild/, dan Anda memilih Peran layanan baru, Anda harus memperbarui peran layanan tersebut untuk mengizinkan akses ke nama parameter yang tidak dimulai/CodeBuild/. Ini karena peran layanan itu memungkinkan akses hanya ke nama parameter yang dimulai dengan/CodeBuild/.

Jika Anda memilih Peran layanan baru, peran layanan menyertakan izin untuk mendekripsi semua parameter di bawah /CodeBuild/ namespace di Amazon EC2 Systems Manager Parameter Store.

Variabel lingkungan yang Anda tetapkan menggantikan variabel lingkungan yang ada. Misalnya, jika image Docker sudah berisi variabel lingkungan bernama MY\_VAR dengan nilai `my_value`, dan Anda menetapkan variabel lingkungan bernama MY\_VAR dengan nilai `other_value`, `my_value` maka diganti dengan `other_value`. Demikian pula, jika image Docker sudah berisi variabel lingkungan bernama PATH dengan nilai `/usr/local/sbin:/usr/local/bin`, dan Anda menetapkan variabel lingkungan bernama PATH dengan nilai `$PATH:/usr/share/ant/bin,/usr/local/sbin:/usr/local/bin` maka digantikan oleh nilai `$PATH:/usr/share/ant/bin` literal. Jangan mengatur variabel lingkungan apa pun dengan nama yang dimulai dengan `CODEBUILD_`. Awalan ini dicadangkan untuk penggunaan internal .

Jika variabel lingkungan dengan nama yang sama didefinisikan di beberapa tempat, nilainya ditentukan sebagai berikut:

- Nilai dalam panggilan operasi start build lebih diutamakan.
- Nilai dalam definisi proyek build akan diutamakan berikutnya.
- Nilai dalam deklarasi buildspec diutamakan paling rendah.

Jika Anda menggunakan Secrets Manager, untuk Type, pilih Secrets Manager. Untuk Nama, masukkan pengenal CodeBuild untuk referensi. Untuk Nilai, masukkan `reference-key` menggunakan pola `secret-id:json-key:version-stage:version-id`. Untuk informasi, lihat [Secrets Manager reference-key in the buildspec file](#).

#### Important

Jika Anda menggunakan Secrets Manager, kami sarankan Anda menyimpan rahasia dengan nama yang dimulai dengan /CodeBuild/ (misalnya, /CodeBuild/dockerLoginPassword). Untuk informasi lebih lanjut, lihat [Apa itu AWS Secrets Manager?](#) di Panduan Pengguna AWS Secrets Manager.



Jika proyek build Anda mengacu pada rahasia yang disimpan di Secrets Manager, peran layanan proyek build harus mengizinkan `secretsmanager:GetSecretValue` tindakan tersebut. Jika Anda memilih Peran layanan baru lebih awal, CodeBuild sertakan tindakan ini dalam peran layanan default untuk proyek build Anda. Namun, jika Anda memilih Peran layanan yang ada, Anda harus menyertakan tindakan ini ke peran layanan Anda secara terpisah.

Jika proyek build mengacu pada rahasia yang disimpan di Secrets Manager dengan nama rahasia yang tidak dimulai/`CodeBuild/`, dan Anda memilih Peran layanan baru, Anda harus memperbarui peran layanan untuk mengizinkan akses ke nama rahasia yang tidak dimulai/`CodeBuild/`. Ini karena peran layanan memungkinkan akses hanya ke nama rahasia yang dimulai dengan/`CodeBuild/`.

Jika Anda memilih Peran layanan baru, peran layanan menyertakan izin untuk mendekripsi semua rahasia di bawah `/CodeBuild/` namespace di Secrets Manager.

## Buildspec

Di bagian Buildspec, pilih Edit. Ketika perubahan Anda selesai, pilih Perbarui konfigurasi untuk menyimpan konfigurasi baru.

Anda dapat memodifikasi properti berikut:

### Membangun spesifikasi

Lakukan salah satu dari cara berikut:

- Jika kode sumber Anda menyertakan file buildspec, pilih Gunakan file buildspec. Secara default, CodeBuild mencari file bernama `buildspec.yml` di direktori root kode sumber. Jika file buildspec Anda menggunakan nama atau lokasi yang berbeda, masukkan jalurnya dari root sumber dalam nama Buildspec (misalnya, `atau. buildspec-two.yml configuration/buildspec.yml`). Jika file buildspec ada di bucket S3, file tersebut harus berada di AWS Region yang sama dengan project build Anda. Tentukan file buildspec menggunakan ARN (misalnya, `arn:aws:s3:::<my-codebuild-sample2>/buildspec.yml`).
- Jika kode sumber Anda tidak menyertakan file buildspec, atau jika Anda ingin menjalankan perintah build yang berbeda dari yang ditentukan untuk build fase dalam `buildspec.yml` file di direktori root kode sumber, pilih Sisipkan perintah build. Untuk perintah Build, masukkan perintah yang ingin Anda jalankan dalam build fase. Untuk beberapa perintah, pisahkan setiap perintah dengan `&&` (misalnya, `mvn test && mvn package`). Untuk menjalankan perintah di fase lain, atau jika Anda memiliki daftar panjang perintah untuk build fase tersebut,

tambahkan `buildspec.yml` file ke direktori root kode sumber, tambahkan perintah ke file, lalu pilih Gunakan `buildspec.yml` di direktori root kode sumber.

Lihat informasi yang lebih lengkap di [Referensi Buildspec](#).

## Konfigurasi Batch

Di bagian konfigurasi Batch, pilih Edit. Ketika perubahan Anda selesai, pilih Perbarui konfigurasi untuk menyimpan konfigurasi baru. Untuk informasi selengkapnya, lihat [Batch dibangun di AWS CodeBuild](#).

Anda dapat memodifikasi properti berikut:

### Peran layanan Batch

Menyediakan peran layanan untuk build batch.

Pilih salah satu cara berikut:

- Jika Anda tidak memiliki peran layanan batch, pilih Peran layanan baru. Di peran Layanan, masukkan nama untuk peran baru.
- Jika Anda memiliki peran layanan batch, pilih Peran layanan yang ada. Dalam Peran layanan, pilih peran layanan.

Batch build memperkenalkan peran keamanan baru dalam konfigurasi batch. Peran baru ini diperlukan karena CodeBuild harus dapat memanggil `StartBuild`, `StopBuild`, dan `RetryBuild` tindakan atas nama Anda untuk menjalankan build sebagai bagian dari batch. Pelanggan harus menggunakan peran baru, dan bukan peran yang sama yang mereka gunakan dalam build mereka, karena dua alasan:

- Memberikan peran `StartBuild`, `StopBuild`, dan `RetryBuild` izin akan memungkinkan satu build untuk memulai lebih banyak build melalui `buildspec`.
- CodeBuild build batch memberikan batasan yang membatasi jumlah build dan tipe komputasi yang dapat digunakan untuk build dalam batch. Jika peran build memiliki izin ini, ada kemungkinan build itu sendiri dapat melewati batasan ini.

### Jenis komputasi yang diizinkan untuk batch

Pilih jenis komputasi yang diizinkan untuk batch. Pilih semua yang berlaku.

## Build maksimum yang diizinkan dalam batch

Masukkan jumlah maksimum build yang diizinkan dalam batch. Jika batch melebihi batas ini, batch akan gagal.

## Batas waktu Batch

Masukkan jumlah waktu maksimum untuk menyelesaikan batch build.

## Gabungkan artefak

Pilih Gabungkan semua artefak dari batch ke satu lokasi agar semua artefak dari batch digabungkan menjadi satu lokasi.

## Mode laporan Batch

Pilih mode laporan status build yang diinginkan untuk build batch.

### Note

Bidang ini hanya tersedia jika sumber proyek adalah Bitbucket, GitHub, atau GitHub Enterprise, dan Laporkan status build ke penyedia sumber saat build Anda mulai dan selesai dipilih di bawah Sumber.

## Build agregat

Pilih agar status untuk semua build dalam batch digabungkan menjadi satu laporan status.

## Membangun individu

Pilih agar status build untuk semua build dalam batch dilaporkan secara terpisah.

## Artifacts

Di bagian Artefak, pilih Edit. Ketika perubahan Anda selesai, pilih Perbarui konfigurasi untuk menyimpan konfigurasi baru.

Anda dapat memodifikasi properti berikut:

### Jenis

Lakukan salah satu dari cara berikut:

- Jika Anda tidak ingin membuat artefak keluaran build apa pun, pilih Tidak ada artefak. Anda mungkin ingin melakukan ini jika Anda hanya menjalankan pengujian build atau Anda ingin mendorong image Docker ke repositori Amazon ECR.
- Untuk menyimpan output build dalam bucket S3, pilih Amazon S3, lalu lakukan hal berikut:
  - Jika Anda ingin menggunakan nama proyek Anda untuk file atau folder ZIP keluaran build, biarkan Nama kosong. Jika tidak, masukkan nama. (Jika Anda ingin mengeluarkan file ZIP, dan Anda ingin file ZIP memiliki ekstensi file, pastikan untuk memasukkannya setelah nama file ZIP.)
  - Pilih Aktifkan versi semantik jika Anda ingin nama yang ditentukan dalam file buildspec untuk mengganti nama apa pun yang ditentukan di konsol. Nama dalam file buildspec dihitung pada waktu pembuatan dan menggunakan bahasa perintah Shell. Misalnya, Anda dapat menambahkan tanggal dan waktu untuk nama artefak Anda sehingga selalu unik. Nama artefak unik mencegah artefak ditimpa. Untuk informasi selengkapnya, lihat [Sintaks Buildspec](#).
  - Untuk nama Bucket, pilih nama bucket keluaran.
  - Jika Anda memilih Sisipkan perintah build sebelumnya dalam prosedur ini, maka untuk file Output, masukkan lokasi file dari build yang ingin Anda masukkan ke dalam file atau folder ZIP keluaran build. Untuk beberapa lokasi, pisahkan setiap lokasi dengan koma (misalnya, `appspec.yml, target/my-app.jar`). Untuk informasi lebih lanjut, lihat deskripsi files di [Sintaks Buildspec](#).
  - Jika Anda tidak ingin artefak build Anda dienkripsi, pilih Hapus enkripsi artefak.

Untuk setiap set artefak sekunder yang Anda inginkan:

1. Untuk pengenalan Artifact, masukkan nilai yang kurang dari 128 karakter dan hanya berisi karakter alfanumerik dan garis bawah.
2. Pilih Tambahkan artefak.
3. Ikuti langkah-langkah sebelumnya untuk mengonfigurasi artefak sekunder Anda.
4. Pilih Simpan artefak.

## Konfigurasi tambahan

### Kunci enkripsi

Lakukan salah satu dari cara berikut:

- Untuk menggunakan Kunci yang dikelola AWS Amazon S3 di akun Anda untuk mengenkripsi artefak keluaran build, biarkan kunci Enkripsi kosong. Ini menjadi opsi default.
- Untuk menggunakan kunci yang dikelola pelanggan untuk mengenkripsi artefak keluaran build, di kunci Enkripsi, masukkan ARN kunci yang dikelola pelanggan. Gunakan format `arn:aws:kms:region-ID:account-ID:key/key-ID`.

## Jenis cache

Untuk jenis Cache, pilih salah satu dari berikut ini:

- Jika Anda tidak ingin menggunakan cache, pilih Tidak ada cache.
- Jika Anda ingin menggunakan cache Amazon S3, pilih Amazon S3, lalu lakukan hal berikut:
  - Untuk Bucket, pilih nama bucket S3 tempat cache disimpan.
  - (Opsional) Untuk awalan jalur Cache, masukkan awalan jalur Amazon S3. Nilai awalan jalur Cache mirip dengan nama direktori. Itu memungkinkan Anda untuk menyimpan cache di bawah direktori yang sama dalam ember.

### Important

Jangan menambahkan garis miring (/) ke ujung awalan jalur.

- Jika Anda ingin menggunakan cache lokal, pilih Lokal, lalu pilih satu atau beberapa mode cache lokal.

### Note

Mode cache lapisan Docker hanya tersedia untuk Linux. Jika Anda memilikinya, proyek Anda harus berjalan dalam mode istimewa.

Menggunakan cache menghemat banyak waktu pembuatan karena bagian lingkungan build yang dapat digunakan kembali disimpan dalam cache dan digunakan di seluruh build. Untuk informasi tentang menentukan cache dalam file buildspec, lihat [Sintaks Buildspec](#) Untuk informasi selengkapnya tentang caching, lihat [Membangun caching di AWS CodeBuild](#).

## Log

Di bagian Log, pilih Edit. Ketika perubahan Anda selesai, pilih Perbarui konfigurasi untuk menyimpan konfigurasi baru.

Anda dapat memodifikasi properti berikut:

Pilih log yang ingin Anda buat. Anda dapat membuat CloudWatch Log Amazon, log Amazon S3, atau keduanya.

### CloudWatch

Jika Anda ingin CloudWatch log Amazon Log:

CloudWatch log

Pilih CloudWatch log.

Nama grup

Masukkan nama grup CloudWatch log Amazon Logs Anda.

Nama aliran

Masukkan nama aliran CloudWatch log Amazon Logs Anda.

### S3

Jika Anda ingin log Amazon S3:

Log S3

Pilih log S3.

Bucket

Pilih nama bucket S3 untuk log Anda.

Awalan jalur

Masukkan awalan untuk log Anda.

Nonaktifkan enkripsi log S3

Pilih jika Anda tidak ingin log S3 Anda dienkripsi.

## Mengubah setelan proyek pembangunan (AWS CLI)

Untuk informasi tentang penggunaan AWS CLI bersama AWS CodeBuild, lihat [Referensi baris perintah](#).

Untuk memperbarui CodeBuild proyek dengan AWS CLI, Anda membuat file JSON dengan properti yang diperbarui dan meneruskan file itu ke [update-project](#) perintah. Properti apa pun yang tidak terdapat dalam file pembaruan tetap tidak berubah.

Dalam file update JSON, hanya `name` properti dan properti yang dimodifikasi diperlukan. Yang `name` properti mengidentifikasi proyek untuk memodifikasi. Untuk setiap struktur yang dimodifikasi, parameter yang diperlukan untuk struktur tersebut juga harus disertakan. Misalnya, untuk memodifikasi lingkungan untuk proyek, `environment/type` dan `environment/computeType` properti diperlukan. Berikut adalah contoh yang memperbarui citra lingkungan:

```
{
 "name": "<project-name>",
 "environment": {
 "type": "LINUX_CONTAINER",
 "computeType": "BUILD_GENERAL1_SMALL",
 "image": "aws/codebuild/amazonlinux2-x86_64-standard:4.0"
 }
}
```

Jika Anda perlu untuk mendapatkan nilai properti saat ini untuk proyek, menggunakan [batch-get-projects](#) perintah untuk mendapatkan properti saat proyek Anda memodifikasi, dan menulis output ke file.

```
aws codebuild batch-get-projects --names "<project-name>" > project-info.json
```

Yang `project-info.json` file berisi array proyek, sehingga tidak dapat digunakan secara langsung untuk memperbarui proyek. Namun, Anda dapat menyalin properti yang ingin Anda modifikasi dari `project-info.json` file dan paste ke file update Anda sebagai dasar untuk properti yang ingin Anda modifikasi. Untuk informasi selengkapnya, lihat [Lihat detail proyek build \(AWS CLI\)](#).

Memodifikasi file JSON pembaruan seperti yang dijelaskan dalam [Buat proyek build \(AWS CLI\)](#), dan simpan hasil Anda. Setelah selesai memodifikasi file JSON pemutakhiran, jalankan [update-project](#) perintah, melewati file update JSON.

```
aws codebuild update-project --cli-input-json file://<update-project-file>
```

Jika berhasil, JSON proyek yang diperbarui muncul di output. Jika ada parameter yang diperlukan yang hilang, pesan kesalahan ditampilkan dalam output yang mengidentifikasi parameter yang hilang. Misalnya, ini adalah pesan kesalahan yang ditampilkan jika `environment/type` parameter hilang:

```
aws codebuild update-project --cli-input-json file://update-project.json
```

```
Parameter validation failed:
Missing required parameter in environment: "type"
```

## Mengubah pengaturan proyek build (AWSSDK)

Untuk informasi tentang penggunaan AWS CodeBuild dengan AWSSDK, lihat [AWS Referensi SDK dan alat](#).

## Menghapus proyek pembangunan di AWS CodeBuild

Anda dapat menggunakan konsol CodeBuild, AWS CLI, atau AWSSDK untuk menghapus proyek build di CodeBuild. Jika Anda menghapus proyek, build-nya tidak dihapus.

### Warning

Anda tidak dapat menghapus proyek yang telah membangun dan kebijakan sumber daya. Untuk menghapus proyek dengan kebijakan sumber daya dan build, Anda harus terlebih dahulu menghapus kebijakan sumber daya dan menghapus membangunnya.

### Topik

- [Menghapus proyek pembangunan \(konsol\)](#)
- [Menghapus proyek pembangunan \(AWS CLI\)](#)
- [Menghapus proyek pembangunan \(AWSSDK\)](#)

## Menghapus proyek pembangunan (konsol)

1. Buka AWS CodeBuild konsol di <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Di panel navigasi, pilih **Membuat proyek**.
3. Lakukan salah satu dari berikut:
  - Pilih tombol radio di samping proyek pembangunan yang ingin Anda hapus, lalu pilih **Hapus**.
  - Pilih proyek pembangunan yang ingin Anda hapus, lalu pilih **Hapus**.



**Note**

Secara default, hanya 10 proyek build terbaru yang ditampilkan. Untuk melihat lebih banyak proyek build, pilih nilai yang berbeda `Proyek` per halaman atau gunakan panah belakang dan depan untuk melihat proyek.

## Menghapus proyek pembangunan (AWS CLI)

1. Jalankan perintah `delete-project`:

```
aws codebuild delete-project --name nama
```

Ganti placeholder berikut:

- *nama*: String yang dibutuhkan. Nama proyek pembangunan yang akan dihapus. Untuk mendapatkan daftar proyek build yang tersedia, jalankan `list-projects` perintah. Untuk informasi selengkapnya, lihat [Melihat daftar nama proyek build \(AWS CLI\)](#).
2. Jika berhasil, tidak ada data dan tidak ada kesalahan yang muncul dalam output.

Untuk informasi lebih lanjut tentang penggunaan AWS CLI bersama AWS CodeBuild, lihat [Referensi basis perintah](#).

## Menghapus proyek pembangunan (AWS SDK)

Untuk informasi lebih lanjut tentang penggunaan AWS CodeBuild dengan AWS SDK, lihat [AWS Referensi SDK dan alat](#).

## Bekerja dengan proyek bersama

Berbagi proyek memungkinkan pemilik proyek untuk berbagi AWS CodeBuild proyek dengan AWS akun atau pengguna. Dalam model ini, akun yang memiliki proyek (pemilik) berbagi proyek dengan akun lain (konsumen). Konsumen tidak dapat mengedit atau menjalankan proyek.

Isi

- [Prasyarat untuk berbagi proyek](#)

- [Prasyarat untuk mengakses proyek bersama yang dibagikan kepada Anda](#)
- [Layanan terkait](#)
- [Berbagi proyek](#)
- [Membatalkan berbagi proyek bersama](#)
- [Mengidentifikasi proyek bersama](#)
- [Izin proyek bersama](#)

## Prasyarat untuk berbagi proyek

Untuk berbagi proyek, AWS Akun harus memilikinya. Anda tidak dapat membagikan proyek yang telah dibagikan dengan Anda.

## Prasyarat untuk mengakses proyek bersama yang dibagikan kepada Anda

Untuk mengakses proyek bersama, peran IAM konsumen memerlukan `BatchGetProjects` izin. Anda dapat memberlakukan kebijakan berikut ke peran IAM mereka:

```
{
 "Effect": "Allow",
 "Resource": [
 "*"
],
 "Action": [
 "codebuild:BatchGetProjects"
]
}
```

Untuk informasi selengkapnya, lihat [Menggunakan kebijakan berbasis identitas untuk AWS CodeBuild](#).

## Layanan terkait

Berbagi proyek terintegrasi dengan AWS Resource Access Manager (AWS RAM), layanan yang memungkinkan Anda untuk berbagi AWS sumber daya dengan AWS Akun atau melalui AWS Organizations. Dengan AWS RAM, Anda berbagi sumber daya dengan membuat berbagi sumber daya yang menentukan sumber daya dan konsumen untuk berbagi dengan mereka. Konsumen bisa bersifat individual AWS Akun, unit organisasi di AWS Organizations, atau seluruh organisasi di AWS Organizations.

Untuk informasi selengkapnya, lihat Panduan Pengguna [AWS RAM](#).

## Berbagi proyek

Konsumen dapat menggunakan kedua AWS CLI dan AWS CodeBuild konsol untuk melihat proyek dan membangun yang telah Anda bagikan. Konsumen tidak dapat mengedit atau menjalankan proyek.

Anda dapat menambahkan proyek untuk berbagi sumber daya yang ada atau Anda dapat membuat satu di [AWS RAM konsol](#).

### Note

Anda tidak dapat menghapus proyek dengan build yang telah ditambahkan ke bagian sumber daya.

Untuk berbagi proyek dengan unit organisasi atau seluruh organisasi, Anda harus mengaktifkan berbagi dengan AWS Organizations. Untuk informasi lebih lanjut, lihat [Aktifkan pembagian dengan AWS Organizations](#) dalam Panduan Pengguna AWS RAM .

Anda dapat menggunakan AWS CodeBuild konsol, AWS RAM konsol, atau AWS CLI untuk berbagi proyek yang Anda miliki.

Untuk berbagi proyek yang Anda miliki (CodeBuild konsol)

1. Buka AWS CodeBuild konsol di <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Di panel navigasi, pilih Proyek pembangunan.

### Note

Secara default, hanya 10 proyek build terbaru yang ditampilkan. Untuk melihat proyek pembangunan lainnya, pilih ikon roda gigi, lalu pilih nilai yang berbeda Proyek per halaman atau gunakan panah belakang dan depan.

3. Pilih proyek yang ingin Anda bagikan, lalu pilih Bagikan. Untuk informasi selengkapnya, lihat [Membuat berbagi sumber daya](#) di AWS RAM Panduan Pengguna.

Untuk berbagi proyek yang Anda miliki (AWS RAM konsol)

Lihat [Membuat berbagi sumber daya](#) di AWS RAM Panduan Pengguna.

Untuk berbagi proyek yang Anda miliki (AWS RAMperintah)

Gunakan perintah [create-resource-share](#).

Untuk berbagi proyek yang Anda miliki (perintah CodeBuild)

Menggunakan [kebijakan-put-resource](#) perintah:

1. Membuat file bernamapolicy .json dan salin yang berikut ke dalamnya.

```
{
 "Version": "2012-10-17",
 "Statement": [{
 "Effect": "Allow",
 "Principal": {
 "AWS": "<consumer-aws-account-id-or-user>"
 },
 "Action": [
 "codebuild:BatchGetProjects",
 "codebuild:BatchGetBuilds",
 "codebuild:ListBuildsForProject"],
 "Resource": "<arn-of-project-to-share>"
]
}
```

2. Perbaruipolicy .json dengan proyek ARN dan pengidentifikasi untuk berbagi dengan. Contoh berikut memberikan akses hanya-baca ke pengguna root untuk AWS akun diidentifikasi oleh 123456789012.

```
{
 "Version": "2012-10-17",
 "Statement": [{
 "Effect": "Allow",
 "Principal": {
 "AWS": [
 "123456789012"
]
 },
 "Action": [
 "codebuild:BatchGetProjects",
 "codebuild:BatchGetBuilds",
 "codebuild:ListBuildsForProject"],
 "Resource": "arn:aws:codebuild:us-west-2:123456789012:project/my-project"
]
}
```

```
 }]
}
```

3. Jalankan [kebijakan-put-resource](#) perintah.

```
aws codebuild put-resource-policy --resource-arn <project-arn> --policy file://
policy.json
```

4. Dapatkan AWS RAM berbagi sumber daya ARN.

```
aws ram list-resources --resource-owner SELF --resource-arns <project-arn>
```

Ini akan menghasilkan respons yang serupa dengan ini:

```
{
 "resources": [
 {
 "arn": "<project-arn>",
 "type": "<type>",
 "resourceShareArn": "<resource-share-arn>",
 "creationTime": "<creation-time>",
 "lastUpdatedTime": "<last-update-time>"
 }
]
}
```

Dari respon, salin *<resource-share-arn>* Nilai untuk digunakan pada langkah berikutnya.

5. Jalankan AWS RAM [mempromosikan-sumber daya-bagian-dibuat-dari-kebijakan](#) perintah.

```
aws ram promote-resource-share-created-from-policy --resource-share-arn <resource-
share-arn>
```

## Membatalkan berbagi proyek bersama

Proyek yang tidak dibagikan, termasuk membangunnya, hanya dapat diakses oleh pemiliknya. Jika Anda membatalkan berbagi proyek, ada AWS akun atau pengguna yang sebelumnya Anda bagikan tidak dapat mengakses proyek atau membangunnya.

Untuk membatalkan berbagi proyek bersama yang Anda miliki, Anda harus menghapusnya dari berbagi sumber daya. Anda dapat menggunakan AWS CodeBuild konsol, AWS RAM konsol, atau AWS CLI untuk melakukan ini.

Untuk membatalkan berbagi proyek bersama yang Anda miliki (AWS RAM konsol)

Lihat [Memperbarui Pembagian Sumber Daya](#) di Panduan Pengguna AWS RAM

Untuk membatalkan berbagi proyek bersama yang Anda miliki (AWS CLI)

Gunakan perintah [disassociate-resource-share](#).

Untuk unshare proyek yang Anda miliki (perintah CodeBuild)

Jalankan [hapus-kebijakan](#) perintah dan menentukan ARN proyek Anda ingin unshare:

```
aws codebuild delete-resource-policy --resource-arn project-arn
```

## Mengidentifikasi proyek bersama

Pemilik dan konsumen dapat menggunakan AWS CLI untuk mengidentifikasi proyek bersama.

Untuk mengidentifikasi proyek yang dibagikan dengan AWS Akun atau pengguna (AWS CLI)

Menggunakan [daftar-dibagikan proyek](#) perintah untuk mengembalikan proyek-proyek yang dibagikan dengan Anda.

## Izin proyek bersama

Izin untuk pemilik

Pemilik proyek dapat mengedit proyek dan menggunakannya untuk menjalankan build.

Izin untuk konsumen

Konsumen proyek dapat melihat proyek dan membangunnya, tetapi tidak dapat mengedit proyek atau menggunakannya untuk menjalankan build.

## Proyek penandaan di AWS CodeBuild

Tanda adalah label atribut khusus yang Anda atau AWS tetapkan ke sumber daya AWS. Setiap tanda AWS memiliki dua bagian:

- Sebuah kunci tag (misalnya, `CostCenter`, `Environment`, `Project`, atau `Secret`). Kunci tanda peka terhadap huruf besar dan kecil.
- Bidang opsional yang dikenal sebagai nilai tag (misalnya, `111122223333`, `Production`, atau nama tim). Mengabaikan nilai tag sama dengan menggunakan rangkaian kosong. Seperti kunci tanda, nilai tanda peka huruf besar dan kecil.

Bersama-sama ini dikenal sebagai pasangan nilai-kunci. Untuk informasi tentang jumlah tanda yang dapat Anda miliki di proyek dan batasan pada kunci dan nilai tag, lihat [Tanda](#).

Tag membantu Anda mengidentifikasi dan mengatur sumber daya AWS. Banyak layanan AWS yang mendukung penandaan, sehingga Anda dapat menetapkan tag yang sama ke sumber daya dari layanan yang berbeda untuk menunjukkan bahwa sumber daya tersebut terkait. Misalnya, Anda dapat menugaskan tag yang sama ke CodeBuild proyek yang Anda tetapkan ke sebuah bucket S3. Untuk informasi selengkapnya tentang menggunakan tanda, lihat [Menandai Praktik Terbaik](#).

Di CodeBuild, sumber daya utama adalah proyek dan kelompok laporan. Anda dapat menggunakan CodeBuild konsol, CodeBuild API, AWS CLI, atau AWS SDK untuk menambah, mengelola, dan menghapus tag untuk proyek. Selain mengidentifikasi, mengatur, dan melacak proyek Anda dengan tanda, Anda dapat menggunakan tanda dalam kebijakan IAM untuk membantu mengontrol siapa yang dapat melihat dan berinteraksi dengan proyek Anda. Untuk contoh kebijakan akses berbasis tag, lihat [Menggunakan tag untuk mengontrol akses ke AWS CodeBuild sumber daya](#).

## Topik

- [Tambahkan tag ke proyek](#)
- [Melihat tanda untuk proyek](#)
- [Edit tag untuk proyek](#)
- [Hapus tag dari proyek](#)

## Tambahkan tag ke proyek

Menambahkan tanda ke proyek dapat membantu Anda mengidentifikasi dan mengatur AWS sumber daya serta mengelola akses ke mereka. Pertama, Anda menambahkan satu atau lebih tag (pasangan nilai-kunci) ke proyek. Perlu diingat bahwa ada batasan pada jumlah tanda yang dapat Anda miliki di proyek. Ada pembatasan pada karakter yang dapat Anda gunakan di bidang kunci dan nilai. Untuk informasi selengkapnya, lihat [Tanda](#). Setelah Anda memiliki tanda, Anda dapat membuat kebijakan

IAM untuk mengelola akses ke proyek berdasarkan tag ini. Anda dapat menggunakan CodeBuild konsol atau AWS CLI untuk menambahkan tanda ke proyek.

**⚠ Important**

Sebelum Anda menambahkan tag ke proyek, pastikan untuk meninjau kebijakan IAM yang mungkin menggunakan tag untuk mengontrol akses ke sumber daya seperti proyek pembangunan. Untuk contoh kebijakan akses berbasis tag, lihat [Menggunakan tag untuk mengontrol akses ke AWS CodeBuild sumber daya](#).

Untuk informasi selengkapnya tentang menambahkan tanda ke proyek saat Anda membuatnya, lihat [Tambahkan tag ke proyek \(konsol\)](#).

### Topik

- [Tambahkan tag ke proyek \(konsol\)](#)
- [Tambahkan tag ke proyek \(AWS CLI\)](#)

### Tambahkan tag ke proyek (konsol)

Anda dapat menggunakan CodeBuild konsol untuk menambahkan satu atau lebih tag ke CodeBuild proyek.

1. Buka CodeBuild konsol di <https://console.aws.amazon.com/codebuild/>.
2. Dalam proyek Build, pilih nama proyek yang Anda ingin tambahkan tag.
3. Di panel navigasi, pilih Pengaturan. Pilih Membangun tag proyek.
4. Jika tidak ada tanda yang telah ditambahkan ke proyek, pilih Tambahkan tag. Jika tidak, pilih Edit, lalu pilih Tambahkan tanda.
5. Di Kunci, masukkan sebuah nama untuk tag tersebut. Anda dapat menambahkan nilai opsional untuk tag di Nilai.
6. (Opsional) Untuk menambahkan tag lain, pilih Tambahkan tag lagi.
7. Setelah Anda selesai menambahkan tag, pilih Kirim.



## Tambahkan tag ke proyek (AWS CLI)

Untuk menambahkan tag ke proyek saat Anda membuatnya, lihat [Buat proyek build \(AWS CLI\)](#). Di `create-project.json`, tambahkan tag Anda.

Dalam langkah-langkah ini, kami menganggap bahwa Anda telah menginstal versi terbaru dari AWS CLI atau diperbarui ke versi terkini. Untuk informasi lebih lanjut, lihat [Menginstal AWS Command Line Interface](#).

Jika berhasil, perintah ini tidak mengembalikan apa pun.

## Melihat tanda untuk proyek

Tag dapat membantu Anda mengidentifikasi dan mengatur sumber daya AWS Anda serta mengelola akses ke mereka. Untuk informasi lebih lanjut tentang penggunaan tanda, lihat laporan resmi [Praktik terbaik](#). Untuk contoh kebijakan akses berbasis tag, lihat [Menggunakan tag untuk mengontrol akses ke AWS CodeBuild sumber daya](#).

## Melihat tanda untuk proyek (konsol)

Anda dapat menggunakan CodeBuild konsol untuk melihat tanda yang terkait dengan CodeBuild proyek.

1. Buka CodeBuild konsol di <https://console.aws.amazon.com/codebuild/>.
2. Dalam proyek Build, pilih nama proyek yang Anda ingin tampilkan tag.
3. Di panel navigasi, pilih Pengaturan. Pilih Membangun tag proyek.

## Tampilkan tanda untuk proyek (AWS CLI)

Untuk menampilkan tag untuk proyek build, jalankan perintah berikut. Gunakan nama proyek Anda untuk `--names` parameter.

```
aws codebuild batch-get-projects --names your-project-name
```

Jika berhasil, perintah ini mengembalikan informasi berformat JSON tentang proyek build Anda yang menyertakan sesuatu seperti berikut ini:

```
{
```

```
"tags": {
 "Status": "Secret",
 "Team": "JanesProject"
}
```

Jika proyek tidak memiliki tag, tags bagian tersebut kosong:

```
"tags": []
```

## Edit tag untuk proyek

Anda dapat mengubah nilai untuk tag yang terkait dengan proyek. Anda juga dapat mengubah nama kunci, yang setara dengan menghapus tag saat ini dan menambahkan tag yang berbeda dengan nama baru dan nilai yang sama dengan kunci lainnya. Perlu diingat bahwa ada batasan pada karakter yang dapat Anda gunakan di bidang kunci dan nilai. Untuk informasi selengkapnya, lihat [Tanda](#).

### Important

Mengedit tag untuk proyek dapat memengaruhi akses ke proyek tersebut. Sebelum Anda mengedit nama (kunci) atau nilai tag untuk resensi, pastikan untuk meninjau kebijakan IAM yang mungkin menggunakan kunci atau nilai tag untuk mengontrol akses ke sumber daya seperti proyek build. Untuk contoh kebijakan akses berbasis tag, lihat [Menggunakan tag untuk mengontrol akses ke AWS CodeBuild sumber daya](#).

## Edit tag untuk proyek (konsol)

Anda dapat menggunakan CodeBuild konsol untuk mengedit tag yang terkait dengan CodeBuild proyek.

1. Buka CodeBuild konsol di <https://console.aws.amazon.com/codebuild/>.
2. Dalam proyek Build, pilih nama proyek tempat Anda ingin mengedit tag.
3. Di panel navigasi, pilih Pengaturan. Pilih Membangun tag proyek.
4. Pilih Edit.
5. Lakukan salah satu dari berikut:

- Untuk mengubah tag, masukkan nama baru di Kunci. Mengubah nama tag adalah setara dengan menghapus tag dan menambahkan tag baru dengan nama kunci baru.
  - Untuk mengubah nilai sebuah tag, masukkan nilai baru. Jika Anda ingin mengubah nilai menjadi nol, hapus nilai saat ini dan biarkan bidang kosong.
6. Setelah Anda selesai mengedit tag, pilih Kirim.

### Edit tag untuk proyek (AWS CLI)

Untuk menambah, mengubah, atau menghapus tag dari proyek pembangunan, lihat [Mengubah setelan proyek pembangunan \(AWS CLI\)](#). Perbarui tags bagian dalam data berformat JSON yang Anda gunakan untuk memperbarui proyek.

### Hapus tag dari proyek

Anda dapat menghapus satu atau lebih tag yang terkait dengan proyek. Menghapus tag tidak menghapus tag dari AWS sumber daya yang terkait dengan tag tersebut.

#### Important

Menghapus tag untuk proyek dapat memengaruhi akses ke proyek tersebut. Sebelum menghapus tag dari proyek, pastikan untuk meninjau kebijakan IAM yang mungkin menggunakan kunci atau nilai tag untuk mengontrol akses ke sumber daya seperti proyek pembangunan. Untuk contoh kebijakan akses berbasis tag, lihat [Menggunakan tag untuk mengontrol akses ke AWS CodeBuild sumber daya](#).

### Hapus tag dari proyek (konsol)

Anda dapat menggunakan CodeBuild konsol untuk menghapus pengaitan antara tag dan CodeBuild proyek.

1. Buka CodeBuild konsol di <https://console.aws.amazon.com/codebuild/>.
2. Dalam proyek Build, pilih nama proyek tempat Anda ingin menghapus tag.
3. Di panel navigasi, pilih Pengaturan. Pilih Membangun tag proyek.
4. Pilih Edit.
5. Temukan tag yang ingin Anda hapus, lalu pilih Hapus tag.
6. Setelah Anda selesai menghapus tag, pilih Kirim.

## Hapus tag dari proyek (AWS CLI)

Untuk menghapus satu tanda atau lebih dari proyek build, lihat [Mengubah setelan proyek pembangunan \(AWS CLI\)](#). Perbaruitags bagian dalam data berformat JSON dengan daftar tag terbaru yang tidak berisi yang ingin Anda hapus. Jika Anda ingin menghapus semua tag, perbaruitags bagian ke:

```
"tags: []"
```

### Note

Jika menghapus proyekCodeBuild build, semua pengaitan tanda akan dihapus dari proyek build yang dihapus. Anda tidak perlu menghapus tanda sebelum menghapus proyek build.

## Batch dibangun di AWS CodeBuild

Anda dapat menggunakan AWS CodeBuild untuk menjalankan build proyek secara bersamaan dan terkoordinasi dengan build batch.

### Topik

- [Peran keamanan](#)
- [Jenis pembuatan Batch](#)
- [Mode laporan Batch](#)
- [Informasi lain](#)

## Peran keamanan

Batch build memperkenalkan peran keamanan baru dalam konfigurasi batch. Peran baru ini diperlukan karena CodeBuild harus dapat memanggilStartBuild,StopBuild, dan RetryBuild tindakan atas nama Anda untuk menjalankan build sebagai bagian dari batch. Pelanggan harus menggunakan peran baru, dan bukan peran yang sama yang mereka gunakan dalam build mereka, karena dua alasan:

- Memberikan peran buildStartBuild,StopBuild, dan RetryBuild izin akan memungkinkan satu build untuk memulai lebih banyak build melalui buildspec.

- CodeBuild build batch memberikan batasan yang membatasi jumlah build dan tipe komputasi yang dapat digunakan untuk build dalam batch. Jika peran build memiliki izin ini, ada kemungkinan build itu sendiri dapat melewati batasan ini.

## Jenis pembuatan Batch

CodeBuild mendukung jenis build batch berikut:

### Jenis pembuatan Batch

- [Membangun grafik](#)
- [Membangun daftar](#)
- [Membangun matriks](#)

### Membangun grafik

Grafik build mendefinisikan sekumpulan tugas yang memiliki dependensi pada tugas lain dalam batch.

Contoh berikut mendefinisikan grafik build yang membuat rantai dependensi.

```
batch:
 fast-fail: false
 build-graph:
 - identifier: build1
 env:
 variables:
 BUILD_ID: build1
 ignore-failure: false
 - identifier: build2
 buildspec: build2.yml
 env:
 variables:
 BUILD_ID: build2
 depend-on:
 - build1
 - identifier: build3
 env:
 variables:
 BUILD_ID: build3
 depend-on:
```

```
- build2
```

Dalam contoh ini:

- `build1` berjalan terlebih dahulu karena tidak memiliki dependensi.
- `build2` memiliki ketergantungan pada `build1`, jadi `build2` jalankan setelah `build1` selesai.
- `build3` memiliki ketergantungan pada `build2`, jadi `build3` jalankan setelah `build2` selesai.

Untuk informasi selengkapnya tentang sintaks `buildspec` grafik build, lihat. [batch/build-graph](#)

## Membangun daftar

Daftar build mendefinisikan sejumlah tugas yang berjalan secara paralel.

Contoh berikut mendefinisikan daftar build. The `build1` and `build2` build akan berjalan secara paralel.

```
batch:
 fast-fail: false
 build-list:
 - identifier: build1
 env:
 variables:
 BUILD_ID: build1
 ignore-failure: false
 - identifier: build2
 buildspec: build2.yml
 env:
 variables:
 BUILD_ID: build2
 ignore-failure: true
```

Untuk informasi selengkapnya tentang sintaks `buildspec` daftar build, lihat. [batch/build-list](#)

## Membangun matriks

Matriks build mendefinisikan tugas dengan konfigurasi berbeda yang berjalan secara paralel. CodeBuild membuat build terpisah untuk setiap kombinasi konfigurasi yang mungkin.

Contoh berikut menunjukkan matriks build dengan dua file `buildspec` dan tiga nilai untuk variabel lingkungan.

```
batch:
 build-matrix:
 static:
 ignore-failure: false
 dynamic:
 buildspec:
 - matrix1.yml
 - matrix2.yml
 env:
 variables:
 MY_VAR:
 - VALUE1
 - VALUE2
 - VALUE3
```

Dalam contoh ini, CodeBuild buat enam build:

- matrix1.yml dengan \$MY\_VAR=VALUE1
- matrix1.yml dengan \$MY\_VAR=VALUE2
- matrix1.yml dengan \$MY\_VAR=VALUE3
- matrix2.yml dengan \$MY\_VAR=VALUE1
- matrix2.yml dengan \$MY\_VAR=VALUE2
- matrix2.yml dengan \$MY\_VAR=VALUE3

Setiap build akan memiliki pengaturan berikut:

- ignore-failure disetel ke false
- env/type disetel ke LINUX\_CONTAINER
- env/image disetel ke aws/codebuild/amazonlinux2-x86\_64-standard:4.0
- env/privileged-mode disetel ke true

Build ini berjalan secara paralel.

Untuk informasi selengkapnya tentang sintaks buildspec matriks build, lihat. [batch/build-matrix](#)

## Mode laporan Batch

Jika penyedia sumber untuk project Anda adalah Bitbucket, GitHub, atau GitHub Enterprise, dan project Anda dikonfigurasi untuk melaporkan status build ke penyedia sumber, Anda dapat memilih bagaimana status build batch Anda dikirim ke penyedia sumber. Anda dapat memilih agar status dikirim sebagai laporan status agregat tunggal untuk batch, atau memiliki status setiap build dalam batch yang dilaporkan satu per satu.

Untuk informasi selengkapnya, lihat topik berikut:

- [Konfigurasi Batch \(buat\)](#)
- [Konfigurasi Batch \(pembaruan\)](#)

## Informasi lain

Untuk informasi selengkapnya, lihat topik berikut:

- [Batch membangun referensi buildspec](#)
- [Konfigurasi Batch](#)
- [Jalankan build batch \(AWS CLI\)](#)
- [Menghentikan pembuatan batchAWS CodeBuild](#)

## GitHubPelari aksi diAWS CodeBuild

Anda dapat menggunakan action runner untuk menjalankanGitHubTindakan dalamCodeBuild. Hal ini dapat dilakukan dengan menambahkanstepsuntuk setiapfase dalam file buildspec Anda.

CodeBuildbuildspecs mendukung daftar sekuensialGitHubLangkah tindakan yang berjalan dalam fase terpisah dariCodeBuildperintah. IniGitHubTindakan terintegrasi denganCodeBuildfitur yang ada, yang meliputi cache ketergantungan, pembuatan batch, akses keAWS Secrets Manager, dan banyak lagi.

### Topik

- [Bagaimana saya memulai denganGitHubpelari aksi?](#)
- [Yang manaGitHubTindakan yang bisa saya gunakan?](#)
- [Dapatkah saya menggunakan penyedia sumber selainGitHubsaat menggunakanGitHubTindakan di sayaCodeBuildproyek?](#)



- [Mengapa saya harus terhubung keGitHubsebagai penyedia sumber untuk menggunakanGitHubTindakan?](#)
- [Berapa biaya untuk menggunakanGitHubPelari aksi?](#)
- [Daerah mana yang mendukungGitHubPelari aksi?](#)
- [Praktik terbaik untukGitHubTindakan](#)
- [Keterbatasan dariGitHubpelari aksi diCodeBuild](#)
- [GitHubSampel tindakan denganAWS CodeBuild](#)

## Bagaimana saya memulai denganGitHubpelari aksi?

Langkah-langkah tingkat tinggi untuk menambahkanGitHubTindakan adalah sebagai berikut:

1. Jika Anda belum melakukannya, hubungkan proyek Anda keGitHub.

Untuk melakukannya, Anda dapat melakukan salah satu dari yang berikut:

- Anda dapat menambahkanGitHubsebagai penyedia sumber di konsol. Untuk informasi selengkapnya, lihat [Connect GitHub dengan token akses \(konsol\)](#).
- Anda dapat mengimporGitHubkredensial melalui[CodeBuildAPI](#). Untuk informasi selengkapnya, lihat [Connect GitHub dengan token akses \(CLI\)](#).

### Note

Ini hanya perlu dilakukan jika Anda belum terhubung keGitHubdalam proyek lain.

2. Di buildspec proyek Anda, Anda dapat menambahkansteps, yang masing-masing mereferensikan aGitHubTindakan. Ini dapat diedit diCodeBuildkonsol atau di repositori sumber Anda. Setiap fase build mendukung daftar perintah atau daftar langkah, tetapi keduanya tidak dapat digunakan dalam fase yang sama. Untuk informasi selengkapnya, lihat [GitHubReferensi buildspec pelari aksi](#).

## Yang manaGitHubTindakan yang bisa saya gunakan?

Anda dapat menggunakan tindakan apa pun yang tersedia di[GitHubPasaryang](#) tidak bertentangan dengan iniketerbatasan.

Dapatkah saya menggunakan penyedia sumber selain GitHub saat menggunakan GitHub Tindakan di saya CodeBuild proyek?

Ya, tetapi menghubungkan ke GitHub masih diperlukan untuk mengautentikasi dengan GitHub dan akses GitHub Tindakan. Untuk informasi selengkapnya, lihat [GitHub dan token akses Server GitHub Perusahaan](#).

Mengapa saya harus terhubung ke GitHub sebagai penyedia sumber untuk menggunakan GitHub Tindakan?

Untuk menggunakan GitHub Tindakan, sumber harus diunduh pada komputasi build. Unduhan anonim akan dibatasi tarif, jadi dengan menghubungkan ke GitHub, ini dapat membantu memastikan akses yang konsisten.

Berapa biaya untuk menggunakan GitHub Pelari aksi?

Berlari GitHub Tindakan didukung tanpa biaya tambahan.

Daerah mana yang mendukung GitHub Pelari aksi?

GitHub Tindakan didukung di semua CodeBuild daerah. Untuk informasi lebih lanjut tentang Wilayah AWS di mana CodeBuild tersedia, lihat [AWS Layanan menurut Wilayah](#).

Praktik terbaik untuk GitHub Tindakan

GitHub Tindakan bersifat open source, dibangun dan dipelihara oleh masyarakat. Kami mengikuti [model tanggung jawab bersama](#) dan pertimbangkan GitHub Kode sumber tindakan sebagai data pelanggan yang menjadi tanggung jawab Anda. GitHub Tindakan dapat diberikan akses ke rahasia, token repositori, kode sumber, dan tautan akun. Pastikan Anda yakin dengan kepercayaan dan keamanan GitHub Tindakan yang Anda rencanakan untuk dijalankan.

Panduan yang lebih spesifik dan praktik terbaik keamanan untuk GitHub Tindakan:

- [Pengerasan keamanan](#)
- [Mencegah permintaan pwn](#)
- [Masukan tidak tepercaya](#)
- [Bagaimana mempercayai blok bangunan Anda](#)

## Keterbatasan dari GitHub Pelari aksi di CodeBuild

- GitHub Tindakan yang secara internal bergantung pada [githubkonteks](#) atau referensi itu GitHub-sumber daya khusus, seperti permintaan tarik dan masalah, tidak didukung di CodeBuild. Misalnya, tindakan berikut tidak akan berfungsi di CodeBuild:
  - GitHub Tindakan yang mencoba menambah, mengubah, atau memperbarui GitHub sumber daya, seperti tindakan yang memperbarui permintaan tarik, atau membuat masalah di GitHub.

### Note

Paling resmi GitHub Tindakan yang tercantum dalam <https://github.com/actions> mengandalkan [githubkonteks](#). Sebagai gantinya, gunakan tindakan yang tersedia di [GitHub Pasar](#).

- GitHub Tindakan yang [Tindakan kontainer Docker](#) akan berhasil, tetapi proyek pembangunan Anda harus memiliki [mode istimewa](#) diaktifkan dan dijalankan oleh pengguna Docker default (root).
  - Tindakan harus dijalankan sebagai pengguna root. Untuk informasi lebih lanjut, lihat [PENGGUNA](#) topik di [Dukungan Dockerfile untuk GitHub Tindakan](#).
- GitHub Tindakan tidak didukung di CodeBuild proyek yang dikonfigurasi untuk berjalan di Windows.
- GitHub Pekerjaan aksi (kelompok langkah) dan GitHub Properti pekerjaan tindakan tidak didukung.
- GitHub Tindakan tidak didukung di CodeBuild proyek yang dikonfigurasi untuk dipicu oleh webhook untuk repositori Git publik. Untuk informasi lebih lanjut, lihat [git-credential-helper](#).
- VPC dibangun tanpa akses internet publik tidak dapat berjalan GitHub Tindakan.
- Setiap fase build mendukung daftar perintah atau daftar langkah, tetapi keduanya tidak dapat digunakan dalam fase yang sama. Misalnya, dalam contoh berikut, langkah-langkah digunakan dalam fase pra-pembuatan untuk dicantumkan GitHub Tindakan, sementara perintah digunakan dalam fase build ke daftar CodeBuild perintah.

```
version: 0.2
phases:
 pre-build:
 steps:
 - name: Lint Code Base
 uses: github/super-linter@v4
 env:
 VALIDATE_ALL_CODEBASE: 'true'
 DEFAULT_BRANCH: main
```

```
build:
 commands:
 - echo "Building..."
 - npm run build
```

## GitHubSampel tindakan denganAWS CodeBuild

Kelompok sampel ini dapat digunakan untuk bereksperimen denganGitHubTindakan diCodeBuild.

### Topik

- [Super-LinterGitHubSampel tindakan](#)
- [Contoh grafik pembuatan batch](#)
- [AmazonCodeGuruSampel pengulas](#)
- [Sampel AWS Secrets Manager](#)
- [Sampel variabel lingkungan](#)
- [Sampel variabel lingkungan yang diekspor](#)

### Super-LinterGitHubSampel tindakan

Contoh ini menunjukkan cara menambahkan[Super-Linter](#) GitHubTindakan ke aCodeBuildproyek. Tindakan Super-Linter memeriksa kode, menemukan area di mana kode memiliki kesalahan, masalah pemformatan, dan konstruksi yang mencurigakan, dan kemudian menampilkan hasilnya keCodeBuildkonsol.

Anda dapat menambahkan Super-LinterGitHubTindakan untuk AndaCodeBuildproyek dengan memperbarui bagian fase file buildspec Anda.

```
version: 0.2
phases:
 build:
 steps:
 - name: Lint Code Base
 uses: github/super-linter@v5
 env:
 VALIDATE_ALL_CODEBASE: true
```

Log Super-Linter akan terlihat mirip dengan yang berikut ini:

```
/github/workspace/hello-world/app.js:3:13: Extra semicolon.
/github/workspace/hello-world/app.js:9:92: Trailing spaces not allowed.
/github/workspace/hello-world/app.js:21:7: Unnecessarily quoted property 'body' found.
/github/workspace/hello-world/app.js:31:1: Expected indentation of 2 spaces but found
4.
/github/workspace/hello-world/app.js:32:2: Newline required at end of file but not
found.
```

## Contoh grafik pembuatan batch

Contoh berikut mendefinisikan grafik build yang membuat rantai dependensi dan menjalankan perintah menggunakan `steps`. Dalam contoh ini, `build1` berjalan terlebih dahulu karena tidak memiliki dependensi. Sejak `build2` memiliki ketergantungan pada `build1`, jadi `build2` berjalan setelah `build1` selesai. Untuk informasi, lihat [Membangun grafik](#).

```
version: 0.2
batch:
 fast-fail: false
 build-graph:
 - identifier: build1
 env:
 variables:
 BUILD_ID: build1
 ignore-failure: false
 - identifier: build2
 env:
 variables:
 BUILD_ID: build2
 depend-on:
 - build1

phases:
 build:
 steps:
 - run: echo $BUILD_ID
```

## AmazonCodeGuruSampel pengulas

AmazonCodeGuruPeninjau menemukan masalah dalam kode Java dan Python Anda dan merekomendasikan cara memperbaikinya. Contoh berikut menggunakan `CodeGuruReviewer` untuk memberikan ulasan kode analisis repositori lengkap. Ulasan kode ini

memindai semua kode di cabang tertentu. Untuk informasi, lihat [Buat ulasan kode dengan GitHub Tindakan](#) di Amazon CodeGuru Panduan Pengguna Reviewer.

```
version: 0.2
phases:
 build:
 steps:
 - name: Amazon CodeGuru Reviewer Scanner
 if: ${{ always() }}
 uses: aws-actions/codeguru-reviewer@v1.1
 with:
 s3_bucket: codeguru-reviewer-user

artifacts:
 files:
 - codeguru-results.sarif.json
```

#### Note

Bucket Amazon S3 Anda harus dimulai dengan `codeguru-reviewer-awalan`.

Log akan terlihat mirip dengan yang berikut:

```
INFO CodeReview created with arn=arn:aws:codeguru-reviewer:region:account-id:association:id:code-review:RepositoryAnalysis-job for job=job
INFO SARIF persisted to /github/workspace/codeguru-results.sarif.json
INFO Amazon CodeGuru Reviewer job execution completed
```

Setelah Amazon CodeGuru Pekerjaan reviewer selesai, laporan sarif dihasilkan sebagai CodeBuild artefak. Untuk informasi, lihat [Analisis repositori lengkap](#) di Amazon CodeGuru Panduan Pengguna Reviewer.

## Sampel AWS Secrets Manager

AWS Secrets Manager membantu Anda mengelola, mengambil, dan memutar kredensial basis data, kredensial aplikasi, token OAuth, kunci API, dan rahasia lainnya sepanjang siklus hidupnya. Contoh berikut mendefinisikan rahasia menggunakan Secrets Manager dan menjalankan perintah menggunakan `steps`. Untuk informasi, lihat [Apa itu AWS Secrets Manager?](#) di AWS Secrets Manager Panduan Pengguna.

```

version: 0.2
env:
 secrets-manager:
 SECRET_VALUE: "arn:aws:secretsmanager:us-east-1:xxxx:secret:/secret-
13IJg9:my_super_secret_key"
phases:
 build:
 steps:
 - run: echo $SECRET_VALUE

```

Log akan terlihat mirip dengan yang berikut:

```

echo $SECRET_VALUE
env:
 SECRET_VALUE: ***

```

## Sampel variabel lingkungan

Contoh berikut mendefinisikan variabel lingkungan di bawahenvvurutan. SEBUAH*S3\_EMBER*variabel didefinisikan dalam buildspec dan ditugaskan<*bucket-name*>sebagai nilainya. Variabel ini direferensikan dalam kondisional if seperti variabel lingkungan biasa dengan menggunakan tanda dolar (\$) untuk mengaksesGitHubKonteks env tindakan. Untuk informasi lebih lanjut, lihatenvvurutan.

```

version: 0.2
env:
 variables:
 S3_BUCKET: "<bucket-name>"
phases:
 build:
 steps:
 - if: ${{ env.S3_BUCKET == '<bucket-name>' }}
 run: echo "S3 bucket is $S3_BUCKET"

```

Log akan terlihat mirip dengan yang berikut:

```

echo "S3 bucket is $S3_BUCKET"
env:
 S3_BUCKET: my-s3-bucket
S3 bucket is my-s3-bucket

```

## Sampel variabel lingkungan yang diekspor

Variabel lingkungan yang diekspor digunakan bersama dengan CodePipeline untuk mengeksport variabel lingkungan dari tahap build saat ini ke tahap selanjutnya dalam pipeline. Contoh berikut mendefinisikan variabel lingkungan yang diekspor di bawah `env` urutan bernama `MY_VARIABEL` dan menulis ke `GITHUB_ENV` file lingkungan.

```
version: 0.2
env:
 exported-variables:
 - MY_VARIABEL
phases:
 build:
 steps:
 - run: echo "MY_VARIABEL=my-value" >> $GITHUB_ENV
```

Untuk informasi lebih lanjut, lihat [Exported Environment Variable](#) di AWS CodeBuild Referensi API.

## Proyek pembangunan publik di AWS CodeBuild

AWS CodeBuild memungkinkan Anda untuk membuat hasil pembangunan, log, dan artefak untuk proyek membangun Anda tersedia untuk masyarakat umum. Hal ini memungkinkan kontributor ke repositori sumber Anda untuk melihat hasil dan men-download artefak membangun, tanpa mengharuskan mereka untuk memiliki akses ke AWS Akun.

Ketika Anda membuat proyek Anda membangun tersedia untuk umum, semua proyek membangun hasil, log, dan artefak, termasuk membangun yang dijalankan ketika proyek bersifat pribadi, tersedia untuk umum. Demikian juga, ketika Anda membuat proyek pembangunan publik pribadi, hasil pembangunan untuk proyek itu tidak lagi tersedia untuk umum.

Untuk informasi tentang cara mengubah visibilitas publik hasil pembangunan proyek Anda, lihat [Aktifkan akses build publik](#).

CodeBuild menyediakan URL untuk membangun publik untuk proyek Anda yang unik untuk proyek Anda. Untuk mendapatkan URL publik untuk proyek build Anda, lakukan prosedur berikut:

1. Buka AWS CodeBuild konsol di <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Di panel navigasi, pilih Proyek membangun.
3. Pilih tautan untuk proyek build yang Anda ingin dapatkan URL publik.



4. URL publik ditampilkan dalam URL proyek publik bidang di Konfigurasi bagian. Anda dapat memilih tautan untuk membuka URL, atau menyalin URL dengan tombol salin.

#### Warning

Berikut ini harus diingat ketika membuat proyek Anda membangun hasil publik:

- Semua proyek membangun hasil, log, dan artefak, termasuk membangun yang dijalankan ketika proyek bersifat pribadi, tersedia untuk umum.
- Semua log bangunan dan artefak tersedia untuk umum. variabel lingkungan, kode sumber, dan informasi sensitif lainnya mungkin telah output untuk membangun log dan artefak. Anda harus berhati-hati tentang informasi apa yang output ke log membangun. Beberapa praktik terbaik adalah:
  - Jangan menyimpan nilai sensitif, terutama AWS akses ID kunci dan kunci akses rahasia, dalam variabel lingkungan. Kami menyarankan Anda menggunakan Parameter Store di Amazon EC2 Systems Manager AWS Secrets Manager untuk menyimpan nilai-nilai sensitif.
  - Ikuti [Praktik terbaik untuk menggunakan webhooks](#) untuk membatasi entitas mana yang dapat memicu membangun, dan tidak menyimpan buildspec dalam proyek itu sendiri, untuk memastikan bahwa webhooks Anda seaman mungkin.
- Pengguna berbahaya dapat menggunakan build publik untuk mendistribusikan artefak berbahaya. Kami menyarankan administrator proyek meninjau semua permintaan tarik untuk memverifikasi bahwa pull request adalah perubahan yang sah. Kami juga menyarankan Anda memvalidasi artefak dengan checksum mereka untuk memastikan bahwa artefak yang benar sedang diunduh.

## Bekerja dengan membangun di AWS CodeBuild

SEBUAH membangun merupakan satu set tindakan yang dilakukan oleh AWS CodeBuild untuk membuat artefak keluaran (misalnya, file JAR) berdasarkan satu set artefak masukan (misalnya, koleksi file kelas Java).

Aturan berikut berlaku saat Anda menjalankan beberapa build:

- Bila memungkinkan, membangun berjalan secara bersamaan. Jumlah maksimum build yang berjalan secara bersamaan dapat bervariasi. Untuk informasi selengkapnya, lihat [Kuota untuk AWS CodeBuild](#).
- Jika proyek build memiliki set limit build bersamaan, build mengembalikan kesalahan jika jumlah build yang berjalan mencapai batas build bersamaan untuk proyek. Untuk informasi selengkapnya, lihat [Aktifkan batas build bersamaan](#).
- Jika proyek build tidak memiliki set limit build bersamaan, build akan antri jika jumlah build yang berjalan mencapai batas build bersamaan untuk platform dan tipe komputasi. Jumlah maksimum build dalam antrian adalah lima kali batas pembangunan bersamaan. Untuk informasi selengkapnya, lihat [Kuota untuk AWS CodeBuild](#).

Sebuah membangun dalam antrian yang tidak dimulai setelah jumlah menit yang ditentukan dalam nilai waktu keluar dihapus dari antrian. Nilai batas waktu default adalah delapan jam. Anda dapat mengganti batas waktu antrian build dengan nilai antara lima menit dan delapan jam ketika Anda menjalankan build. Untuk informasi selengkapnya, lihat [Jalankan build diAWS CodeBuild](#).

Hal ini tidak mungkin untuk memprediksi urutan di mana antrian membangun mulai.

#### Note

Anda dapat mengakses sejarah membangun selama satu tahun.

Anda dapat melakukan tugas-tugas ini saat bekerja dengan build:

#### Topik

- [Jalankan build diAWS CodeBuild](#)
- [Lihat detail build diAWS CodeBuild](#)
- [Melihat daftar ID build diAWS CodeBuild](#)
- [Melihat daftar ID build untuk proyek build diAWS CodeBuild](#)
- [Hentikan membangun diAWS CodeBuild](#)
- [Menghentikan pembuatan batchAWS CodeBuild](#)
- [Coba lagi membangun diAWS CodeBuild](#)
- [Melihat build yang sedang berjalan di Session Manager](#)
- [Hapus build diAWS CodeBuild](#)

## Jalankan build diAWS CodeBuild

Anda dapat menggunakanAWS CodeBuildkonsol,AWS CLI, atauAWSSDK untuk menjalankan membangun di CodeBuild.

### Topik

- [Menjalankan build \(konsol\)](#)
- [Jalankan build \(AWS CLI\)](#)
- [Jalankan build batch \(AWS CLI\)](#)
- [Mulai menjalankan build secara otomatis \(AWS CLI\)](#)
- [Berhenti menjalankan build secara otomatis \(AWS CLI\)](#)
- [Jalankan membangun \(AWSSDK\)](#)

### Menjalankan build (konsol)

Untuk menggunakanAWS CodePipelineuntuk menjalankan build dengan CodeBuild, lewati langkah-langkah ini dan ikuti petunjuk di[Gunakan CodePipeline dengan CodeBuild](#).

1. BukaAWS CodeBuildkonsol di<https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Di panel navigasi, pilihProyek pembangunan.
3. Dalam daftar proyek build, pilih proyek build.
4. Anda dapat menjalankan build dengan pengaturan project build default, atau mengganti pengaturan build untuk build ini saja.
  - a. Jika Anda ingin menjalankan build dengan pengaturan proyek build default, pilihMulai membangun. Membangun dimulai segera.
  - b. Jika Anda ingin mengganti pengaturan proyek build default, pilihMulai membangun dengan menimpa. DiMulai membangunHalaman, Anda dapat menimpa berikut:
    - Konfigurasi build
    - Sumber
    - Penggantian variabel lingkungan

Jika Anda perlu memilih penggantian yang lebih maju, pilihBuild lanjutan. Di halaman ini, Anda dapat menimpa berikut:

- Konfigurasi build
- Sumber
- Lingkungan
- spesifikasi buildspec
- Artefak
- Beberapa catatan

Bila Anda telah membuat pilihan menimpa Anda, pilih **Mulai** membangun.

Untuk informasi mendetail tentang build ini, lihat [Lihat detail build \(konsol\)](#).

## Jalankan build (AWS CLI)

### Note

Untuk menggunakan CodePipeline untuk menjalankan build dengan AWS CodeBuild, lewati langkah-langkah ini dan ikuti instruksi di [Buat pipeline yang menggunakan CodeBuild \(AWS CLI\)](#).

Untuk informasi lebih lanjut tentang menggunakan AWS CLI bersama CodeBuild, lihat [Referensi baris perintah](#).

1. Jalankan `start-build` perintah dalam salah satu cara berikut:

```
aws codebuild start-build --project-name <project-name>
```

Gunakan ini jika Anda ingin menjalankan build yang menggunakan artefak input build versi terbaru dan setelan proyek build yang ada.

```
aws codebuild start-build --generate-cli-skeleton
```

Gunakan ini jika Anda ingin menjalankan build dengan artefak input build versi sebelumnya atau jika Anda ingin mengganti pengaturan untuk artefak keluaran build, variabel lingkungan, buildspec, atau periode batas waktu build default.

2. Jika Anda menjalankan `start-build` perintah dengan `--project-name` opsi, ganti `<project-name>` dengan nama proyek build, lalu lewati ke langkah 6 dari prosedur ini. Untuk mendapatkan daftar proyek pembangunan, lihat [Melihat daftar nama proyek build](#).
3. Jika Anda menjalankan `start-build` perintah dengan `--idempotency-token` opsi, pengidentifikasi atau token peka huruf besar/kecil yang unik, disertakan dengan `start-build` permintaan. Token berlaku selama 5 menit setelah permintaan. Jika Anda mengulang `start-build` permintaan dengan token yang sama, tetapi ubah parameter, CodeBuild mengembalikan kesalahan ketidakcocokan parameter.
4. Jika Anda menjalankan `start-build` perintah dengan `--generate-cli-skeleton` opsi, data berformat JSON muncul di output. Salin data ke file (misalnya, `start-build.json`) di lokasi di komputer lokal atau contoh di mana AWS CLI diinstal. Ubah data yang disalin agar sesuai dengan format berikut, dan simpan hasil Anda:

```
{
 "projectName": "projectName",
 "sourceVersion": "sourceVersion",
 "artifactsOverride": {
 "type": "type",
 "location": "location",
 "path": "path",
 "namespaceType": "namespaceType",
 "name": "artifactsOverride-name",
 "packaging": "packaging"
 },
 "buildspecOverride": "buildspecOverride",
 "cacheOverride": {
 "location": "cacheOverride-location",
 "type": "cacheOverride-type"
 },
 "certificateOverride": "certificateOverride",
 "computeTypeOverride": "computeTypeOverride",
 "environmentTypeOverride": "environmentTypeOverride",
 "environmentVariablesOverride": {
 "name": "environmentVariablesOverride-name",
 "value": "environmentVariablesValue",
 "type": "environmentVariablesOverride-type"
 },
 "gitCloneDepthOverride": "gitCloneDepthOverride",
 "imageOverride": "imageOverride",
 "idempotencyToken": "idempotencyToken",
 "insecureSslOverride": "insecureSslOverride",
```

```
"privilegedModeOverride": "privilegedModeOverride",
"queuedTimeoutInMinutesOverride": "queuedTimeoutInMinutesOverride",
"reportBuildStatusOverride": "reportBuildStatusOverride",
"timeoutInMinutesOverride": "timeoutInMinutesOverride",
"sourceAuthOverride": "sourceAuthOverride",
"sourceLocationOverride": "sourceLocationOverride",
"serviceRoleOverride": "serviceRoleOverride",
"sourceTypeOverride": "sourceTypeOverride"
}
```

Ganti placeholder berikut:

- ***ProjectName***: String yang dibutuhkan. Nama proyek build yang akan digunakan untuk build ini.
- ***SourceVersion***: String opsional. Versi kode sumber yang akan dibangun, sebagai berikut:
  - Untuk Amazon S3, ID versi yang sesuai dengan versi file ZIP input yang ingin Anda buat. Jika ***SourceVersion*** tidak ditentukan, maka versi terbaru digunakan.
  - Untuk CodeCommit, ID komit yang sesuai dengan versi kode sumber yang ingin Anda buat. Jika ***SourceVersion*** tidak ditentukan, ID komit HEAD cabang default digunakan. (Anda tidak dapat menentukan nama tag untuk ***SourceVersion***, tetapi Anda dapat menentukan ID komit tag.)
  - Untuk GitHub, ID komit, ID permintaan tarik, nama cabang, atau nama tag yang sesuai dengan versi kode sumber yang ingin Anda buat. Jika ID permintaan tarik ditentukan, ia harus menggunakan format `pr/pull-request-ID` (misalnya `pr/25`). Jika nama cabang ditentukan, ID melakukan HEAD cabang digunakan. Jika ***SourceVersion*** tidak ditentukan, ID komit HEAD cabang default digunakan.
  - Untuk Bitbucket, ID komit, nama cabang, atau nama tag yang sesuai dengan versi kode sumber yang ingin Anda buat. Jika nama cabang ditentukan, ID melakukan HEAD cabang digunakan. Jika ***SourceVersion*** tidak ditentukan, ID komit HEAD cabang default digunakan.
- Placeholder berikut adalah untuk ***artifactsOverride***.
  - ***type***: Opsional. Jenis artefak keluaran build yang menggantikan build ini adalah yang ditentukan dalam proyek build.
  - ***lokasi***: Opsional. Lokasi artefak keluaran build yang menggantikan build ini adalah lokasi yang ditentukan dalam proyek build.

- ***jalan***: Opsional. Jalur artefak keluaran build yang menggantikan build ini adalah yang ditentukan dalam proyek build.
- ***Namespacetype***: Opsional. Jenis jalur artefak keluaran build yang menggantikan build ini adalah yang ditentukan dalam proyek build.
- ***nama***: Opsional. Nama artefak keluaran build yang menggantikan build ini adalah nama yang ditentukan dalam proyek build.
- ***kemasan***: Opsional. Jenis kemasan artefak keluaran build yang menggantikan build ini adalah yang ditentukan dalam proyek build.
- ***BuildSpecoverride***: Opsional. Deklarasi buildspec yang mengganti build ini yang ditentukan dalam proyek build. Jika nilai ini disetel, itu bisa berupa definisi buildspec sebaris, jalur ke file buildspec alternatif relatif terhadap nilai built-inCODEBUILD\_SRC\_DIRvariabel lingkungan, atau jalur ke bucket S3. Bucket S3 harus samaAWSWilayah sebagai proyek pembangunan. Tentukan file buildspec menggunakan ARN (misalnya,`arn:aws:s3:::<my-codebuild-sample2>/buildspec.yml`). Jika nilai ini tidak disediakan atau diatur ke string kosong, kode sumber harus berisibuildspec.ymlberkas di direktori root-nya. Untuk informasi selengkapnya, lihat [Nama file Buildspec dan lokasi penyimpanan](#).
- Placeholder berikut adalah untukcacheOverride.
  - ***Lokasi Cacheoverride***: Opsional. Lokasi aProjectCacheobjek untuk build ini yang mengesampingkanProjectCacheobjek yang ditentukan dalam proyek build.cacheOverridebersifat opsional dan membutuhkanProjectCacheobjek.locationdiperlukan dalamProjectCacheobjek.
  - ***Jenis Cacheoverride***: Opsional. Jenis aProjectCacheobjek untuk build ini yang mengesampingkanProjectCacheobjek yang ditentukan dalam proyek build.cacheOverridebersifat opsional dan membutuhkanProjectCacheobjek.typediperlukan dalamProjectCacheobjek.
- ***CertificateOverride***: Opsional. Nama sertifikat untuk build ini yang menggantikan sertifikat yang ditentukan dalam proyek build.
- ***environmentTypeOverride***: Opsional. Jenis kontainer untuk build ini yang mengganti yang ditentukan dalam proyek build. String yang valid saat ini adalahLINUX\_CONTAINER.
- Placeholder berikut adalah untukenvironmentVariablesOverride.
  - ***environmentVariablesOverride-nama***: Opsional. Nama variabel lingkungan dalam proyek build yang nilainya ingin Anda timpa untuk build ini.
  - ***environmentVariablesOverride-jenis***: Opsional. Jenis variabel lingkungan dalam proyek build yang nilainya ingin Anda timpa untuk build ini.

- *environmentVariablesValue*: Opsional. Nilai variabel lingkungan yang ditentukan dalam proyek build yang ingin Anda timpa untuk build ini.
- *gitCloneDepthGanti*: Opsional. Nilai dari kedalaman klon git dalam proyek build yang nilainya ingin Anda timpa untuk build ini. Jika jenis sumber Anda adalah Amazon S3, nilai ini tidak didukung.
- *ImageOverride*: Opsional. Nama gambar untuk build ini yang menggantikan yang ditentukan dalam proyek build.
- *IdempotencyToken*: Opsional. String yang berfungsi sebagai token untuk menentukan bahwa permintaan build adalah idempoten. Anda dapat memilih string yang 64 karakter atau kurang. Token berlaku selama 5 menit setelah permintaan start-build. Jika Anda mengulangi permintaan start-build dengan token yang sama, tetapi mengubah parameter, CodeBuild mengembalikan kesalahan ketidakcocokan parameter.
- *insecureSslOverride*: Boolean opsional yang menentukan apakah akan mengganti setelah TLS tidak aman yang ditentukan dalam proyek build. Pengaturan TLS yang tidak aman menentukan apakah akan mengabaikan peringatan TLS saat menghubungkan ke kode sumber proyek. Penggantian ini hanya berlaku jika sumber build GitHub Server Perusahaan.
- *privilegedModeOverride*: Boolean opsional. Jika disetel ke true, build akan mengganti mode istimewa dalam proyek build.
- *queuedTimeoutInMinutesOverride*: Integer opsional yang menentukan jumlah menit build diizinkan untuk antri sebelum waktu habis. Nilai minimumnya adalah lima menit dan nilai maksimumnya adalah 480 menit (delapan jam).
- *reportBuildStatusGanti*: Boolean opsional yang menentukan apakah akan mengirim status awal dan penyelesaian build kepada penyedia sumber Anda. Jika Anda menyetel ini dengan penyedia sumber selain GitHub, GitHub Server Perusahaan, atau Bitbucket, `invalidInputException` dilemparkan.
- *sourceAuthOverride*: String opsional. Jenis otorisasi untuk build ini yang menggantikan yang ditentukan dalam proyek build. Penggantian ini hanya berlaku jika sumber proyek build adalah Bitbucket atau GitHub.
- *sourceLocationOverride*: String opsional. Lokasi yang menggantikan lokasi sumber ini akan membangun lokasi sumber untuk lokasi yang ditentukan dalam proyek build.
- *serviceRoleOverride*: String opsional. Nama peran layanan untuk build ini yang menggantikan peran yang ditentukan dalam proyek build.
- *sourceTypeOverride*: String opsional. Tipe input sumber untuk build ini yang menggantikan input sumber yang ditentukan dalam proyek build. String yang valid



adalah `NO_SOURCE`, `CODECOMMIT`, `CODEPIPELINE`, `GITHUB`, `S3`, `BITBUCKET`, dan `GITHUB_ENTERPRISE`.

- ***timeoutInMinutesGanti***: Nomor opsional. Jumlah menit batas waktu build yang menggantikan build ini adalah yang ditentukan dalam proyek build.

Kami menyarankan Anda menyimpan variabel lingkungan dengan nilai sensitif, seperti `AWSID` kunci akses, sebuah `AWS` kunci akses rahasia, atau kata sandi sebagai parameter di Amazon EC2 Systems Manager Parameter Store. CodeBuild dapat menggunakan parameter yang disimpan di Amazon EC2 Systems Manager Parameter Store hanya jika nama parameter itu dimulai dengan `/CodeBuild/` (Sebagai contoh, `/CodeBuild/dockerLoginPassword`). Anda dapat menggunakan CodeBuild konsol untuk membuat parameter di Amazon EC2 Systems Manager. Pilih **Buat parameter**, dan kemudian ikuti instruksinya. (Dalam kotak dialog itu, untuk **Kunci KMS**, Anda dapat secara opsional menentukan ARN dari sebuah `AWS` **KMS** kunci di akun Anda. Amazon EC2 Systems Manager menggunakan kunci ini untuk mengenkripsi nilai parameter selama penyimpanan dan mendekripsi selama pengambilan.) Jika Anda menggunakan CodeBuild konsol untuk membuat parameter, konsol memulai parameter dengan `/CodeBuild/` seperti yang sedang disimpan. Namun, jika Anda menggunakan konsol Amazon EC2 Systems Manager Parameter Store untuk membuat parameter, Anda harus memulai nama parameter dengan `/CodeBuild/`, dan Anda harus mengatur **Jenis** kepada `String Aman`. Untuk informasi lebih lanjut, lihat [AWS Systems Manager penyimpanan parameter](#) dan [Walkthrough: Buat dan uji parameter String \(konsol\)](#) di Panduan Pengguna Manajer Sistem Amazon EC2.

Jika project build Anda mengacu pada parameter yang disimpan di Amazon EC2 Systems Manager Parameter Store, peran layanan project build harus mengizinkan `ssm:GetParameter` tindakan. Jika Anda memilih **Buat** peran layanan baru di akun Anda sebelumnya, maka CodeBuild menyertakan tindakan ini dalam peran layanan default untuk proyek build Anda secara otomatis. Namun, jika Anda memilih **Pilih** peran layanan yang ada dari akun Anda, maka Anda harus menyertakan tindakan ini dalam peran layanan Anda secara terpisah.

Variabel lingkungan yang Anda tetapkan menggantikan variabel lingkungan yang ada. Misalnya, jika image Docker sudah berisi variabel lingkungan bernama `MY_VAR` dengan nilai `my_value`, dan Anda menetapkan variabel lingkungan bernama `MY_VAR` dengan nilai `other_value`, maka `my_value` digantikan oleh `other_value`. Demikian pula, jika image Docker sudah berisi variabel lingkungan bernama `PATH` dengan nilai `/usr/local/sbin:/usr/local/bin`, dan Anda menetapkan variabel lingkungan bernama `PATH` dengan nilai `$PATH:/usr/share/ant/`

bin, maka `/usr/local/sbin:/usr/local/bin` digantikan oleh nilai literal `$PATH:/usr/share/ant/bin`.

Jangan mengatur variabel lingkungan apa pun dengan nama yang dimulai dengan `CODEBUILD_`. Awalan ini dicadangkan untuk penggunaan internal.

Jika variabel lingkungan dengan nama yang sama didefinisikan di beberapa tempat, nilai variabel lingkungan ditentukan sebagai berikut:

- Nilai dalam panggilan operasi `start build` lebih diutamakan.
- Nilai dalam definisi proyek build akan diutamakan berikutnya.
- Nilai dalam deklarasi file `buildspec` diutamakan paling rendah.

Untuk informasi tentang nilai valid untuk placeholder ini, lihat [Buat proyek build \(AWS CLI\)](#). Untuk daftar setelan terbaru untuk proyek build, lihat [Lihat detail proyek pembangunan](#).

5. Beralih ke direktori yang berisi file yang baru saja Anda simpan, dan jalankan `start-build` perintah lagi.

```
aws codebuild start-build --cli-input-json file://start-build.json
```

6. Jika berhasil, data serupa dengan yang dijelaskan dalam [Untuk menjalankan build](#) prosedur muncul di output.

Untuk bekerja dengan informasi terperinci tentang build ini, catatidnilai dalam output, dan kemudian lihat [Lihat detail build \(AWS CLI\)](#).

## Jalankan build batch (AWS CLI)

1. Jalankan `start-build-batch` perintah dalam salah satu cara berikut:

```
aws codebuild start-build-batch --project-name <project-name>
```

Gunakan ini jika Anda ingin menjalankan build yang menggunakan artefak input build versi terbaru dan pengaturan proyek build yang ada.

```
aws codebuild start-build-batch --generate-cli-skeleton > <json-file>
```

Gunakan ini jika Anda ingin menjalankan build dengan versi artefak input build sebelumnya atau jika Anda ingin mengganti pengaturan untuk artefak keluaran build, variabel lingkungan, buildspec, atau periode batas waktu build default.

2. Jika Anda menjalankan `start-build-batch` perintah dengan `--project-name` pilihan, ganti `<project-name>` dengan nama proyek membangun, dan kemudian melompat ke langkah 6 dari prosedur ini. Untuk mendapatkan daftar proyek pembangunan, lihat [Melihat daftar nama proyek build](#).
3. Jika Anda menjalankan `start-build-batch` perintah dengan `--idempotency-token` pilihan, pengenal case-sensitive unik, atau token, disertakan dengan `start-build-batch` permintaan. Token berlaku selama 5 menit setelah permintaan. Jika Anda mengulang `start-build-batch` permintaan dengan token yang sama, tetapi mengubah parameter, CodeBuild mengembalikan kesalahan parameter ketidakcocokan.
4. Jika Anda menjalankan `start-build-batch` perintah dengan `--generate-cli-skeleton` pilihan, data JSON-diformat adalah output ke `<json-file>` berkas. File ini mirip dengan skelton yang dihasilkan oleh `start-build` perintah, dengan penambahan objek berikut. Untuk informasi selengkapnya tentang objek umum, lihat [Jalankan build \(AWS CLI\)](#).

Ubah file ini untuk menambahkan penggantian build, dan simpan hasil Anda.

```
"buildBatchConfigOverride": {
 "combineArtifacts": combineArtifacts,
 "restrictions": {
 "computeTypesAllowed": [
 allowedComputeTypes
],
 "maximumBuildsAllowed": maximumBuildsAllowed
 },
 "serviceRole": "batchServiceRole",
 "timeoutInMins": batchTimeout
}
```

Parameter `buildBatchConfigOverride` objek adalah [ProjectBuildBatchConfig](#) struktur yang berisi overrides konfigurasi build batch untuk build ini.

### *Menggabungkan Tifacts*

Sebuah boolean yang menentukan apakah membangun artefak untuk membangun batch harus digabungkan menjadi lokasi artefak tunggal.

### *DibolehComputeTypes*

Array string yang menentukan jenis komputasi yang diizinkan untuk pembangunan batch. Lihat [Membangun jenis komputasi lingkungan](#) untuk nilai-nilai ini.

### *MaximumBuildsAllowed*

Menentukan jumlah maksimum pembangunan yang diizinkan.

### *BatchServiceRole*

Menentukan ARN peran layanan untuk proyek build batch.

### *BatchTimeout*

Menentukan jumlah waktu maksimum, dalam menit, bahwa build batch harus diselesaikan.

5. Beralih ke direktori yang berisi file yang baru saja Anda simpan, dan jalankan `start-build-batch` perintah lagi.

```
aws codebuild start-build-batch --cli-input-json file://start-build.json
```

6. Jika berhasil, representasi JSON [BuildBatch](#) objek muncul dalam output konsol. Lihat [Sintaks Respons StartBuildBatch](#) untuk contoh data ini.

## Mulai menjalankan build secara otomatis (AWS CLI)

Jika kode sumber Anda disimpan dalam repositori GitHub atau GitHub Enterprise Server, Anda dapat menggunakan webhooks GitHub untuk memilikinya AWS CodeBuild membangun kembali kode sumber Anda setiap kali perubahan kode didorong ke repositori.

Jalankan `create-webhook` perintah sebagai berikut:

```
aws codebuild create-webhook --project-name <project-name>
```

*<project-name>* adalah nama proyek build yang berisi kode sumber yang akan dibangun kembali.

Untuk GitHub, informasi yang serupa dengan berikut akan muncul dalam output:

```
{
 "webhook": {
 "url": "<url>"
 }
}
```

}

`<url>` adalah URL ke webhook GitHub.

Untuk GitHub Enterprise Server, informasi yang serupa dengan berikut muncul dalam output:

```
{
 "webhook": {
 "secret": "YRV4JYAGfsekJiirp5ytx86oZpyhUdySNSDTLNUxOXX1c7aZ6XYDF37-ZFyY02rs4JSE70mLW3w-gh-ryoVB80SSSC1aAtBtuPkHwYuncCCmdogCVCfniQ7ukYX2_xM--n1Dma5EngIg_Bi_N465yi33zyTUNPoQ1xCpL0-BwghcVa91AurwR77-uY7i-_XCJFahwMx1f4ub0gBBsMT2A16apqjqQJoKSb61XVKyZy1Giuy4nliAXFv9WmN76CaCsndb3fVIE78fpygfo41xYxSQ6vpo6LRTKtPzbyeTHbVXGda1PJvnkBlnKmJDo0RTgI1m2oYr17dwziQ1rrvoCoNgy1S00_7LkFA-nNXFc_f1SiFy0AqeMB43-d00cdkzybHncE81QTRwEUCFfmX-AJCwmlXV0kg0G67T925jbpz0fRlkh5pwIF193_bB_jOHdInK6i0iPpf2dIDAIZgGMagqZewb-axDeTAbopoU8J6gFI1yKo5aq9q151zC1PERUsMgJFtJr_a-Z-L_ky1r-4hSSxasSJNuJ43_XOBRWqT51xqvH-A69bv07KbVT_Kc6wxkSHyYCEMoa_Pfa7ZQgyfY6B00gMNj31yFbjth0RNL1cDo6-3J-McDLoyrRtSEOV9QnxvsG5zu1N5-z20rkJtg_M0fNwocfUutFXb7vrGTduH1R1dzXLrusHuxOVVuDUWm9vhwMr-hUkeGo_1kDKyk4E2QFvZxpjYw0vFfV-dwxFRR_mifzxW1wyfmt2iFtlkp_YZj_4WeFackGefr-ilNaYvsZpzXj78Ae1adVoLf48AmDdn2pwsWJjatU9zt942gLisFFmKakcvJuy5yxXHaxxbhUyC8NHYiESUWPfcfnqrMsr8op3P4AUCH1piZCYUuiwI_cac-pIUB00Xaur_lu_fyFghg0Jc7cfTnA36rv5X5DnFDM8P3HNBeLjaF9QZ6AijegPEwTHIKJON3AUDwpkz_hwTXyUoAU8MdZfPTXbBoT6N5Z5THBHsYxR",
 "payloadUrl": "https://codebuild.us-east-2.amazonaws.com/webhooks?t=eyJlbmNyeXB0ZWREYXRhIjoieUwFqMmJERGRQbGhwLzNTN1d3R0VGRjZzOTNwLz1ZVG1NZ1pIR1E0RUSxdzhGeWhnVFFqWTR0WEFwT2dJRnNmRHc3S3Rnc0xYMEncXFTakg1cE1nSy9zPSIsIm12UGFyYW1ldGVyU3B1YyI6IndsQ1Qrc2VpQjBCZzhPeVYlLCJtYXR1cm1hbFNldFN1cm1hbCI6MX0%3D&v=1"
 }
}
```

1. Salin kunci rahasia dan URL payload dari output. Anda membutuhkan mereka untuk menambahkan webhook di GitHub Enterprise Server.
2. Di GitHub Enterprise Server, pilih repositori tempat proyek CodeBuild Anda disimpan. Pilih Pengaturan, pilih Hooks & layanan, dan kemudian pilih Tambahkan webhook.
3. Masukkan URL payload dan kunci rahasia, terima default untuk bidang lainnya, dan kemudian pilih Tambahkan webhook.

## Berhenti menjalankan build secara otomatis (AWS CLI)

Jika kode sumber Anda disimpan dalam repositori GitHub atau GitHub Enterprise Server, Anda dapat mengatur webhook GitHub agar memiliki AWS CodeBuild membangun kembali kode sumber Anda setiap kali perubahan kode didorong ke repositori. Untuk informasi selengkapnya, lihat [Mulai menjalankan build secara otomatis \(AWS CLI\)](#).

Jika Anda telah mengaktifkan perilaku ini, Anda dapat mematikannya dengan menjalankan `delete-webhook` perintah sebagai berikut:

```
aws codebuild delete-webhook --project-name <project-name>
```

- di mana `<project-name>` adalah nama proyek build yang berisi kode sumber yang akan dibangun kembali.

Jika perintah ini berhasil, tidak ada informasi dan tidak ada kesalahan yang muncul dalam output.

**Note**

Ini menghapus webhook dari proyek CodeBuild Anda saja. Anda juga harus menghapus webhook dari repositori GitHub atau GitHub Enterprise Server Anda.

## Jalankan membangun (AWSSDK)

Untuk menggunakan CodePipeline untuk menjalankan build dengan AWS CodeBuild, lewati langkah-langkah ini dan ikuti petunjuk di [Gunakan AWS CodePipeline dengan AWS CodeBuild untuk menguji kode dan menjalankan build](#) sebagai gantinya.

Untuk informasi tentang menggunakan CodeBuild dengan AWSSDK, lihat [AWS Referensi SDK dan alat](#).

## Lihat detail build di AWS CodeBuild

Anda dapat menggunakan AWS CodeBuild konsol, AWS CLI, atau AWSSDK untuk melihat detail tentang build yang dikelola oleh CodeBuild.

### Topik

- [Lihat detail build \(konsol\)](#)
- [Lihat detail build \(AWS CLI\)](#)
- [Lihat detail build \(AWSSDKs\)](#)
- [Transisi fase](#)

### Lihat detail build (konsol)

1. Buka AWS CodeBuild konsol di <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Lakukan salah satu dari berikut:
  - Di panel navigasi, pilih Riwayat pembangunan. Dalam daftar membangun, di Jalankan pembangunankolom, pilih tautan untuk build.
  - Di panel navigasi, pilih Proyek pembangunan. Dalam daftar proyek pembangunan, di Namakolom, pilih link untuk nama proyek build. Kemudian, dalam daftar membangun, di Jalankan pembangunankolom, pilih tautan untuk build.

**Note**

Secara default, hanya 10 proyek build atau build terbaru yang ditampilkan. Untuk melihat proyek build atau build, pilih ikon roda gigi, dan kemudian pilih nilai yang berbedaBuild per halamanatauProyek per halamanatau gunakan panah belakang dan depan.

## Lihat detail build (AWS CLI)

Untuk informasi lebih lanjut tentang penggunaanAWS CLIdenganAWS CodeBuild, lihat[Referensi baris perintah](#).

Jalankan perintah batch-get-builds:

```
aws codebuild batch-get-builds --ids ids
```

Ganti placeholder berikut:

- *ids*: String yang dibutuhkan. Satu atau lebih ID build untuk melihat detail tentang. Untuk menentukan lebih dari satu ID build, pisahkan setiap ID build dengan spasi. Anda dapat menentukan hingga 100 ID build. Untuk mendapatkan daftar ID build, lihat topik berikut:
  - [Melihat daftar ID build \(AWS CLI\)](#)
  - [Melihat daftar ID build untuk proyek build \(AWS CLI\)](#)

Misalnya, jika Anda menjalankan perintah ini:

```
aws codebuild batch-get-builds --ids codebuild-demo-project:e9c4f4df-3f43-41d2-ab3a-60fe2EXAMPLE codebuild-demo-project:815e755f-bade-4a7e-80f0-efe51EXAMPLE my-other-project:813bb6c6-891b-426a-9dd7-6d8a3EXAMPLE
```

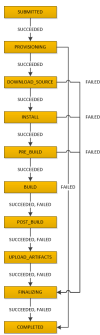
Jika perintah berhasil, data yang mirip dengan yang dijelaskan di[Untuk melihat informasi build yang dirangkum](#) muncul dalam output.

## Lihat detail build (AWSSDKs)

Untuk informasi lebih lanjut tentang penggunaanAWS CodeBuilddenganAWSSDK, lihat[AWSReferensi SDK dan alat](#).

## Transisi fase

Build diAWS CodeBuildmelanjutkan secara bertahap:



### **⚠ Important**

Parameter `UPLOAD_ARTIFACTS` selalu berusaha, bahkan jika `BUILD` fase gagal.

## Melihat daftar ID build diAWS CodeBuild

Anda dapat menggunakanAWS CodeBuildkonsol,AWS CLI, atauAWSSDK untuk melihat daftar ID build untuk build yang dikelola oleh CodeBuild.

### Topik

- [Melihat daftar ID build \(console\)](#)
- [Melihat daftar ID build \(AWS CLI\)](#)
- [Melihat daftar ID build batch \(AWS CLI\)](#)
- [Melihat daftar ID build \(AWSSDK\)](#)

### Melihat daftar ID build (console)

1. BukaAWS CodeBuildkonsol di<https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Di panel navigasi, pilihRiwayat bangunan.



**Note**

Secara default, hanya 10 build terbaru yang ditampilkan. Untuk melihat versi lainnya, pilih ikon roda gigi, dan kemudian pilih nilai yang berbeda. Membangun per halaman atau gunakan panah belakang dan depan.

## Melihat daftar ID build (AWS CLI)

Untuk informasi selengkapnya tentang penggunaan AWS CLI dengan CodeBuild, lihat [Referensi baris perintah](#).

- Jalankan perintah list-builds:

```
aws codebuild list-builds --sort-order sort-order --next-token next-token
```

Di perintah sebelumnya, ganti placeholder berikut:

- sort-order*: String opsional yang digunakan untuk menunjukkan bagaimana daftar ID build. Nilai yang valid mencakup ASCENDING dan DESCENDING.
- Token berikutnya*: String opsional. Selama menjalankan sebelumnya, jika ada lebih dari 100 item dalam daftar, hanya 100 item pertama yang dikembalikan, bersama dengan string unik yang disebut Token berikutnya. Untuk mendapatkan batch item berikutnya dalam daftar, jalankan perintah ini lagi, tambahkan token berikutnya ke panggilan. Untuk mendapatkan semua item dalam daftar, terus jalankan perintah ini dengan setiap token berikutnya berikutnya, sampai tidak ada lagi token berikutnya dikembalikan.

Misalnya, jika Anda menjalankan perintah ini:

```
aws codebuild list-builds --sort-order ASCENDING
```

Hasil yang mirip dengan berikut ini mungkin muncul dalam output:

```
{
 "nextToken": "4AEA6u7J...The full token has been omitted for brevity...MzY20A==",
 "ids": [
 "codebuild-demo-project:815e755f-bade-4a7e-80f0-efe51EXAMPLE"
]
}
```

```
"codebuild-demo-project:84a7f3d1-d40e-4956-b4cf-7a9d4EXAMPLE"
... The full list of build IDs has been omitted for brevity ...
"codebuild-demo-project:931d0b72-bf6f-4040-a472-5c707EXAMPLE"
]
}
```

Jika Anda menjalankan perintah ini lagi:

```
aws codebuild list-builds --sort-order ASCENDING --next-token 4AEA6u7J...The full
token has been omitted for brevity...MzY20A==
```

Hasil yang mirip dengan berikut ini mungkin muncul dalam output:

```
{
 "ids": [
 "codebuild-demo-project:49015049-21cf-4b50-9708-df115EXAMPLE",
 "codebuild-demo-project:543e7206-68a3-46d6-a4da-759abEXAMPLE",
 ... The full list of build IDs has been omitted for brevity ...
 "codebuild-demo-project:c282f198-4582-4b38-bdc0-26f96EXAMPLE"
]
}
```

## Melihat daftar ID build batch (AWS CLI)

Untuk informasi selengkapnya tentang penggunaan AWS CLI dengan CodeBuild, lihat [Referensi baris perintah](#).

- Jalankan perintah `list-build-batches`:

```
aws codebuild list-build-batches --sort-order sort-order --next-token next-token
```

Di perintah sebelumnya, ganti placeholder berikut:

- ***sort-order***: String opsional yang digunakan untuk menunjukkan bagaimana daftar ID batch build. Nilai yang valid mencakup ASCENDING dan DESCENDING.
- ***Token berikutnya***: String opsional. Selama menjalankan sebelumnya, jika ada lebih dari 100 item dalam daftar, hanya 100 item pertama yang dikembalikan, bersama dengan string unik yang disebut `Token berikutnya`. Untuk mendapatkan batch item berikutnya dalam daftar, jalankan perintah ini lagi, tambahkan token berikutnya ke panggilan. Untuk

mendapatkan semua item dalam daftar, terus jalankan perintah ini dengan setiap token berikutnya berikutnya, sampai tidak ada lagi token berikutnya dikembalikan.

Misalnya, jika Anda menjalankan perintah ini:

```
aws codebuild list-build-batches --sort-order ASCENDING
```

Hasil yang mirip dengan berikut ini mungkin muncul dalam output:

```
{
 "nextToken": "4AEA6u7J...The full token has been omitted for brevity...MzY20A==",
 "ids": [
 "codebuild-demo-project:815e755f-bade-4a7e-80f0-efe51EXAMPLE",
 "codebuild-demo-project:84a7f3d1-d40e-4956-b4cf-7a9d4EXAMPLE",
 ... The full list of build IDs has been omitted for brevity ...
 "codebuild-demo-project:931d0b72-bf6f-4040-a472-5c707EXAMPLE"
]
}
```

Jika Anda menjalankan perintah ini lagi:

```
aws codebuild list-build-batches --sort-order ASCENDING --next-token 4AEA6u7J...The full token has been omitted for brevity...MzY20A==
```

Hasil yang mirip dengan berikut ini mungkin muncul dalam output:

```
{
 "ids": [
 "codebuild-demo-project:49015049-21cf-4b50-9708-df115EXAMPLE",
 "codebuild-demo-project:543e7206-68a3-46d6-a4da-759abEXAMPLE",
 ... The full list of build IDs has been omitted for brevity ...
 "codebuild-demo-project:c282f198-4582-4b38-bdc0-26f96EXAMPLE"
]
}
```

## Melihat daftar ID build (AWSSDK)

Untuk informasi selengkapnya tentang penggunaan CodeBuild dengan AWSSDK, lihat [AWSReferensi SDK dan alat](#).

## Melihat daftar ID build untuk proyek build diAWS CodeBuild

Anda dapat menggunakanAWS CodeBuildkonsol,AWS CLI, atauAWSSDK untuk melihat daftar ID build untuk proyek build di CodeBuild.

### Topik

- [Melihat daftar ID build untuk proyek build \(console\)](#)
- [Melihat daftar ID build untuk proyek build \(AWS CLI\)](#)
- [Melihat daftar ID build batch untuk proyek build \(AWS CLI\)](#)
- [Melihat daftar ID build untuk proyek build \(AWSSDK\)](#)

### Melihat daftar ID build untuk proyek build (console)

1. Buka konsol CodeBuild di <https://console.aws.amazon.com/codebuild/>.
2. Di panel navigasi, pilihBuat proyek. Dalam daftar proyek pembangunan, diNamakolom, pilih proyek build.

#### Note

Secara default, hanya proyek build atau build 100 terbaru yang ditampilkan. Untuk melihat proyek build atau build, pilih ikon roda gigi, dan kemudian pilih nilai yang berbedaMembangun per halamanatauProyek per halamanatau gunakan panah belakang dan depan.

### Melihat daftar ID build untuk proyek build (AWS CLI)

Untuk informasi lebih lanjut tentang penggunaanAWS CLIbersamaAWS CodeBuild, lihat[Referensi baris perintah](#).

Jalankanlist-builds-for-projectperintah, sebagai berikut:

```
aws codebuild list-builds-for-project --project-name project-name --sort-order sort-order --next-token next-token
```

Di perintah sebelumnya, ganti placeholder berikut:

- **Proyek-nama**: String yang diperlukan digunakan untuk menunjukkan nama proyek build untuk daftar membangun ID untuk. Untuk mendapatkan daftar proyek pembangunan, lihat [Melihat daftar nama proyek build \(AWS CLI\)](#).
- **sort-order**: String opsional yang digunakan untuk menunjukkan bagaimana daftar ID build. Nilai yang valid mencakup ASCENDING dan DESCENDING.
- **Token berikutnya**: String opsional. Selama menjalankan sebelumnya, jika ada lebih dari 100 item dalam daftar, hanya 100 item pertama yang dikembalikan, bersama dengan string unik yang disebut Token berikutnya. Untuk mendapatkan batch item berikutnya dalam daftar, jalankan perintah ini lagi, tambahkan token berikutnya ke panggilan. Untuk mendapatkan semua item dalam daftar, terus jalankan perintah ini dengan setiap token berikutnya berikutnya yang dikembalikan, sampai tidak ada lagi token berikutnya dikembalikan.

Misalnya, jika Anda menjalankan perintah ini mirip dengan ini:

```
aws codebuild list-builds-for-project --project-name codebuild-demo-project --sort-order ASCENDING
```

Hasil seperti berikut mungkin muncul dalam output:

```
{
 "nextToken": "4AEA6u7J...The full token has been omitted for brevity...MzY2OA==",
 "ids": [
 "codebuild-demo-project:9b175d16-66fd-4e71-93a0-50a08EXAMPLE"
 "codebuild-demo-project:a9d1bd09-18a2-456b-8a36-7d65aEXAMPLE"
 ... The full list of build IDs has been omitted for brevity ...
 "codebuild-demo-project:fe70d102-c04f-421a-9cfa-2dc15EXAMPLE"
]
}
```

Jika Anda menjalankan perintah ini lagi:

```
aws codebuild list-builds-for-project --project-name codebuild-demo-project --sort-order ASCENDING --next-token 4AEA6u7J...The full token has been omitted for brevity...MzY2OA==
```

Anda mungkin melihat hasil seperti hal-hal berikut di output:

```
{
```

```
"ids": [
 "codebuild-demo-project:98253670-7a8a-4546-b908-dc890EXAMPLE"
 "codebuild-demo-project:ad5405b2-1ab3-44df-ae2d-fba84EXAMPLE"
 ... The full list of build IDs has been omitted for brevity ...
 "codebuild-demo-project:f721a282-380f-4b08-850a-e0ac1EXAMPLE"
]
}
```

## Melihat daftar ID build batch untuk proyek build (AWS CLI)

Untuk informasi lebih lanjut tentang penggunaan AWS CLI bersama AWS CodeBuild, lihat [Referensi baris perintah](#).

Jalankan `list-build-batches-for-project` perintah, sebagai berikut:

```
aws codebuild list-build-batches-for-project --project-name project-name --sort-order sort-order --next-token next-token
```

Di perintah sebelumnya, ganti placeholder berikut:

- ***Proyek-nama***: String yang diperlukan digunakan untuk menunjukkan nama proyek build untuk daftar membangun ID untuk. Untuk mendapatkan daftar proyek pembangunan, lihat [Melihat daftar nama proyek build \(AWS CLI\)](#).
- ***sort-order***: String opsional yang digunakan untuk menunjukkan bagaimana daftar ID build. Nilai yang valid mencakup ASCENDING dan DESCENDING.
- ***Token berikutnya***: String opsional. Selama menjalankan sebelumnya, jika ada lebih dari 100 item dalam daftar, hanya 100 item pertama yang dikembalikan, bersama dengan string unik yang disebut Token berikutnya. Untuk mendapatkan batch item berikutnya dalam daftar, jalankan perintah ini lagi, tambahkan token berikutnya ke panggilan. Untuk mendapatkan semua item dalam daftar, terus jalankan perintah ini dengan setiap token berikutnya berikutnya yang dikembalikan, sampai tidak ada lagi token berikutnya dikembalikan.

Misalnya, jika Anda menjalankan perintah ini mirip dengan ini:

```
aws codebuild list-build-batches-for-project --project-name codebuild-demo-project --sort-order ASCENDING
```

Hasil seperti berikut mungkin muncul dalam output:

```
{
 "nextToken": "4AEA6u7J...The full token has been omitted for brevity...MzY20A==",
 "ids": [
 "codebuild-demo-project:9b175d16-66fd-4e71-93a0-50a08EXAMPLE"
 "codebuild-demo-project:a9d1bd09-18a2-456b-8a36-7d65aEXAMPLE"
 ... The full list of build IDs has been omitted for brevity ...
 "codebuild-demo-project:fe70d102-c04f-421a-9cfa-2dc15EXAMPLE"
]
}
```

Jika Anda menjalankan perintah ini lagi:

```
aws codebuild list-build-batches-for-project --project-name codebuild-demo-project
--sort-order ASCENDING --next-token 4AEA6u7J...The full token has been omitted for
brevity...MzY20A==
```

Anda mungkin melihat hasil seperti hal-hal berikut di output:

```
{
 "ids": [
 "codebuild-demo-project:98253670-7a8a-4546-b908-dc890EXAMPLE"
 "codebuild-demo-project:ad5405b2-1ab3-44df-ae2d-fba84EXAMPLE"
 ... The full list of build IDs has been omitted for brevity ...
 "codebuild-demo-project:f721a282-380f-4b08-850a-e0ac1EXAMPLE"
]
}
```

## Melihat daftar ID build untuk proyek build (AWSSDK)

Untuk informasi lebih lanjut tentang penggunaan AWS CodeBuild dengan AWSSDK, lihat [AWS Referensi SDK dan alat](#).

## Hentikan membangun di AWS CodeBuild

Anda dapat menggunakan AWS CodeBuild konsol, AWS CLI, atau AWSSDK untuk menghentikan build di AWS CodeBuild.

### Topik

- [Menghentikan build \(konsol\)](#)

- [Menghentikan membangun \(AWS CLI\)](#)
- [Menghentikan membangun \(AWSSDK\)](#)

## Menghentikan build (konsol)

1. Buka AWS CodeBuild konsol di <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Lakukan salah satu dari berikut:
  - Jika **build-project-nama:ID bangunan** halaman ditampilkan, pilih **Hentikan build**.
  - Di panel navigasi, pilih **Riwayat pembangunan**. Dalam daftar build, pilih kotak untuk build, lalu pilih **Hentikan build**.
  - Di panel navigasi, pilih **Proyek pembangunan**. Dalam daftar proyek pembangunan, di **Nama kolom**, pilih link untuk nama proyek build. Dalam daftar build, pilih kotak untuk build, lalu pilih **Hentikan build**.

### Note

Secara default, hanya proyek build atau build 100 terbaru yang ditampilkan. Untuk melihat proyek build atau build, pilih ikon roda gigi, dan kemudian pilih nilai yang berbeda **Build per halaman** atau **Proyek per halaman** atau gunakan panah belakang dan depan.

Jika AWS CodeBuild tidak dapat berhasil menghentikan build (misalnya, jika proses build sudah selesai), **Berhenti** tombol dinonaktifkan atau mungkin tidak muncul.

## Menghentikan membangun (AWS CLI)

- Jalankan perintah stop-build:

```
aws codebuild stop-build --id id
```

Di perintah sebelumnya, ganti placeholder berikut:

- **id**: String yang dibutuhkan. ID membangun untuk berhenti. Untuk mendapatkan daftar ID build, lihat topik berikut:
  - [Melihat daftar ID build \(AWS CLI\)](#)
  - [Melihat daftar ID build untuk proyek build \(AWS CLI\)](#)



Jika AWS CodeBuild berhasil menghentikan build, `buildStatus` nilai dalam `build` objek dalam output adalah `STOPPED`.

Jika CodeBuild tidak berhasil menghentikan build (misalnya, jika build sudah selesai), maka `buildStatus` nilai dalam `build` objek dalam output adalah status build akhir (misalnya, `SUCCEEDED`).

## Menghentikan membangun (AWSSDK)

Untuk informasi lebih lanjut tentang penggunaan AWS CodeBuild dengan AWSSDK, lihat [AWS Referensi SDK dan alat](#).

## Menghentikan pembuatan batch AWS CodeBuild

Anda dapat menggunakan AWS CodeBuild konsol, AWS CLI, atau AWSSDK untuk menghentikan build batch AWS CodeBuild.

### Topik

- [Menghentikan build batch \(konsol\)](#)
- [Menghentikan pembuatan batch \(AWS CLI\)](#)
- [Menghentikan pembuatan batch \(AWSSDK\)](#)

## Menghentikan build batch (konsol)

1. Buka AWS CodeBuild konsol di <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Lakukan salah satu dari berikut:
  - Jika **`build-project-nama:ID bangunan`** halaman ditampilkan, pilih **Hentikan build**.
  - Di panel navigasi, pilih **Riwayat pembangunan**. Dalam daftar build, pilih kotak untuk build, lalu pilih **Hentikan build**.
  - Di panel navigasi, pilih **Proyek pembangunan**. Dalam daftar proyek pembangunan, di **Nama kolom**, pilih link untuk nama proyek build. Dalam daftar build, pilih kotak untuk build, lalu pilih **Hentikan build**.

**Note**

Secara default, hanya proyek build atau build 100 terbaru yang ditampilkan. Untuk melihat proyek build atau build, pilih ikon roda gigi, dan kemudian pilih nilai yang berbedaBuild per halamanatauProyek per halamanatau gunakan panah belakang dan depan.

JikaAWS CodeBuildtidak dapat berhasil menghentikan build batch (misalnya, jika proses build sudah selesai),Hentikan buildtombol dinonaktifkan.

## Menghentikan pembuatan batch (AWS CLI)

- Jalankan perintah [stop-build-batch](#):

```
aws codebuild stop-build-batch --id <batch-build-id>
```

Di perintah sebelumnya, ganti placeholder berikut:

- <batch-build-id>*: String yang dibutuhkan. Pengenal batch membangun untuk berhenti. Untuk mendapatkan daftar identifikasi build, lihat topik berikut:
  - [Melihat daftar ID build batch \(AWS CLI\)](#)
  - [Melihat daftar ID build batch untuk proyek build \(AWS CLI\)](#)

JikaAWS CodeBuildberhasil menghentikan batch build,buildBatchStatusnilai dalambuildBatchobjek dalam output adalahSTOPPED.

Jika CodeBuild tidak berhasil menghentikan batch build (misalnya, jika batch build sudah selesai),buildBatchStatusnilai dalambuildBatchobjek dalam output adalah status build akhir (misalnya,SUCCEDED).

## Menghentikan pembuatan batch (AWSSDK)

Untuk informasi lebih lanjut tentang penggunaanAWS CodeBuilddenganAWSSDK, lihat[AWSReferensi SDK dan alat](#).

## Coba lagi membangun diAWS CodeBuild

Anda dapat menggunakanAWS CodeBuildkonsol,AWS CLI, atauAWSSDK untuk mencoba kembali membangun tunggal atau membangun batchAWS CodeBuild.

### Topik

- [Coba lagi membangun \(konsol\)](#)
- [Coba lagi membangun \(AWS CLI\)](#)
- [Coba lagi membangun \(AWSSDK\)](#)

### Coba lagi membangun (konsol)

1. BukaAWS CodeBuildkonsol di<https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Lakukan salah satu dari berikut:
  - Jika***build-project-nama:ID bangunan***halaman ditampilkan, pilihCoba lagi membangun.
  - Di panel navigasi, pilihRiwayat membangun. Dalam daftar build, pilih kotak untuk build, lalu pilihCoba lagi membangun.
  - Di panel navigasi, pilihProyek membangun. Dalam daftar proyek pembangunan, diNamakolom, pilih link untuk nama proyek build. Dalam daftar build, pilih kotak untuk build, lalu pilihCoba lagi membangun.

#### Note

Secara default, hanya proyek build atau build 100 terbaru yang ditampilkan. Untuk melihat proyek build atau build, pilih ikon roda gigi, dan kemudian pilih nilai yang berbedaMembangun per halamanatauProyek per halamanatau gunakan panah belakang dan depan.

### Coba lagi membangun (AWS CLI)

- Jalankan perintah `retry-build`:

```
aws codebuild retry-build --id <build-id> --idempotency-token <idempotencyToken>
```

Di perintah sebelumnya, ganti placeholder berikut:

- `<build-id>`: String yang dibutuhkan. ID membangun atau batch membangun untuk mencoba lagi. Untuk mendapatkan daftar ID build, lihat topik berikut:
  - [Melihat daftar ID build \(AWS CLI\)](#)
  - [Melihat daftar ID build batch \(AWS CLI\)](#)
  - [Melihat daftar ID build untuk proyek build \(AWS CLI\)](#)
  - [Melihat daftar ID build batch untuk proyek build \(AWS CLI\)](#)
- `--idempotency-token`: Opsional. Jika Anda menjalankan `retry-build` perintah dengan opsi, pengenal case-sensitive unik, atau token, disertakan dengan `retry-build` permintaan. Token berlaku selama 5 menit setelah permintaan. Jika Anda mengulang `retry-build` permintaan dengan token yang sama, tetapi mengubah parameter, CodeBuild mengembalikan kesalahan parameter ketidakcocokan.

## Coba lagi membangun (AWSSDK)

Untuk informasi lebih lanjut tentang penggunaan AWS CodeBuild dengan AWSSDK, lihat [AWS Referensi SDK dan alat](#).

## Melihat build yang sedang berjalan di Session Manager

Di AWS CodeBuild, Anda dapat menjeda build yang sedang berjalan dan kemudian menggunakan AWS Systems Manager Session Manager untuk menyambung ke container build dan melihat status container.

### Note

Fitur ini tidak tersedia di lingkungan Windows.

### Topik

- [Prasyarat](#)
- [Jeda build](#)
- [Mulai membangun](#)
- [Connect ke container build](#)
- [Lanjutkan build](#)

## Prasyarat

Untuk mengizinkan Session Manager digunakan dengan sesi build, Anda harus mengaktifkan koneksi sesi untuk build. Ada dua prasyarat:

- CodeBuild Gambar kurasi standar Linux sudah menginstal agen SSM dan agen SSM diaktifkan. ContainerMode

Jika Anda menggunakan gambar kustom untuk build Anda, lakukan hal berikut:

1. Instal SSM Agent. Untuk informasi selengkapnya, lihat [Instal Agen SSM secara manual pada instans EC2 untuk Linux di Panduan Pengguna](#). AWS Systems Manager Versi Agen SSM harus 3.0.1295.0 atau yang lebih baru.
2. Salin file <https://github.com/aws/aws-codebuild-docker-images/blob/master/ubuntu/standard/5.0/.json> ke direktori di gambar [amazon-ssm-agent](#) Anda. `/etc/amazon/ssm/` Ini memungkinkan Mode Kontainer di agen SSM.

### Note

Gambar khusus akan membutuhkan agen SSM terbaru agar fitur ini berfungsi seperti yang diharapkan.

- Peran CodeBuild layanan harus memiliki kebijakan SSM berikut:

```
{
 "Effect": "Allow",
 "Action": [
 "ssmmessages:CreateControlChannel",
 "ssmmessages:CreateDataChannel",
 "ssmmessages:OpenControlChannel",
 "ssmmessages:OpenDataChannel"
],
 "Resource": "*"
}
```

CodeBuild Konsol dapat melampirkan kebijakan ini secara otomatis ke peran layanan saat memulai pembuatan. Atau, Anda dapat melampirkan kebijakan ini ke peran layanan Anda secara manual.

- Jika aktivitas sesi Audit dan logging diaktifkan di preferensi Systems Manager, peran CodeBuild layanan juga harus memiliki izin tambahan. Izin berbeda, tergantung di mana log disimpan.

## CloudWatch Log

Jika menggunakan CloudWatch Log untuk menyimpan log Anda, tambahkan izin berikut ke peran CodeBuild layanan:

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": "logs:DescribeLogGroups",
 "Resource": "arn:aws:logs:<region-id>:<account-id>:log-group:*:*"
 },
 {
 "Effect": "Allow",
 "Action": [
 "logs:CreateLogStream",
 "logs:PutLogEvents"
],
 "Resource": "arn:aws:logs:<region-id>:<account-id>:log-group:<log-group-name>:*"
 }
]
}
```

## Amazon S3

Jika menggunakan Amazon S3 untuk menyimpan log Anda, tambahkan izin berikut ke peran CodeBuild layanan:

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": [
 "s3:GetEncryptionConfiguration",
 "s3:PutObject"
],
 "Resource": [
 "arn:aws:s3:::<bucket-name>",
 "arn:aws:s3:::<bucket-name>/*"
]
 }
]
}
```

```
]
 }
]
}
```

Untuk informasi selengkapnya, lihat [Aktivitas sesi audit dan pencatatan](#) di Panduan AWS Systems Manager Pengguna.

## Jeda build

Untuk menjeda build, masukkan `codebuild-breakpoint` perintah di salah satu fase build di file `buildspec` Anda. Build akan dijeda pada titik ini, yang memungkinkan Anda terhubung ke container build dan melihat container dalam statusnya saat ini.

Misalnya, tambahkan yang berikut ini ke fase build di file `buildspec` Anda.

```
phases:
 pre_build:
 commands:
 - echo Entered the pre_build phase...
 - echo "Hello World" > /tmp/hello-world
 - codebuild-breakpoint
```

Kode ini membuat `/tmp/hello-world` file dan kemudian menjeda build pada saat ini.

## Mulai membangun

Agar Session Manager dapat digunakan dengan sesi build, Anda harus mengaktifkan koneksi sesi untuk build. Untuk melakukan ini, saat memulai build, ikuti langkah-langkah berikut:

1. Buka AWS CodeBuild konsol di <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Di panel navigasi, pilih Membangun proyek. Pilih proyek build, lalu pilih Start build with overrides.
3. Pilih Penggantian build lanjutan.
4. Di bagian Lingkungan, pilih opsi Aktifkan koneksi sesi. Jika opsi ini tidak dipilih, semua `codebuild-resume` perintah `codebuild-breakpoint` dan diabaikan.
5. Buat perubahan lain yang diinginkan, dan pilih Mulai membangun.
6. Pantau status build di konsol. Saat sesi tersedia, tautan Pengelola AWS Sesi akan muncul di bagian Status build.

## Connect ke container build

Anda dapat terhubung ke kontainer build dengan salah satu dari dua cara:

### CodeBuild konsol

Di browser web, buka tautan AWS Session Manager untuk terhubung ke container build. Sesi terminal terbuka yang memungkinkan Anda menelusuri dan mengontrol container build.

### AWS CLI

#### Note

Mesin lokal Anda harus menginstal plugin Session Manager untuk prosedur ini. Untuk informasi selengkapnya, lihat [Menginstal Plugin Session Manager untuk AWS CLI](#) di AWS Systems Manager Panduan Pengguna.

1. Panggil batch-get-builds api dengan ID build untuk mendapatkan informasi tentang build, termasuk pengenalan target sesi. Nama properti pengidentifikasi target sesi bervariasi tergantung pada jenis output aws perintah. Inilah `--output json` sebabnya mengapa ditambahkan ke perintah.

```
aws codebuild batch-get-builds --ids <buildID> --region <region> --output json
```

2. Salin nilai `sessionTarget` properti. Nama `sessionTarget` properti dapat bervariasi tergantung pada jenis output aws perintah. Inilah `--output json` sebabnya mengapa ditambahkan ke perintah di langkah sebelumnya.
3. Gunakan perintah berikut untuk menyambung ke container build.

```
aws ssm start-session --target <sessionTarget> --region <region>
```

Untuk contoh ini, verifikasi bahwa `/tmp/hello-world` file itu ada dan berisi teks `Hello World`.

## Lanjutkan build

Setelah Anda selesai memeriksa container build, keluarkan `codebuild-resume` perintah dari shell container.



```
$ codebuild-resume
```

## Hapus build diAWS CodeBuild

Anda dapat menggunakanAWS CLIatauAWSSDK untuk menghapus build diAWS CodeBuild.

### Menghapus build (AWS CLI)

Jalankan perintah `batch-delete-builds`:

```
aws codebuild batch-delete-builds --ids ids
```

Di perintah sebelumnya, ganti placeholder berikut:

- *ids*: String yang dibutuhkan. ID dari build untuk menghapus. Untuk menentukan beberapa build, pisahkan setiap ID build dengan spasi. Untuk mendapatkan daftar build ID, lihat topik berikut:
  - [Melihat daftar ID build \(AWS CLI\)](#)
  - [Melihat daftar ID build untuk proyek build \(AWS CLI\)](#)

Jika berhasil,`buildsDeletedArray` akan muncul di dalam output, berisi Amazon Resource Name (ARN) dari setiap build yang berhasil dihapus. Informasi tentang build yang tidak berhasil dihapus muncul dalam output dalam`buildsNotDeletedArray`.

Misalnya, jika Anda menjalankan perintah ini:

```
aws codebuild batch-delete-builds --ids my-demo-build-project:f8b888d2-5e1e-4032-8645-b115195648EX my-other-demo-build-project:a18bc6ee-e499-4887-b36a-8c90349c7eEX
```

Informasi yang mirip dengan berikut akan muncul di dalam output:

```
{
 "buildsNotDeleted": [
 {
 "id": "arn:aws:codebuild:us-west-2:123456789012:build/my-demo-build-project:f8b888d2-5e1e-4032-8645-b115195648EX",
 "statusCode": "BUILD_IN_PROGRESS"
 }
],
 "buildsDeleted": [
```

```
"arn:aws:codebuild:us-west-2:123456789012:build/my-other-demo-build-
project:a18bc6ee-e499-4887-b36a-8c90349c7eEX"
]
}
```

## Menghapus build (AWSSDK)

Untuk informasi tentang penggunaan AWS CodeBuild dengan AWSSDK, lihat [AWS Referensi SDK dan alat](#).

# Bekerja dengan AWS Lambda komputasi di AWS CodeBuild

AWS Lambda compute menawarkan kecepatan start-up yang dioptimalkan untuk build Anda.

AWS Lambda mendukung build yang lebih cepat karena latensi start-up yang lebih rendah.

AWS Lambda juga secara otomatis menskalakan, jadi build tidak menunggu dalam antrian untuk dijalankan. Namun, ada beberapa kasus penggunaan yang AWS Lambda tidak mendukung, dan jika berdampak pada Anda, gunakan komputasi EC2. Untuk informasi selengkapnya, lihat [Keterbatasan AWS Lambda komputasi](#).

## Topik

- [Alat dan runtime mana yang akan disertakan dalam gambar docker lingkungan runtime yang dikuratori yang berjalan? AWS Lambda](#)
- [Wilayah mana yang mendukung AWS Lambda komputasi? CodeBuild](#)
- [Keterbatasan AWS Lambda komputasi](#)
- [AWS Lambda menghitung sampel dengan AWS CodeBuild](#)

## Alat dan runtime mana yang akan disertakan dalam gambar docker lingkungan runtime yang dikuratori yang berjalan? AWS Lambda

AWS Lambda mendukung alat-alat berikut: AWS CLI v2, AWS SAM CLI, git, go, Java, Node.js, Python, pip, Ruby, dan .NET.

## Wilayah mana yang mendukung AWS Lambda komputasi?

### CodeBuild

Dalam CodeBuild, AWS Lambda komputasi didukung sebagai berikut Wilayah AWS: AS Timur (Virginia N.), AS Timur (Ohio), AS Barat (Oregon), Asia Pasifik (Mumbai), Asia Pasifik (Singapura), Asia Pasifik (Sydney), Asia Pasifik (Tokyo), Eropa (Frankfurt), Eropa (Irlandia), dan Amerika Selatan (São Paulo). Untuk informasi selengkapnya tentang Wilayah AWS tempat CodeBuild tersedia, lihat [AWS Layanan menurut Wilayah](#).

## Keterbatasan AWS Lambda komputasi

Ada beberapa kasus penggunaan yang AWS Lambda tidak mendukung, dan jika memengaruhi Anda, gunakan komputasi EC2:

- AWS Lambdatidak mendukung alat yang memerlukan izin root. Untuk alat seperti yum atau rpm, gunakan jenis komputasi EC2 atau alat lain yang tidak memerlukan izin root.
- AWS Lambdatidak mendukung build atau run Docker. Anda dapat menggunakan alternatif yang tidak memerlukan izin root, seperti Podman.
- AWS Lambdatidak mendukung penulisan ke file di luar/tmp. Manajer paket yang disertakan dikonfigurasi untuk menggunakan /tmp direktori secara default untuk mengunduh dan mereferensikan paket.
- AWS Lambdatidak mendukung jenis lingkungan LINUX\_GPU\_CONTAINER dan tidak didukung pada Windows Server Core 2019.
- AWS Lambdatidak mendukung caching, build batch, batas waktu build kustom, batas waktu antrian, lencana build, mode istimewa, lingkungan runtime kustom, atau runtime lebih dari 15 menit.
- AWS Lambdatidak mendukung konektivitas VPC, rentang tetap alamat IP CodeBuild sumber, EFS, versi semantik, menginstal sertifikat, atau akses SSH dengan Session Manager.

## AWS Lambdamenghitung sampel dengan AWS CodeBuild

Kelompok sampel ini dapat digunakan untuk bereksperimen dengan AWS Lambda compute in CodeBuild.

Topik

- [Menyebarkan fungsi Lambda menggunakan AWS SAM dengan CodeBuild Lambda Java](#)
- [Buat aplikasi React satu halaman dengan CodeBuild Lambda Node.js](#)
- [Perbarui konfigurasi fungsi Lambda dengan CodeBuild Lambda Python](#)

## Menyebarkan fungsi Lambda menggunakan AWS SAM dengan CodeBuild Lambda Java

The AWS Serverless Application Model (AWS SAM) adalah kerangka kerja sumber terbuka untuk membangun aplikasi tanpa server. Untuk informasi selengkapnya, lihat [AWS Serverless Application](#)

[Modelrepositori](#) di. GitHub Contoh Java berikut menggunakan Gradle untuk membangun dan menguji AWS Lambda fungsi. Setelah itu, AWS SAM CLI digunakan untuk menyebarkan AWS CloudFormation template dan bundel penerapan. Dengan menggunakan CodeBuild Lambda, langkah-langkah pembuatan, pengujian, dan penerapan semuanya ditangani secara otomatis, memungkinkan infrastruktur diperbarui dengan cepat tanpa intervensi manual dalam satu build.

## Siapkan AWS SAM repositori Anda

Buat AWS SAM Hello World proyek menggunakan AWS SAM CLI.

Untuk membuat AWS SAM Proyek Anda

1. Ikuti petunjuk di Panduan AWS Serverless Application Model Pengembang untuk [Menginstal AWS SAM CLI di mesin](#) lokal Anda.
2. Jalankan `sam init` dan pilih konfigurasi proyek berikut.

```
Which template source would you like to use?: 1 - AWS Quick Start Templates
Choose an AWS Quick Start application template: 1 - Hello World Example
Use the most popular runtime and package type? (Python and zip) [y/N]: N
Which runtime would you like to use?: 8 - java21
What package type would you like to use?: 1 - Zip
Which dependency manager would you like to use?: 1 - gradle
Would you like to enable X-Ray tracing on the function(s) in your application? [y/N]: N
Would you like to enable monitoring using CloudWatch Application Insights? [y/N]: N
Would you like to set Structured Logging in JSON format on your Lambda functions? [y/N]: N
Project name [sam-app]: <insert project name>
```

3. Unggah folder AWS SAM proyek ke repositori sumber yang didukung. Untuk daftar jenis sumber yang didukung, lihat [ProjectSource](#).

## Buat proyek CodeBuild Lambda Java

Buat proyek AWS CodeBuild Lambda Java dan siapkan izin IAM yang diperlukan untuk build.

Untuk membuat proyek CodeBuild Lambda Java Anda

1. Buka AWS CodeBuild konsol di <https://console.aws.amazon.com/codesuite/codebuild/home>.

2. Jika halaman CodeBuild informasi ditampilkan, pilih Buat proyek build. Jika tidak, pada panel navigasi, perluas Build, pilih Build projects, lalu pilih Create build project.
3. Di Nama proyek, masukkan nama untuk proyek pembangunan ini. Membangun nama proyek harus unik di setiap AWS akun. Anda juga dapat menyertakan deskripsi opsional proyek build untuk membantu pengguna lain memahami tujuan proyek ini.
4. Di Sumber, pilih repositori sumber tempat AWS SAM proyek Anda berada.
5. Di Lingkungan:
  - Untuk Compute, pilih Lambda.
  - Untuk Runtime, pilih Java.
  - Untuk Gambar, pilih `aws/codebuild/amazonlinux-x86_64-lambda-standard:corretto21`.
  - Untuk peran Layanan, biarkan peran Layanan baru dipilih. Catat nama Peran. Ini akan diperlukan saat Anda memperbarui izin IAM proyek nanti dalam sampel ini.
6. Pilih Buat proyek build.
7. Buka konsol IAM di <https://console.aws.amazon.com/iam/>.
8. Di panel navigasi, pilih Peran dan pilih peran layanan yang terkait dengan proyek Anda. Anda dapat menemukan peran proyek Anda CodeBuild dengan memilih proyek build, memilih Edit, Lingkungan, dan kemudian peran Layanan.
9. Pilih tab Trust relationship, lalu pilih Edit trust policy.
10. Tambahkan kebijakan inline berikut ke peran IAM Anda. Ini akan digunakan untuk menyebarkan AWS SAM infrastruktur Anda nanti. Untuk informasi lebih lanjut, lihat [Menambahkan dan menghapus izin identitas IAM](#) dalam Panduan Pengguna IAM.

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Sid": "",
 "Effect": "Allow",
 "Action": [
 "cloudformation:*",
 "lambda:*",
 "iam:*",
 "apigateway:*",
 "s3:*"
],
 "Resource": [
```

```
 "*"
]
}
]
```

## Siapkan proyek buildspec

Untuk membangun, menguji, dan menerapkan fungsi Lambda Anda CodeBuild , membaca dan mengeksekusi perintah build dari buildspec.

Untuk menyiapkan buildspec proyek Anda

1. Di CodeBuild konsol, pilih proyek build Anda, lalu pilih Edit dan Buildspec.
2. Di Buildspec, pilih Sisipkan perintah build dan kemudian Beralih ke editor.
3. Hapus perintah build yang telah diisi sebelumnya dan tempel di buildspec berikut.

```
version: 0.2
env:
 variables:
 GRADLE_DIR: "HelloWorldFunction"
phases:
 build:
 commands:
 - echo "Running unit tests..."
 - cd $GRADLE_DIR; gradle test; cd ..
 - echo "Running build..."
 - sam build --template-file template.yaml
 - echo "Running deploy..."
 - sam package --output-template-file packaged.yaml --resolve-s3 --template-
file template.yaml
 - yes | sam deploy
```

4. Pilih Perbarui buildspec.

## Terapkan infrastruktur AWS SAM Lambda Anda

Gunakan CodeBuild Lambda untuk menyebarkan infrastruktur Lambda Anda secara otomatis

## Untuk menyebarkan infrastruktur Lambda Anda

1. Pilih Mulai membangun. Ini akan secara otomatis membangun, menguji, dan menyebarkan AWS SAM aplikasi Anda untuk AWS Lambda digunakan AWS CloudFormation.
2. Setelah build selesai, navigasikan ke AWS Lambda konsol dan cari fungsi Lambda baru Anda di bawah nama AWS SAM proyek.
3. Uji fungsi Lambda Anda dengan memilih API Gateway di bawah ikhtisar Fungsi, lalu mengklik URL titik akhir API. Anda akan melihat halaman terbuka dengan pesan "message": "hello world".

## Bersihkan infrastruktur Anda

Untuk menghindari biaya lebih lanjut untuk sumber daya yang Anda gunakan selama tutorial ini, hapus sumber daya yang dibuat oleh AWS SAM template Anda dan CodeBuild.

### Untuk membersihkan infrastruktur Anda

1. Arahkan ke AWS CloudFormation konsol dan pilih `aws-sam-cli-managed-default`.
2. Di Resources, kosongkan bucket `SamCliSourceBucket` penerapan.
3. Hapus `aws-sam-cli-managed-default` tumpukan.
4. Hapus AWS CloudFormation tumpukan yang terkait dengan AWS SAM proyek Anda. Tumpukan ini harus memiliki nama yang sama dengan AWS SAM proyek Anda.
5. Arahkan ke CloudWatch konsol dan hapus grup CloudWatch log yang terkait dengan CodeBuild proyek Anda.
6. Arahkan ke CodeBuild konsol dan hapus CodeBuild project Anda dengan memilih Delete build project.

## Buat aplikasi React satu halaman dengan CodeBuild Lambda Node.js

[Create React App](#) adalah cara untuk membuat aplikasi React satu halaman. Contoh Node.js berikut menggunakan Node.js untuk membangun artefak sumber dari Create React App dan mengembalikan artefak build.

### Siapkan ember repositori dan artefak sumber

Buat repositori sumber untuk proyek Anda menggunakan yarn dan Create React App.



Untuk mengatur repositori sumber dan ember artefak

1. Di mesin lokal Anda, jalankan `yarn create react-app <app-name>` untuk membuat aplikasi React sederhana.
2. Unggah folder proyek aplikasi React ke repositori sumber yang didukung. Untuk daftar jenis sumber yang didukung, lihat [ProjectSource](#).

## Buat proyek CodeBuild Lambda Node.js

Buat proyek AWS CodeBuild Lambda Node.js.

Untuk membuat proyek CodeBuild Lambda Node.js

1. Buka AWS CodeBuild konsol di <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Jika halaman CodeBuild informasi ditampilkan, pilih Buat proyek build. Jika tidak, pada panel navigasi, perluas Build, pilih Build projects, lalu pilih Create build project.
3. Di Nama proyek, masukkan nama untuk proyek pembangunan ini. Membangun nama proyek harus unik di setiap AWS akun. Anda juga dapat menyertakan deskripsi opsional proyek build untuk membantu pengguna lain memahami tujuan proyek ini.
4. Di Sumber, pilih repositori sumber tempat AWS SAM proyek Anda berada.
5. Di Lingkungan:
  - Untuk Compute, pilih Lambda.
  - Untuk Runtime, pilih Node.js.
  - Untuk Gambar, pilih `aws/codebuild/amazonlinux-x86_64-lambda-standard:nodejs20`.
6. Dalam Artefak:
  - Untuk Jenis, pilih Amazon S3.
  - Untuk nama Bucket, pilih bucket artefak proyek yang Anda buat sebelumnya.
  - Untuk kemasan Artefak, pilih Zip.
7. Pilih Buat proyek build.

## Siapkan proyek buildspec

Untuk membangun aplikasi React Anda, CodeBuild membaca dan mengeksekusi perintah build dari file buildspec.

## Untuk menyiapkan buildspec proyek Anda

1. Di CodeBuild konsol, pilih proyek build Anda, lalu pilih Edit dan Buildspec.
2. Di Buildspec, pilih Sisipkan perintah build dan kemudian Beralih ke editor.
3. Hapus perintah build yang telah diisi sebelumnya dan tempel di buildspec berikut.

```
version: 0.2
phases:
 build:
 commands:
 - yarn
 - yarn add --dev jest-junit @babel/plugin-proposal-private-property-in-object
 - yarn run build
 - yarn run test -- --coverage --watchAll=false --testResultsProcessor="jest-junit" --detectOpenHandles
artifacts:
 name: "build-output"
 files:
 - "**/*"
reports:
 test-report:
 files:
 - 'junit.xml'
 file-format: 'JUNITXML'
 coverage-report:
 files:
 - 'coverage/clover.xml'
 file-format: 'CLOVERXML'
```

4. Pilih Perbarui buildspec.

## Membangun dan menjalankan aplikasi React Anda

Bangun aplikasi React di CodeBuild Lambda, unduh artefak build, dan jalankan aplikasi React secara lokal.

Untuk membangun dan menjalankan aplikasi React Anda

1. Pilih Mulai membangun.
2. Setelah build selesai, navigasikan ke bucket artefak proyek Amazon S3 Anda dan unduh artefak aplikasi React.

3. Buka zip artefak build React dan run `npm install -g serve && serve -s build` di folder proyek.
4. `serve`Perintah akan melayani situs statis pada port lokal dan mencetak output ke terminal Anda. Anda dapat mengunjungi URL `localhost Local`: di bawah output terminal untuk melihat aplikasi React Anda.

Untuk mempelajari lebih lanjut tentang cara menangani deployment untuk server berbasis React, lihat [Membuat Penerapan Aplikasi React](#).

## Bersihkan infrastruktur Anda

Untuk menghindari biaya lebih lanjut untuk sumber daya yang Anda gunakan selama tutorial ini, hapus sumber daya yang dibuat untuk CodeBuild proyek Anda.

Untuk membersihkan infrastruktur Anda

1. Hapus artefak proyek Anda Amazon S3 bucket
2. Arahkan ke CloudWatch konsol dan hapus grup CloudWatch log yang terkait dengan CodeBuild proyek Anda.
3. Arahkan ke CodeBuild konsol dan hapus CodeBuild project Anda dengan memilih Delete build project.

## Perbarui konfigurasi fungsi Lambda dengan CodeBuild Lambda Python

Contoh Python berikut menggunakan [Boto3](#) dan Lambda Python untuk memperbarui konfigurasi fungsi CodeBuild Lambda. Sampel ini dapat diperluas untuk mengelola AWS sumber daya lain secara terprogram. Untuk informasi selengkapnya, lihat [dokumentasi Boto3](#).

### Prasyarat

Buat atau temukan fungsi Lambda di akun Anda.

Contoh ini mengasumsikan bahwa Anda telah membuat fungsi Lambda di akun Anda dan akan CodeBuild digunakan untuk memperbarui variabel lingkungan fungsi Lambda. Untuk informasi selengkapnya tentang pengaturan fungsi Lambda CodeBuild, lihat [Menyebarkan fungsi Lambda menggunakan AWS SAM dengan CodeBuild Lambda Java](#) sampel atau kunjungi [AWS Lambda](#)

## Siapkan repositori sumber Anda

Buat repositori sumber untuk menyimpan skrip python Boto3 Anda.

Untuk mengatur repositori sumber

1. Salin script python berikut ke file baru bernama `update_lambda_environment_variables.py`.

```
import boto3
from os import environ

def update_lambda_env_variable(lambda_client):
 lambda_function_name = environ['LAMBDA_FUNC_NAME']
 lambda_env_variable = environ['LAMBDA_ENV_VARIABLE']
 lambda_env_variable_value = environ['LAMBDA_ENV_VARIABLE_VALUE']
 print("Updating lambda function " + lambda_function_name + " environment
variable "
 + lambda_env_variable + " to " + lambda_env_variable_value)
 lambda_client.update_function_configuration(
 FunctionName=lambda_function_name,
 Environment={
 'Variables': {
 lambda_env_variable: lambda_env_variable_value
 }
 },
)

if __name__ == "__main__":
 region = environ['AWS_REGION']
 client = boto3.client('lambda', region)
 update_lambda_env_variable(client)
```

2. Unggah file python ke repositori sumber yang didukung. Untuk daftar jenis sumber yang didukung, lihat [ProjectSource](#).

## Buat proyek CodeBuild Lambda Python

Buat proyek CodeBuild Lambda Python.

## Untuk membuat proyek CodeBuild Lambda Java Anda

1. Buka AWS CodeBuild konsol di <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Jika halaman CodeBuild informasi ditampilkan, pilih Buat proyek build. Jika tidak, pada panel navigasi, perluas Build, pilih Build projects, lalu pilih Create build project.
3. Di Nama proyek, masukkan nama untuk proyek pembangunan ini. Membangun nama proyek harus unik di setiap AWS akun. Anda juga dapat menyertakan deskripsi opsional proyek build untuk membantu pengguna lain memahami tujuan proyek ini.
4. Di Sumber, pilih repositori sumber tempat AWS SAM proyek Anda berada.
5. Di Lingkungan:
  - Untuk Compute, pilih Lambda.
  - Untuk Runtime, pilih Python.
  - Untuk Gambar, pilih `aws/codebuild/amazonlinux-x86_64-lambda-standard:python3.12`.
  - Untuk peran Layanan, biarkan peran Layanan baru dipilih. Catat nama Peran. Ini akan diperlukan saat Anda memperbarui izin IAM proyek nanti dalam sampel ini.
6. Pilih Buat proyek build.
7. Buka konsol IAM di <https://console.aws.amazon.com/iam/>.
8. Di panel navigasi, pilih Peran dan pilih peran layanan yang terkait dengan proyek Anda. Anda dapat menemukan peran proyek Anda CodeBuild dengan memilih proyek build, memilih Edit, Lingkungan, dan kemudian peran Layanan.
9. Pilih tab Trust relationship, lalu pilih Edit trust policy.
10. Tambahkan kebijakan inline berikut ke peran IAM Anda. Ini akan digunakan untuk menyebarkan AWS SAM infrastruktur Anda nanti. Untuk informasi lebih lanjut, lihat [Menambahkan dan menghapus izin identitas IAM](#) dalam Panduan Pengguna IAM.

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Sid": "UpdateLambdaPermissions",
 "Effect": "Allow",
 "Action": [
 "lambda:UpdateFunctionConfiguration"
],
 "Resource": [
 "*"
]
 }
]
}
```

```
]
 }
]
}
```

## Siapkan proyek buildspec

Untuk memperbarui fungsi Lambda, skrip membaca variabel lingkungan dari buildspec untuk menemukan nama fungsi Lambda, nama variabel lingkungan, dan nilai variabel lingkungan.

Untuk menyiapkan buildspec proyek Anda

1. Di CodeBuild konsol, pilih proyek build Anda, lalu pilih Edit dan Buildspec.
2. Di Buildspec, pilih Sisipkan perintah build dan kemudian Beralih ke editor.
3. Hapus perintah build yang telah diisi sebelumnya dan tempel di buildspec berikut.

```
version: 0.2
env:
 variables:
 LAMBDA_FUNC_NAME: "<lambda-function-name>"
 LAMBDA_ENV_VARIABLE: "FEATURE_ENABLED"
 LAMBDA_ENV_VARIABLE_VALUE: "true"
phases:
 install:
 commands:
 - pip3 install boto3
 build:
 commands:
 - python3 update_lambda_environment_variables.py
```

4. Pilih Perbarui buildspec.

## Perbarui konfigurasi Lambda Anda

Gunakan CodeBuild Lambda Python untuk memperbarui konfigurasi fungsi Lambda Anda secara otomatis.

Untuk memperbarui konfigurasi fungsi Lambda

1. Pilih Mulai membangun.

2. Setelah build selesai, navigasikan ke fungsi Lambda Anda.
3. Pilih Konfigurasi dan kemudian variabel Lingkungan. Anda akan melihat variabel lingkungan baru dengan kunci `FEATURE_ENABLED` dan nilai `true`.

## Bersihkan infrastruktur Anda

Untuk menghindari biaya lebih lanjut untuk sumber daya yang Anda gunakan selama tutorial ini, hapus sumber daya yang dibuat untuk CodeBuild proyek Anda.

### Untuk Membersihkan Infrastruktur Anda

1. Arahkan ke CloudWatch konsol dan hapus grup CloudWatch log yang terkait dengan CodeBuild proyek Anda.
2. Arahkan ke CodeBuild konsol dan hapus CodeBuild project Anda dengan memilih Delete build project.
3. Jika Anda membuat fungsi Lambda untuk tujuan sampel ini, pilih fungsi Tindakan dan Hapus untuk membersihkan fungsi Lambda Anda.

## Ekstensi

Jika Anda ingin memperluas sampel ini untuk mengelola AWS sumber daya lain menggunakan AWS CodeBuild Lambda Python:

- Perbarui skrip Python untuk memodifikasi sumber daya baru menggunakan Boto3.
- Perbarui peran IAM yang terkait dengan CodeBuild proyek Anda agar memiliki izin untuk sumber daya baru.
- Tambahkan variabel lingkungan baru yang terkait dengan sumber daya baru ke buildspec Anda.

# Bekerja dengan kapasitas cadangan di AWS CodeBuild

CodeBuild menawarkan armada komputasi berikut:

- Armada sesuai permintaan
- Armada kapasitas cadangan

Dengan armada sesuai permintaan, CodeBuild menyediakan komputasi untuk build Anda. Mesin-mesin dihancurkan saat build selesai. Armada sesuai permintaan dikelola sepenuhnya, dan mencakup kemampuan penskalaan otomatis untuk menangani lonjakan permintaan.

## Note

Fleets sesuai permintaan tidak mendukung Windows Server 2022.

CodeBuild juga menawarkan armada kapasitas cadangan yang berisi instans yang didukung oleh Amazon EC2 yang dikelola oleh CodeBuild. Dengan armada kapasitas cadangan, Anda mengonfigurasi satu set instans khusus untuk lingkungan build Anda. Mesin ini tetap menganggur, siap untuk memproses build atau pengujian segera dan mengurangi durasi build. Dengan armada kapasitas cadangan, mesin Anda selalu berjalan dan akan terus mengeluarkan biaya selama disediakan.

## Important

Terlepas dari berapa lama Anda menjalankan instance, armada kapasitas cadangan dikenakan biaya awal per instance, dan kemudian mungkin ada biaya terkait tambahan. Untuk informasi selengkapnya, lihat <https://aws.amazon.com/codebuild/pricing/>.

## Topik

- [Bagaimana cara memulai dengan armada kapasitas cadangan?](#)
- [Praktik terbaik](#)
- [Dapatkah saya berbagi armada kapasitas cadangan di beberapa CodeBuild proyek?](#)
- [Wilayah mana yang mendukung armada kapasitas cadangan?](#)



- [Properti armada kapasitas cadangan](#)
- [Sampel kapasitas cadangan dengan AWS CodeBuild](#)
- [Keterbatasan armada kapasitas cadangan](#)

## Bagaimana cara memulai dengan armada kapasitas cadangan?

Untuk membuat armada kapasitas cadangan

1. Masuk ke AWS Management Console dan buka AWS CodeBuild konsol di <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Di panel navigasi, pilih Compute fleet, lalu pilih Create compute fleet.
3. Di bidang teks Compute fleet name, masukkan nama untuk armada Anda.
4. Dari menu drop-down sistem operasi, pilih sistem operasi.
5. Dari menu drop-down Architecture, pilih arsitektur.
6. Dari menu tarik-turun Compute, pilih jenis mesin komputasi untuk mesin Anda.
7. Di bidang teks Kapasitas, masukkan jumlah minimum instance dalam armada.
8. Di bidang perilaku Overflow, pilih perilaku saat permintaan melebihi kapasitas armada. Untuk informasi selengkapnya tentang opsi ini, lihat [Properti armada kapasitas cadangan](#).
9. Pilih Buat armada komputasi.
10. Setelah armada komputasi dibuat, buat CodeBuild proyek baru atau edit yang sudah ada. Dari Lingkungan, pilih Kapasitas cadangan di bawah Model penyediaan, lalu pilih armada yang ditentukan dengan nama Armada.

## Praktik terbaik

Saat menggunakan armada kapasitas cadangan, kami sarankan Anda mengikuti praktik terbaik ini.

- Sebaiknya gunakan mode cache sumber untuk membantu meningkatkan kinerja build dengan menyimpan sumber.
- Kami merekomendasikan penggunaan cache lapisan Docker untuk membantu meningkatkan kinerja build dengan menyimpan lapisan Docker yang ada.

# Dapatkan saya berbagi armada kapasitas cadangan di beberapa CodeBuild proyek?

Ya, Anda dapat memaksimalkan pemanfaatan kapasitas armada dengan menggunakannya di beberapa proyek.

## Wilayah mana yang mendukung armada kapasitas cadangan?

Armada kapasitas cadangan didukung sebagai berikut Wilayah AWS: US East (Virginia N.), US East (Ohio), US West (Oregon), Asia Pasifik (Mumbai), Asia Pasifik (Singapura), Asia Pasifik (Sydney), Asia Pasifik (Tokyo), Eropa (Frankfurt), Eropa (Irlandia), dan Amerika Selatan (São Paulo). Untuk informasi selengkapnya tentang Wilayah AWS tempat CodeBuild tersedia, lihat [AWS Layanan menurut Wilayah](#).

## Properti armada kapasitas cadangan

Armada kapasitas cadangan berisi properti berikut:

### Sistem operasi

Sistem operasi. Sistem operasi berikut tersedia:

- Amazon Linux
- Windows Server 2019
- Windows Server 2022

### Arsitektur

Arsitektur prosesor. Arsitektur berikut tersedia:

- x86\_64
- Arm64

### Hitung

Jenis mesin komputasi untuk setiap instance. Jenis mesin berikut tersedia:

| Jenis komputasi           | Nilai ComputeType Lingkungan | Nilai tipe lingkungan | Memori | vCPU | Ruang disk   |
|---------------------------|------------------------------|-----------------------|--------|------|--------------|
| ARM Kecil                 | BUILD_GENERAL1_SMALL         | ARM_CONTAINER         | 4 GB   | 2    | 50 GB        |
| ARM Besar                 | BUILD_GENERAL1_LARGE         | ARM_CONTAINER         | 16 GB  | 8    | 50 GB        |
| Linux Kecil <sup>1</sup>  | BUILD_GENERAL1_SMALL         | LINUX_CONTAINER       | 3 GB   | 2    | 64 GB        |
| Linux Sedang <sup>1</sup> | BUILD_GENERAL1_MEDIUM        | LINUX_CONTAINER       | 7 GB   | 4    | 128 GB       |
| Linux Besar <sup>1</sup>  | BUILD_GENERAL1_LARGE         | LINUX_CONTAINER       | 15 GB  | 8    | 128 GB       |
| Linux xLarge              | BUILD_GENERAL1_XLARGE        | LINUX_CONTAINER       | 70 GB  | 36   | 256 GB       |
| Linux 2xBesar             | BUILD_GENERAL1_2XLARGE       | LINUX_CONTAINER       | 145 GB | 72   | 824 GB (SSD) |
| Linux GPU Kecil           | BUILD_GENERAL1_SMALL         | LINUX_GPU_CONTAINER   | 16 GB  | 4    | 220 GB       |

| Jenis komputasi | Nilai ComputeType Lingkungan | Nilai tipe lingkungan         | Memori | vCPU | Ruang disk |
|-----------------|------------------------------|-------------------------------|--------|------|------------|
| Linux GPU Besar | BUILD_GENERAL1_LARGE         | LINUX_GPU_CONTAINER           | 255 GB | 32   | 50 GB      |
| Windows Medium  | BUILD_GENERAL1_MEDIUM        | WINDOWS_SERVER_2019_CONTAINER | 7 GB   | 4    | 128 GB     |
| Windows Medium  | BUILD_GENERAL1_MEDIUM        | WINDOWS_SERVER_2022_CONTAINER | 7 GB   | 4    | 128 GB     |
| Jendela Besar   | BUILD_GENERAL1_LARGE         | WINDOWS_SERVER_2019_CONTAINER | 15 GB  | 8    | 128 GB     |
| Jendela Besar   | BUILD_GENERAL1_LARGE         | WINDOWS_SERVER_2022_CONTAINER | 15 GB  | 8    | 128 GB     |

## Kapasitas

Jumlah awal mesin yang dialokasikan untuk armada, yang mendefinisikan jumlah build yang dapat berjalan secara paralel.

## Perilaku meluap

Mendefinisikan perilaku ketika jumlah build melebihi kapasitas armada.

## Sesuai permintaan

Build overflow berjalan sesuai permintaan. CodeBuild

### Important

Jika Anda memilih untuk menyetel perilaku overflow ke sesuai permintaan, perhatikan bahwa build overflow akan ditagih secara terpisah, mirip dengan Amazon EC2 sesuai permintaan. Untuk informasi selengkapnya, lihat <https://aws.amazon.com/codebuild/pricing/>.

## Antrian

Build run ditempatkan dalam antrian hingga mesin tersedia. Ini membatasi biaya tambahan karena tidak ada mesin tambahan yang dialokasikan.

# Sampel kapasitas cadangan dengan AWS CodeBuild

Sampel ini dapat digunakan untuk bereksperimen dengan armada kapasitas cadangan di CodeBuild.

## Topik

- [Caching dengan sampel kapasitas cadangan](#)

## Caching dengan sampel kapasitas cadangan

Cache dapat menyimpan bagian yang dapat digunakan kembali dari lingkungan build Anda dan menggunakannya di beberapa build. Contoh ini menunjukkan cara mengaktifkan caching dalam proyek build Anda menggunakan kapasitas cadangan. Untuk informasi selengkapnya, lihat [Membangun caching di AWS CodeBuild](#).

Anda dapat memulai dengan menentukan satu atau beberapa mode cache dalam pengaturan proyek Anda:

```
Cache:
 Type: LOCAL
 Modes:
 - LOCAL_CUSTOM_CACHE
```

- LOCAL\_DOCKER\_LAYER\_CACHE
- LOCAL\_SOURCE\_CACHE

### Note

Pastikan untuk mengaktifkan mode istimewa untuk menggunakan cache lapisan Docker.

Pengaturan buildspec proyek Anda akan terlihat seperti berikut:

```
version: 0.2
 phases:
 build:
 commands:
 - echo testing local source cache
 - touch /codebuild/cache/workspace/foobar.txt
 - git checkout -b cached_branch
 - echo testing local docker layer cache
 - docker run alpine:3.14 2>&1 | grep 'Pulling from' || exit 1
 - echo testing local custom cache
 - touch foo
 - mkdir bar && ln -s foo bar/foo2
 - mkdir bar/bar && touch bar/bar/foo3 && touch bar/bar/foo4
 - "[-f foo] || exit 1"
 - "[-L bar/foo2] || exit 1"
 - "[-f bar/bar/foo3] || exit 1"
 - "[-f bar/bar/foo4] || exit 1"
 cache:
 paths:
 - './foo'
 - './bar/**/*'
 - './bar/bar/foo3'
```

Anda dapat memulai dengan menjalankan build dengan proyek baru untuk menyemai cache. Setelah selesai, Anda harus memulai build lain dengan buildspec utama, mirip dengan yang berikut ini:

```
version: 0.2
 phases:
 build:
 commands:
 - echo testing local source cache
 - git branch | if grep 'cached_branch'; then (exit 0); else (exit 1); fi
```

```
- ls /codebuild/cache/workspace | if grep 'foobar.txt'; then (exit 0); else
(exit 1); fi
- echo testing local docker layer cache
- docker run alpine:3.14 2>&1 | if grep 'Pulling from'; then (exit 1); else
(exit 0); fi
- echo testing local custom cache
- "[-f foo] || exit 1"
- "[-L bar/foo2] || exit 1"
- "[-f bar/bar/foo3] || exit 1"
- "[-f bar/bar/foo4] || exit 1"
cache:
 paths:
 - './foo'
 - './bar/**/*'
 - './bar/bar/foo3'
```

## Keterbatasan armada kapasitas cadangan

Ada beberapa kasus penggunaan yang tidak didukung oleh armada kapasitas cadangan, dan jika berdampak pada Anda, gunakan armada sesuai permintaan sebagai gantinya:

- Armada kapasitas cadangan tidak mendukung pembuatan batch, metrik pemanfaatan build, atau pembuatan versi semantik.
- Armada kapasitas cadangan tidak mendukung konektivitas VPC.

Untuk informasi lebih lanjut tentang batas dan kuota, lihat [Hitung armada](#).

# Bekerja dengan pelaporan pengujian di AWS CodeBuild

Anda dapat membuat laporan CodeBuild yang berisi detail tentang pengujian yang dijalankan selama build. Anda dapat membuat pengujian seperti pengujian unit, pengujian konfigurasi, dan pengujian fungsional.

Format file laporan pengujian berikut didukung:

- Mentimun JSON (.json)
- JUnit XML (.xml)
- NUnit XML (.xml)
- NUnit3 XML (.xml.xml)
- TestNG XML (.xml)
- Studio Visual TRX (.trx)

## Note

Versi terbaru yang didukung `cucumber-js` adalah 7.3.2.

Buat kasus pengujian Anda dengan kerangka pengujian apa pun yang dapat membuat file laporan dalam salah satu format ini (misalnya, plugin Surefire JUnit, TestNG, atau Mentimun).

Untuk membuat laporan pengujian, Anda menambahkan nama grup laporan ke file buildspec proyek build dengan informasi tentang kasus pengujian Anda. Saat Anda menjalankan proyek build, kasus pengujian dijalankan dan laporan pengujian dibuat. Anda tidak perlu membuat grup laporan sebelum menjalankan pengujian. Jika Anda menentukan nama grup laporan, CodeBuild buat grup laporan untuk Anda saat menjalankan laporan. Jika Anda ingin menggunakan grup laporan yang sudah ada, Anda menentukan ARN dalam file buildspec.

Anda dapat menggunakan laporan pengujian untuk membantu memecahkan masalah selama proses build. Jika Anda memiliki banyak laporan pengujian dari beberapa build proyek build, Anda dapat menggunakan laporan pengujian untuk melihat tren dan tingkat pengujian dan kegagalan untuk membantu mengoptimalkan build.

Laporan kedaluwarsa 30 hari setelah dibuat. Anda tidak dapat melihat laporan pengujian yang kedaluwarsa. Jika ingin menyimpan laporan pengujian selama lebih dari 30 hari, Anda dapat



mengekspor file data mentah hasil pengujian ke bucket Amazon S3. File uji yang diekspor tidak kedaluwarsa. Informasi tentang bucket S3 ditentukan saat Anda membuat grup laporan.

#### Note

Peran CodeBuild layanan yang ditentukan dalam proyek digunakan untuk izin untuk mengunggah ke bucket S3.

## Topik

- [Buat laporan pengujian](#)
- [Bekerja dengan kelompok laporan](#)
- [Bekerja dengan laporan](#)
- [Bekerja dengan izin laporan pengujian](#)
- [Lihat laporan pengujian](#)
- [Pelaporan pengujian dengan kerangka kerja pengujian](#)
- [Laporan cakupan kode](#)

## Buat laporan pengujian

Untuk membuat laporan pengujian, Anda menjalankan proyek build yang dikonfigurasi dengan satu hingga lima grup laporan dalam file buildspec. Laporan pengujian dibuat selama proses dijalankan. Ini berisi hasil kasus uji yang ditentukan untuk grup laporan. Laporan pengujian baru dibuat untuk setiap build berikutnya yang menggunakan file buildspec yang sama.

Untuk membuat laporan pengujian

1. Buat proyek build. Untuk informasi, lihat [Membuat proyek pembangunan diAWS CodeBuild](#).
2. Konfigurasi file buildspec proyek Anda dengan informasi laporan pengujian:
  - a. Tambahkan `reports`: bagian dan tentukan ARN dari grup laporan yang ada, atau nama grup laporan.

Jika Anda menentukan ARN, CodeBuild gunakan grup laporan tersebut.

Jika Anda menentukan nama, CodeBuild membuat grup laporan untuk Anda menggunakan nama proyek Anda, dan nama yang Anda tentukan, dalam format `<project-name>-<`

- report-group-name* >. Jika grup laporan bernama sudah ada, CodeBuild gunakan grup laporan tersebut.
- Di bawah grup laporan, tentukan lokasi file yang berisi hasil pengujian. Jika Anda menggunakan lebih dari satu grup laporan, tentukan lokasi file hasil pengujian untuk masing-masing grup laporan. Laporan pengujian baru dibuat setiap kali proyek build Anda berjalan. Untuk informasi selengkapnya, lihat [Tentukan file uji](#).
  - Di commands bagian `post_build` urutan `build` atau, tentukan perintah yang menjalankan kasus pengujian yang Anda tentukan untuk grup laporan Anda. Untuk informasi selengkapnya, lihat [Tentukan perintah uji](#).

Berikut ini adalah contoh bagian `buildspec:reports`:

```
reports:
 php-reports:
 files:
 - "reports/php/*.xml"
 file-format: "JUNITXML"
 nunit-reports:
 files:
 - "reports/nunit/*.xml"
 file-format: "NUNITXML"
```

- Jalankan build proyek build. Untuk informasi selengkapnya, lihat [Jalankan build diAWS CodeBuild](#).
- Saat build selesai, pilih build baru yang dijalankan dari riwayat Build di halaman project Anda. Pilih Laporan untuk melihat laporan pengujian. Untuk informasi selengkapnya, lihat [Melihat laporan pengujian untuk build](#).

## Bekerja dengan kelompok laporan

Grup laporan berisi laporan pengujian dan menentukan pengaturan bersama. Anda menggunakan file `buildspec` untuk menentukan kasus uji yang akan dijalankan dan perintah untuk menjalankannya saat dibangun. Untuk setiap grup laporan yang dikonfigurasi dalam proyek build, proses proyek build akan membuat laporan pengujian. Beberapa proses proyek build yang dikonfigurasi dengan grup laporan membuat beberapa laporan pengujian dalam grup laporan tersebut, masing-masing dengan hasil kasus pengujian yang sama yang ditentukan untuk grup laporan tersebut.

Kasus uji ditentukan untuk grup laporan dalam file buildspec proyek build. Anda dapat menentukan hingga lima grup laporan dalam satu proyek build. Saat Anda menjalankan build, semua kasus uji berjalan. Laporan pengujian baru dibuat dengan hasil dari setiap kasus uji yang ditentukan untuk grup laporan. Setiap kali Anda menjalankan build baru, kasus pengujian berjalan dan laporan pengujian baru dibuat dengan hasil pengujian baru.

Grup laporan dapat digunakan di lebih dari satu proyek build. Semua laporan pengujian yang dibuat dengan satu grup laporan memiliki konfigurasi yang sama, seperti opsi ekspor dan izinnya, meskipun laporan pengujian dibuat menggunakan proyek build yang berbeda. Laporan pengujian yang dibuat dengan satu grup laporan di beberapa proyek build dapat berisi hasil dari menjalankan kumpulan kasus pengujian yang berbeda (satu set kasus uji untuk setiap proyek build). Ini karena Anda dapat menentukan file kasus uji yang berbeda untuk grup laporan di setiap file buildspec proyek. Anda juga dapat mengubah file kasus uji untuk grup laporan dalam proyek build dengan mengedit file buildspec. Proses build berikutnya membuat laporan pengujian baru yang berisi hasil file kasus uji di buildspec yang diperbarui.

## Topik

- [Membuat grup laporan](#)
- [Memperbarui grup laporan](#)
- [Tentukan file uji](#)
- [Tentukan perintah uji](#)
- [Laporkan penamaan grup](#)
- [Menandai kelompok laporan diAWS CodeBuild](#)
- [Bekerja dengan grup laporan bersama](#)

## Membuat grup laporan

Anda dapat menggunakan CodeBuild konsol, file AWS CLI, atau buildspec untuk membuat grup laporan. Peran IAM Anda harus memiliki izin yang diperlukan untuk membuat grup laporan. Untuk informasi selengkapnya, lihat [Bekerja dengan izin laporan pengujian](#).

## Topik

- [Membuat grup laporan \(buildspec\)](#)
- [Buat grup laporan \(konsol\)](#)
- [Membuat grup laporan \(CLI\)](#)

- [Membuat grup laporan \(AWS CloudFormation\)](#)

## Membuat grup laporan (buildspec)

Grup laporan yang dibuat menggunakan buildspec tidak mengeksport file hasil pengujian mentah. Anda dapat melihat grup laporan dan menentukan setelan ekspor. Untuk informasi selengkapnya, lihat [Memperbarui grup laporan](#).

Untuk membuat grup laporan menggunakan file buildspec

1. Pilih nama grup laporan yang tidak terkait dengan grup laporan di AWS akun Anda.
2. Konfigurasi `reports` bagian file buildspec dengan nama ini. Dalam contoh ini, nama grup laporan adalah `new-report-group` dan kasus uji penggunaan dibuat dengan kerangka JUnit:

```
reports:
 new-report-group: #surefire junit reports
 files:
 - '**/*'
 base-directory: 'surefire/target/surefire-reports'
```

Nama grup laporan juga dapat ditentukan dengan menggunakan variabel lingkungan dalam buildspec:

```
version: 0.2
env:
 variables:
 REPORT_GROUP_NAME: "new-report-group"
phases:
 build:
 commands:
 - ...
...
reports:
 $REPORT_GROUP_NAME:
 files:
 - '**/*'
 base-directory: 'surefire/target/surefire-reports'
```

Lihat informasi yang lebih lengkap di [Tentukan file uji](#) dan [Reports syntax in the buildspec file](#).

3. Di `commands` bagian ini, tentukan perintah untuk menjalankan pengujian Anda. Untuk informasi selengkapnya, lihat [Tentukan perintah uji](#).
4. Jalankan build. Ketika build selesai, grup laporan baru akan dibuat dengan nama yang menggunakan format `project-name-report-group-name`. Untuk informasi selengkapnya, lihat [Laporkan penamaan grup](#).

## Buat grup laporan (konsol)

### Membuat laporan pengujian

1. Buka AWS CodeBuild konsol di <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Di panel navigasi, pilih Grup Laporan.
3. Pilih Buat grup laporan.
4. Untuk Nama grup laporan, masukkan nama untuk grup laporan Anda.
5. (Opsional) Untuk Tag, masukkan nama dan nilai tanda yang ingin Anda dukung AWS layanan untuk digunakan. Gunakan Tambahkan baris untuk menambahkan tanda. Anda dapat menambahkan hingga 50 tanda.
6. Jika Anda ingin mengunggah data mentah hasil laporan pengujian Anda ke bucket Amazon S3:
  - a. Pilih Ekspor ke Amazon S3.
  - b. Untuk Nama bucket S3, masukkan nama bucket S3.
  - c. (Opsional) Untuk Pemilik bucket S3, masukkan AWS pengidentifikasi akun yang memiliki bucket S3. Ini memungkinkan data laporan untuk diekspor ke bucket Amazon S3 yang dimiliki oleh akun selain akun yang menjalankan build.
  - d. Untuk prefiks jalan, masukkan jalur di bucket S3 tempat Anda ingin mengunggah hasil tes.
  - e. Pilih Kompres data hasil uji dalam file zip untuk kompres file data hasil tes mentah Anda.
  - f. Perluas Konfigurasi tambahan untuk menampilkan opsi enkripsi. Pilih salah satu dari berikut:
    - Default AWS kunci yang dikelola untuk menggunakan Kunci yang dikelola AWS untuk Amazon S3. Untuk informasi selengkapnya, lihat [CMK yang dikelola pelanggan](#) di AWS Key Management Service Panduan Pengguna. Ini adalah pilihan enkripsi default.
    - Pilih kunci kustom untuk menggunakan kunci terkelola pelanggan yang Anda buat dan konfigurasi. Untuk AWS KMS kunci enkripsi, masukkan ARN kunci enkripsi Anda. Formatnya adalah `arn:aws:kms:<region-id>:<aws-account-id>:key/<key-`

`id` . Untuk informasi selengkapnya, lihat [Membuat kunci KMS](#) di AWS Key Management Service Panduan Pengguna.

- Enkripsi artefak menonaktifkan untuk menonaktifkan enkripsi. Anda dapat memilih ini jika Anda ingin membagikan hasil tes Anda, atau mempublikasikannya ke situs web statis. (Situs web dinamis dapat menjalankan kode untuk mendekripsi hasil tes.)

Untuk informasi selengkapnya tentang enkripsi data saat istirahat, lihat [Enkripsi data](#).

#### Note

Peran layanan CodeBuild yang ditentukan dalam proyek digunakan untuk izin untuk mengunggah ke bucket S3.

7. Pilih Buat grup laporan.

## Membuat grup laporan (CLI)

Untuk membuat grup laporan

1. Buat file bernama `CreateReportGroup.json`.
2. Tergantung pada kebutuhan Anda, salin salah satu cuplikan kode JSON berikut ke `CreateReportGroup.json`:
  - Gunakan JSON berikut untuk menentukan bahwa grup laporan pengujian Anda mengeksport file hasil pengujian mentah ke bucket Amazon S3.

```
{
 "name": "<report-name>",
 "type": "TEST",
 "exportConfig": {
 "exportConfigType": "S3",
 "s3Destination": {
 "bucket": "<bucket-name>",
 "bucketOwner": "<bucket-owner>",
 "path": "<path>",
 "packaging": "NONE | ZIP",
 "encryptionDisabled": "false",
 "encryptionKey": "<your-key>"
 }
 }
}
```

```
 },
 "tags": [
 {
 "key": "tag-key",
 "value": "tag-value"
 }
]
 }
}
```

- Ganti `<bucket-name>` dengan nama bucket Amazon S3 dan `<path>` dengan jalur di bucket Anda ke tempat Anda ingin mengekspor file.
- Jika Anda ingin kompres file yang diekspor, untuk `packaging`, tentukan `ZIP`. Jika tidak, tentukan `NONE`.
- `bucketOwner` bersifat opsional dan hanya diperlukan jika bucket Amazon S3 dimiliki oleh akun selain akun yang menjalankan build.
- Gunakan `encryptionDisabled` untuk menentukan apakah akan mengenkripsi file yang diekspor. Jika Anda mengenkripsi file yang diekspor, masukkan kunci terkelola pelanggan Anda. Untuk informasi selengkapnya, lihat [Memperbarui grup laporan](#).
- Gunakan JSON berikut untuk menentukan bahwa laporan pengujian Anda tidak mengekspor file uji mentah:

```
{
 "name": "<report-name>",
 "type": "TEST",
 "exportConfig": {
 "exportConfigType": "NO_EXPORT"
 }
}
```

#### Note

Peran layanan CodeBuild yang ditentukan dalam proyek digunakan untuk izin untuk mengunggah ke bucket S3.

### 3. Jalankan perintah berikut:

```
aws codebuild create-report-group --cli-input-json file://
CreateReportGroupInput.json
```

## Membuat grup laporan (AWS CloudFormation)

Untuk membuat laporan pengujian menggunakan AWS CloudFormation template

Anda dapat menggunakan file AWS CloudFormation templat untuk membuat dan menyediakan grup laporan. Untuk informasi selengkapnya, lihat [Panduan Pengguna AWS CloudFormation](#).

Template AWS CloudFormation YAMB berikut membuat grup laporan yang tidak mengekspor file hasil pengujian mentah.

```
Resources:
 CodeBuildReportGroup:
 Type: AWS::CodeBuild::ReportGroup
 Properties:
 Name: my-report-group-name
 Type: TEST
 ExportConfig:
 ExportConfigType: NO_EXPORT
```

Template AWS CloudFormation YAMAL berikut membuat grup laporan yang mengekspor file hasil pengujian mentah ke bucket Amazon S3.

```
Resources:
 CodeBuildReportGroup:
 Type: AWS::CodeBuild::ReportGroup
 Properties:
 Name: my-report-group-name
 Type: TEST
 ExportConfig:
 ExportConfigType: S3
 S3Destination:
 Bucket: my-s3-bucket-name
 Path: path-to-folder-for-exported-files
 Packaging: ZIP
 EncryptionKey: my-KMS-encryption-key
 EncryptionDisabled: false
```



**Note**

Peran CodeBuild layanan yang ditentukan dalam proyek digunakan untuk izin untuk mengunggah ke bucket S3.

## Memperbarui grup laporan

Saat memperbarui grup laporan, Anda dapat menentukan informasi tentang apakah akan mengekspor data hasil pengujian mentah ke file di bucket Amazon S3. Jika Anda memilih untuk mengekspor ke bucket S3, Anda dapat menentukan hal berikut untuk grup laporan Anda:

- Apakah file hasil tes mentah dikompresi dalam file ZIP.
- Apakah file hasil pengujian mentah dienkripsi. Anda dapat menentukan enkripsi dengan salah satu dari berikut ini:
  - An Kunci yang dikelola AWS untuk Amazon S3.
  - Kunci terkelola pelanggan yang Anda buat dan konfigurasi.

Untuk informasi selengkapnya, lihat [Enkripsi data](#).

Jika Anda menggunakan AWS CLI untuk memperbarui grup laporan, Anda juga dapat memperbarui atau menambahkan tag. Untuk informasi selengkapnya, lihat [Menandai kelompok laporan diAWS CodeBuild](#).

**Note**

Peran CodeBuild layanan yang ditentukan dalam proyek digunakan untuk izin untuk mengunggah ke bucket S3.

### Topik

- [Memperbarui grup laporan \(konsol\)](#)
- [Perbarui grup laporan \(CLI\)](#)

## Memperbarui grup laporan (konsol)

Untuk memperbarui grup laporan

1. Buka AWS CodeBuild konsol di <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Di panel navigasi, pilih Laporkan grup.
3. Pilih grup laporan yang ingin Anda perbarui.
4. Pilih Edit.
5. Pilih atau hapus Cadangan ke Amazon S3. Jika Anda memilih opsi ini, tentukan pengaturan ekspor Anda:
  - a. Untuk nama bucket S3, masukkan nama bucket S3.
  - b. Untuk awalan Path, masukkan path di bucket S3 tempat Anda ingin mengunggah hasil pengujian.
  - c. Pilih Kompres data hasil tes dalam file zip untuk mengompres file data hasil pengujian mentah Anda.
  - d. Perluas Konfigurasi tambahan untuk menampilkan opsi enkripsi. Pilih salah satu cara berikut:
    - Kunci AWS terkelola default untuk menggunakan Kunci yang dikelola AWS untuk Amazon S3. Untuk informasi selengkapnya, lihat [CMK yang dikelola pelanggan](#) di Panduan AWS Key Management Service Pengguna. Ini adalah opsi enkripsi default.
    - Pilih kunci kustom untuk menggunakan kunci terkelola pelanggan yang Anda buat dan konfigurasi. Untuk kunci AWS KMS enkripsi, masukkan ARN kunci enkripsi Anda. Formatnya adalah `arn:aws:kms:<region-id>:<aws-account-id>:key/<key-id>`. Untuk informasi selengkapnya, lihat [Membuat kunci KMS](#) di Panduan AWS Key Management Service Pengguna.
    - Nonaktifkan enkripsi artefak untuk menonaktifkan enkripsi. Anda dapat memilih ini jika Anda ingin membagikan hasil tes Anda, atau mempublikasikannya ke situs web statis. (Situs web dinamis dapat menjalankan kode untuk mendekripsi hasil pengujian.)

## Perbarui grup laporan (CLI)

Untuk memperbarui grup laporan

1. Buat file bernama `UpdateReportGroupInput.json`.

## 2. Salin yang berikut ini keUpdateReportGroupInput.json:

```
{
 "arn": "",
 "exportConfig": {
 "exportConfigType": "S3",
 "s3Destination": {
 "bucket": "bucket-name",
 "path": "path",
 "packaging": "NONE | ZIP",
 "encryptionDisabled": "false",
 "encryptionKey": "your-key"
 }
 },
 "tags": [
 {
 "key": "tag-key",
 "value": "tag-value"
 }
]
}
```

## 3. Masukkan ARN grup laporan Anda di arn baris (misalnya,.

"arn": "arn:aws:codebuild:region:123456789012:report-group/report-group-1")

## 4. Perbarui UpdateReportGroupInput.json dengan pembaruan yang ingin Anda terapkan ke grup laporan Anda.

- Jika Anda ingin memperbarui grup laporan untuk mengekspor file hasil pengujian mentah ke bucket S3, perbarui exportConfig bagian tersebut. Ganti bucket-name dengan nama bucket S3 Anda dan path dengan jalur di bucket S3 yang ingin Anda ekspor file. Jika Anda ingin mengompres file yang diekspor, untukpackaging, tentukanZIP. Jika tidak, tentukanNONE. Gunakan encryptionDisabled untuk menentukan apakah akan mengenkripsi file yang diekspor. Jika Anda mengenkripsi file yang diekspor, masukkan kunci yang dikelola pelanggan Anda.
- Jika ingin memperbarui grup laporan agar tidak mengekspor file hasil pengujian mentah ke bucket S3, perbarui exportConfig bagian tersebut dengan JSON berikut:

```
{
 "exportConfig": {
```

```
 "exportConfigType": "NO_EXPORT"
 }
}
```

- Jika Anda ingin memperbarui tag grup laporan, perbarui tags bagian tersebut. Anda dapat mengubah, menambah, atau menghapus tag. Jika Anda ingin menghapus semua tag, perbarui dengan JSON berikut:

```
"tags": []
```

## 5. Jalankan perintah berikut:

```
aws codebuild update-report-group \
--cli-input-json file://UpdateReportGroupInput.json
```

## Tentukan file uji

Anda menentukan file hasil pengujian dan lokasinya untuk setiap grup laporan di `reports` bagian file `buildspec` project build Anda. Untuk informasi selengkapnya, lihat [Reports syntax in the buildspec file](#).

Berikut ini adalah `reports` bagian contoh yang menentukan dua grup laporan untuk proyek build. Satu ditentukan dengan ARN-nya, yang lain dengan nama. `files` bagian ini menentukan file yang berisi hasil kasus uji. `base-directory` bagian opsional menentukan direktori tempat file kasus uji berada. `discard-paths` bagian opsional menentukan apakah jalur untuk menguji file hasil yang diunggah ke bucket Amazon S3 dibuang.

```
reports:
 arn:aws:codebuild:your-region:your-aws-account-id:report-group/report-group-name-1:
 #surefire junit reports
 files:
 - '**/*'
 base-directory: 'surefire/target/surefire-reports'
 discard-paths: false

sampleReportGroup: #Cucumber reports from json plugin
 files:
 - 'cucumber-json/target/cucumber-json-report.json'
 file-format: CUCUMBERJSON #Type of the report, defaults to JUNITXML
```

## Tentukan perintah uji

Anda menentukan perintah yang menjalankan kasus pengujian di `commands` bagian file `buildspec` Anda. Perintah ini menjalankan kasus pengujian yang ditentukan untuk grup laporan Anda di `reports` bagian file `buildspec` Anda. Berikut ini adalah `commands` bagian contoh yang mencakup perintah untuk menjalankan tes dalam file pengujian:

```
commands:
 - echo Running tests for surefire junit
 - mvn test -f surefire/pom.xml -fn
 - echo
 - echo Running tests for cucumber with json plugin
 - mvn test -Dcucumber.options="--plugin json:target/cucumber-json-report.json" -f
 cucumber-json/pom.xml -fn
```

Untuk informasi selengkapnya, lihat [Sintaks Buildspec](#).

## Laporkan penamaan grup

Saat Anda menggunakan AWS CLI atau AWS CodeBuild konsol untuk membuat grup laporan, Anda menentukan nama untuk grup laporan. Jika Anda menggunakan `buildspec` untuk membuat grup laporan baru, itu dinamai menggunakan format. *project-name-report-group-name-specified-in-buildspec* Semua laporan yang dibuat dengan menjalankan build proyek build tersebut milik grup laporan baru yang memiliki nama baru.

Jika Anda tidak CodeBuild ingin membuat grup laporan baru, tentukan ARN grup laporan dalam file `buildspec` proyek build. Anda dapat menentukan ARN grup laporan di beberapa proyek build. Setelah setiap proyek build berjalan, grup laporan berisi laporan pengujian yang dibuat oleh setiap proyek build.

Misalnya, jika Anda membuat satu grup laporan dengan `namamy-report-group`, lalu menggunakan namanya dalam dua proyek build berbeda bernama `my-project-1` dan `my-project-2` dan membuat build dari kedua proyek, dua grup laporan baru akan dibuat. Hasilnya adalah tiga kelompok laporan dengan nama-nama berikut:

- `my-report-group`: Tidak memiliki laporan pengujian.
- `my-project-1-my-report-group`: Berisi laporan dengan hasil pengujian yang dijalankan oleh proyek build bernama `my-project-1`.

- `my-project-2-my-report-group`: Berisi laporan dengan hasil pengujian yang dijalankan oleh proyek build bernama `my-project-2`.

Jika Anda menggunakan ARN dari grup laporan yang dinamai `my-report-group` di kedua proyek, dan kemudian menjalankan build dari setiap proyek, Anda masih memiliki satu grup laporan (`my-report-group`). Grup laporan tersebut berisi laporan pengujian dengan hasil pengujian yang dijalankan oleh kedua proyek build.

Jika Anda memilih nama grup laporan yang bukan milik grup laporan di AWS akun Anda, lalu menggunakan nama tersebut untuk grup laporan dalam file `buildspec` dan menjalankan build proyek build-nya, grup laporan baru akan dibuat. Format nama grup laporan baru adalah `project-name-new-group-name`. Misalnya, jika tidak ada grup laporan di AWS akun Anda dengan nama tersebut `new-report-group`, dan menentukannya dalam proyek build yang disebut `test-project`, build run akan membuat grup laporan baru dengan nama tersebut `test-project-new-report-group`.

## Menandai kelompok laporan di AWS CodeBuild

Tanda adalah label atribut khusus yang Anda atau AWS tetapkan ke sumber daya AWS. Setiap tanda AWS memiliki dua bagian:

- Sebuah kunci tag (misalnya, `CostCenter`, `Environment`, `Project`, atau `Secret`). Kunci tanda peka terhadap huruf besar dan kecil.
- Bidang opsional yang dikenal sebagai nilai tag (misalnya, `111122223333`, `Production`, atau nama tim). Mengabaikan nilai tag sama dengan menggunakan rangkaian kosong. Seperti kunci tanda, nilai tanda peka huruf besar dan kecil.

Bersama-sama ini dikenal sebagai pasangan nilai-kunci. Untuk batas jumlah tag yang dapat Anda miliki di grup laporan dan batasan kunci dan nilai tanda, lihat [Tanda](#).

Tag membantu Anda mengidentifikasi dan mengatur sumber daya AWS. Banyak layanan AWS yang mendukung penandaan, sehingga Anda dapat menetapkan tag yang sama ke sumber daya dari layanan yang berbeda untuk menunjukkan bahwa sumber daya tersebut terkait. Misalnya, Anda dapat menetapkan tag yang sama ke grup laporan CodeBuild yang Anda tetapkan ke bucket Amazon S3. Untuk informasi lebih lanjut tentang penggunaan tanda, lihat laporan resmi [Praktik terbaik](#).

Dalam CodeBuild, sumber daya utama adalah grup laporan dan proyek. Anda dapat menggunakan konsol CodeBuild, yang AWS CLI, CodeBuild API, atau AWS SDK untuk menambahkan, mengelola,

dan menghapus tag untuk grup laporan. Selain mengidentifikasi, mengatur, dan melacak grup laporan Anda dengan tanda, Anda dapat menggunakan tanda di kebijakan IAM untuk membantu mengontrol siapa yang dapat melihat dan berinteraksi dengan grup laporan Anda. Untuk contoh kebijakan akses berbasis tag, lihat [Menggunakan tag untuk mengontrol akses ke AWS CodeBuild sumber daya](#).

## Topik

- [Menambahkan tag ke grup laporan](#)
- [Melihat tanda untuk grup laporan](#)
- [Edit tanda untuk grup laporan](#)
- [Menghapus tag dari grup laporan](#)

## Menambahkan tag ke grup laporan

Menambahkan tanda ke grup laporan dapat membantu Anda mengidentifikasi dan mengatur tagnya AWS sumber daya dan mengelola akses ke mereka. Pertama, Anda menambahkan satu tag atau lebih (pasangan nilai-kunci) ke grup laporan. Perlu diingat bahwa ada batasan jumlah tag yang dapat Anda miliki di grup laporan. Ada pembatasan pada karakter yang dapat Anda gunakan di bidang kunci dan nilai. Untuk informasi selengkapnya, lihat [Tanda](#). Setelah Anda memiliki tag, Anda dapat membuat kebijakan IAM untuk mengelola akses ke grup laporan berdasarkan tag ini. Anda dapat menggunakan konsol CodeBuild atau konsol AWS CLI untuk menambahkan tag ke grup laporan.

### Important

Menambahkan tag ke grup laporan dapat memengaruhi akses ke grup laporan tersebut. Sebelum menambahkan tag ke grup laporan, pastikan untuk meninjau kebijakan IAM apa pun yang mungkin menggunakan tag untuk mengontrol akses ke sumber daya seperti grup laporan. Untuk contoh kebijakan akses berbasis tag, lihat [Menggunakan tag untuk mengontrol akses ke AWS CodeBuild sumber daya](#).

Untuk informasi selengkapnya tentang menambahkan tag ke grup laporan saat Anda membuatnya, lihat [Buat grup laporan \(konsol\)](#).

## Topik

- [Menambahkan tag ke grup laporan \(konsol\)](#)

- [Menambahkan tag ke grup laporan \(AWS CLI\)](#)

## Menambahkan tag ke grup laporan (konsol)

Anda dapat menggunakan konsol CodeBuild untuk menambahkan satu tag atau lebih ke grup laporan CodeBuild.

1. Buka konsol CodeBuild di <https://console.aws.amazon.com/codebuild/>.
2. Masuk Grup Laporan, pilih nama grup laporan tempat Anda ingin menambahkan tanda.
3. Di panel navigasi, pilih Pengaturan.
4. Jika tidak ada tanda yang telah ditambahkan ke grup laporan, pilih Tambahkan tag. Anda juga dapat memilih Edit, dan kemudian pilih Tambahkan tag.
5. Di Kunci, masukkan sebuah nama untuk tag tersebut. Anda dapat menambahkan nilai opsional untuk tag di Nilai.
6. (Opsional) Untuk menambahkan tag lain, pilih Tambahkan tag lagi.
7. Setelah Anda selesai menambahkan tag, pilih Kirim.

## Menambahkan tag ke grup laporan (AWS CLI)

Untuk menambahkan tag ke grup laporan saat Anda membuatnya, lihat [Membuat grup laporan \(CLI\)](#). Masuk `CreateReportGroup.json`, tambahkan tag Anda.

Untuk menambahkan tag ke grup laporan yang ada, lihat [Perbarui grup laporan \(CLI\)](#) dan tambahkan tag Anda di `UpdateReportGroupInput.json`.

Dalam langkah-langkah ini, kami menganggap bahwa Anda telah menginstal versi terbaru dari AWS CLI atau diperbarui ke versi terkini. Untuk informasi lebih lanjut, lihat [Menginstal AWS Command Line Interface](#).

## Melihat tanda untuk grup laporan

Tag dapat membantu Anda mengidentifikasi dan mengatur sumber daya AWS Anda serta mengelola akses ke mereka. Untuk informasi lebih lanjut tentang penggunaan tanda, lihat laporan resmi [Praktik terbaik](#). Untuk contoh kebijakan akses berbasis tag, lihat [Deny or allow actions on report groups based on resource tags](#).



## Melihat tanda untuk grup laporan (konsol)

Anda dapat menggunakan konsol CodeBuild untuk melihat tanda yang terkait dengan grup laporan CodeBuild.

1. Buka konsol CodeBuild di <https://console.aws.amazon.com/codebuild/>.
2. Masuk Grup Laporan, pilih nama grup laporan tempat Anda ingin melihat tanda.
3. Di panel navigasi, pilih Pengaturan.

## Melihat tanda untuk grup laporan (AWS CLI)

Ikuti langkah-langkah berikut untuk menggunakan AWS CLI untuk melihat AWStag untuk grup laporan. Jika tidak ada tanda yang telah ditambahkan, daftar tanda yang dikembalikan kosong.

1. Gunakan konsol atau AWS CLI untuk menemukan ARN grup laporan Anda. Catat itu.

### AWS CLI

Jalankan perintah berikut.

```
aws list-report-groups
```

Perintah ini mengembalikan informasi JSON yang serupa dengan yang berikut ini:

```
{
 "reportGroups": [
 "arn:aws:codebuild:region:123456789012:report-group/report-group-1",
 "arn:aws:codebuild:region:123456789012:report-group/report-group-2",
 "arn:aws:codebuild:region:123456789012:report-group/report-group-3"
]
}
```

Grup laporan ARN berakhir dengan namanya, yang dapat Anda gunakan untuk mengidentifikasi ARN untuk grup laporan Anda.

### Console

1. Buka konsol CodeBuild di <https://console.aws.amazon.com/codebuild/>.
2. Masuk Grup Laporan, pilih nama grup laporan Anda dengan tanda yang ingin Anda lihat.
3. Masuk Konfigurasi cari ARN grup laporan Anda.

2. Jalankan perintah berikut. Gunakan ARN Anda membuat catatan untuk `--report-group-arns` parameter.

```
aws codebuild batch-get-report-groups --report-group-arns
arn:aws:codebuild:region:123456789012:report-group/report-group-name
```

Jika berhasil, perintah ini mengembalikan informasi JSON yang berisi sebuah `tags` bagian yang serupa dengan yang berikut ini:

```
{
 ...
 "tags": {
 "Status": "Secret",
 "Project": "TestBuild"
 }
 ...
}
```

## Edit tanda untuk grup laporan

Anda dapat mengubah nilai untuk tag yang terkait dengan grup laporan. Anda juga dapat mengubah nama kunci, yang setara dengan menghapus tag saat ini dan menambahkan tag yang berbeda dengan nama baru dan nilai yang sama dengan kunci lainnya. Perlu diingat bahwa ada pembatasan pada karakter yang dapat Anda gunakan di bidang kunci dan nilai. Untuk informasi selengkapnya, lihat [Tanda](#).

### Important

Mengedit tag untuk grup laporan dapat memengaruhi akses ke grup laporan tersebut. Sebelum Anda mengedit nama (kunci) atau nilai tag untuk grup laporan, pastikan untuk meninjau kebijakan IAM apa pun yang mungkin menggunakan kunci atau nilai tag untuk mengontrol akses ke sumber daya seperti grup laporan. Untuk contoh kebijakan akses berbasis tag, lihat [Deny or allow actions on report groups based on resource tags](#).

## Mengedit tag untuk grup laporan (konsol)

Anda dapat menggunakan konsol CodeBuild untuk mengedit tag yang terkait dengan grup laporan CodeBuild.

1. Buka konsol CodeBuild di <https://console.aws.amazon.com/codebuild/>.
2. Masuk Grup Laporan, pilih nama grup laporan tempat Anda ingin mengedit tanda.
3. Di panel navigasi, pilih Pengaturan.
4. Pilih Edit.
5. Lakukan salah satu dari berikut:
  - Untuk mengubah tag, masukkan nama baru di Kunci. Mengubah nama tag adalah setara dengan menghapus tag dan menambahkan tag baru dengan nama kunci baru.
  - Untuk mengubah nilai sebuah tag, masukkan nilai baru. Jika Anda ingin mengubah nilai menjadi nol, hapus nilai saat ini dan biarkan bidang kosong.
6. Setelah Anda selesai mengedit tag, pilih Kirim.

### Edit tanda untuk grup laporan (AWS CLI)

Untuk menambahkan, mengubah, atau menghapus tag dari grup laporan, lihat [Perbarui grup laporan \(CLI\)](#). Perbarui tag di `updateReportGroupInput.json`.

### Menghapus tag dari grup laporan

Anda dapat menghapus satu atau beberapa tag yang terkait dengan grup laporan. Menghapus tag tidak menghapus tag dari AWS sumber daya yang terkait dengan tag tersebut.

#### Important

Menghapus tag untuk grup laporan dapat memengaruhi akses ke grup laporan tersebut. Sebelum Anda menghapus tag dari grup laporan, pastikan untuk meninjau kebijakan IAM apa pun yang mungkin menggunakan kunci atau nilai untuk tag untuk mengontrol akses ke sumber daya seperti grup laporan. Untuk contoh kebijakan akses berbasis tag, lihat [Menggunakan tag untuk mengontrol akses ke AWS CodeBuild sumber daya](#).

### Menghapus tag dari grup laporan (konsol)

Anda dapat menggunakan konsol CodeBuild untuk menghapus hubungan antara tag dan grup laporan CodeBuild.

1. Buka konsol CodeBuild di <https://console.aws.amazon.com/codebuild/>.

2. Masuk Grup Laporan, pilih nama grup laporan tempat Anda ingin menghapus tanda.
3. Di panel navigasi, pilih Pengaturan.
4. Pilih Edit.
5. Temukan tag yang ingin Anda hapus, lalu pilih Hapus tag.
6. Setelah Anda selesai menghapus tag, pilih Kirim.

### Menghapus tag dari grup laporan (AWS CLI)

Ikuti langkah-langkah berikut untuk menggunakan AWS CLI untuk menghapus tag dari grup laporan CodeBuild. Menghapus tag tidak menghapusnya, tetapi hanya menghapus hubungan antara tag dan grup laporan.

#### Note

Jika Anda menghapus grup laporan CodeBuild, semua asosiasi tag dihapus dari grup laporan yang dihapus. Anda tidak perlu menghapus tanda sebelum menghapus grup laporan.

Untuk menghapus satu tag atau beberapa dari grup laporan, lihat [Edit tanda untuk grup laporan \(AWS CLI\)](#). Perbarui bagian dalam data JSON diformat dengan daftar tag yang diperbarui yang tidak berisi yang ingin Anda hapus. Jika Anda ingin menghapus semua tag, perbarui bagian ke:

```
"tags: []"
```

## Bekerja dengan grup laporan bersama

Laporan grup berbagi memungkinkan beberapa AWS akun atau pengguna untuk melihat grup laporan, laporan yang belum kedaluwarsa, dan hasil pengujian laporannya. Dalam model ini, akun yang memiliki grup laporan (pemilik) berbagi grup dengan akun lain (konsumen). Konsumen tidak dapat mengedit grup laporan. Laporan kedaluwarsa 30 hari setelah dibuat.

### Isi

- [Prasyarat untuk berbagi grup laporan](#)
- [Prasyarat untuk mengakses grup laporan yang dibagikan dengan Anda](#)
- [Layanan terkait](#)

- [Berbagi grup laporan](#)
- [Membatalkan berbagi grup laporan bersama](#)
- [Mengidentifikasi grup laporan bersama](#)
- [Izin grup bersama](#)

## Prasyarat untuk berbagi grup laporan

Untuk berbagi grup laporan, AWS Akun harus memilikinya. Anda tidak dapat berbagi grup laporan yang telah dibagikan dengan Anda.

## Prasyarat untuk mengakses grup laporan yang dibagikan dengan Anda

Untuk mengakses grup laporan bersama, peran IAM konsumen memerlukan `BatchGetReportGroups` izin. Anda dapat melampirkan kebijakan berikut ke peran IAM mereka:

```
{
 "Effect": "Allow",
 "Resource": [
 "*"
],
 "Action": [
 "codebuild:BatchGetReportGroups"
]
}
```

Untuk informasi selengkapnya, lihat [Menggunakan kebijakan berbasis identitas untuk AWS CodeBuild](#).

## Layanan terkait

Laporan berbagi grup terintegrasi dengan AWS Resource Access Manager (AWS RAM), layanan yang memungkinkan Anda untuk berbagi AWS sumber daya dengan AWS Akun atau melalui AWS Organizations. Dengan AWS RAM, Anda dapat berbagi sumber daya yang Anda miliki dengan membuat berbagi sumber daya yang menentukan sumber daya dan konsumen untuk berbagi dengan mereka. Konsumen bisa bersifat individual AWS Akun, unit organisasi di AWS Organizations, atau seluruh organisasi di AWS Organizations.

Untuk informasi selengkapnya, lihat Panduan Pengguna [AWS RAM](#).

## Berbagi grup laporan

Saat Anda membagikan grup laporan, konsumen diberikan akses hanya-baca ke grup laporan dan laporannya. Konsumen dapat menggunakan AWS CLI untuk melihat grup laporan, laporannya, dan hasil uji kasus untuk setiap laporan. Konsumen tidak dapat:

- Melihat grup laporan bersama atau laporannya di konsol CodeBuild.
- Mengedit grup laporan bersama.
- Gunakan ARN grup laporan bersama dalam proyek untuk menjalankan laporan. Membangun proyek yang menentukan grup laporan bersama gagal.

Anda dapat menggunakan konsol CodeBuild untuk menambahkan grup laporan ke bagian sumber daya yang ada. Jika Anda ingin menambahkan grup laporan ke berbagi sumber daya baru, Anda harus terlebih dahulu membuatnya di [AWS RAM konsol](#).

Untuk berbagi grup dengan unit organisasi atau seluruh organisasi, Anda harus mengaktifkan berbagi dengan AWS Organizations. Untuk informasi lebih lanjut, lihat [Aktifkan pembagian dengan AWS Organizations](#) dalam Panduan Pengguna AWS RAM .

Anda dapat menggunakan konsol CodeBuild, AWS RAM konsol, atau AWS CLI untuk berbagi grup laporan yang Anda miliki.

Untuk berbagi grup laporan yang Anda miliki (CodeBuild console)

1. Buka AWS CodeBuild konsol di <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Di panel navigasi, pilih Grup laporan.
3. Pilih proyek yang ingin Anda bagikan, lalu pilih Bagikan. Untuk informasi selengkapnya, lihat [Membuat berbagi sumber daya](#) di AWS RAM Panduan Pengguna.

Untuk berbagi grup laporan yang Anda miliki (AWS RAM konsol)

Lihat [Membuat berbagi sumber daya](#) di AWS RAM Panduan Pengguna.

Untuk berbagi grup laporan yang Anda miliki (AWS RAM perintah)

Gunakan perintah [create-resource-share](#).

Untuk berbagi grup laporan yang Anda miliki (perintah CodeBuild)

Menggunakan [kebijakan-put-resource](#)-perintah:

1. Membuat file bernama `policy.json` dan salin yang berikut ke dalamnya.

```
{
 "Version": "2012-10-17",
 "Statement": [{
 "Effect": "Allow",
 "Principal": {
 "AWS": "consumer-aws-account-id-or-user"
 },
 "Action": [
 "codebuild:BatchGetReportGroups",
 "codebuild:BatchGetReports",
 "codebuild:ListReportsForReportGroup",
 "codebuild:DescribeTestCases"],
 "Resource": "arn-of-report-group-to-share"
]
}
```

2. Perbarui `policy.json` dengan kelompok laporan ARN dan pengidentifikasi untuk berbagi dengan. Contoh berikut memberikan akses hanya baca ke grup laporan dengan ARN `arn:aws:codebuild:us-west-2:123456789012:report-group/my-report-group` ke Alice dan pengguna root untuk AWS akun diidentifikasi oleh 123456789012.

```
{
 "Version": "2012-10-17",
 "Statement": [{
 "Effect": "Allow",
 "Principal": {
 "AWS": [
 "arn:aws:iam::123456789012:user/Alice",
 "123456789012"
]
 },
 "Action": [
 "codebuild:BatchGetReportGroups",
 "codebuild:BatchGetReports",
 "codebuild:ListReportsForReportGroup",
 "codebuild:DescribeTestCases"],
 "Resource": "arn:aws:codebuild:us-west-2:123456789012:report-group/my-report-group"
]
}
```

```
}
```

3. Jalankan perintah berikut.

```
aws codebuild put-resource-policy --resource-arn report-group-arn --policy file://
policy.json
```

## Membatalkan berbagi grup laporan bersama

Grup laporan yang tidak dibagikan, termasuk laporan dan hasil uji kasusnya, hanya dapat diakses oleh pemiliknya. Jika Anda membatalkan berbagi grup laporan, adaAWSakun atau pengguna yang sebelumnya Anda bagikan tidak dapat mengakses grup laporan, laporannya, atau hasil kasus uji dalam laporan.

Untuk membatalkan berbagi grup laporan bersama yang Anda miliki, Anda harus menghapusnya dari berbagi sumber daya. Anda dapat menggunakanAWS RAMkonsol atauAWS CLIuntuk melakukan ini.

Untuk membatalkan berbagi grup laporan bersama yang Anda miliki (AWS RAMkonsol)

Lihat [Memperbarui Pembagian Sumber Daya](#) di Panduan Pengguna.AWS RAM

Untuk membatalkan berbagi grup laporan bersama yang Anda miliki (AWS RAMperintah)

Gunakan perintah [disassociate-resource-share](#).

Untuk unshare grup laporan bahwa Anda memiliki perintah CodeBuild)

Jalankan[hapus-kebijakan sumber daya](#)perintah dan tentukan ARN dari grup laporan yang ingin Anda nyahbagikan:

```
aws codebuild delete-resource-policy --resource-arn report-group-arn
```

## Mengidentifikasi grup laporan bersama

Pemilik dan konsumen dapat menggunakanAWS CLIuntuk mengidentifikasi grup laporan bersama.

Untuk mengidentifikasi dan mendapatkan informasi tentang grup laporan bersama dan laporannya, gunakan perintah berikut:

- Untuk melihat ARN grup laporan yang dibagikan dengan Anda, jalankan[list-shared-report-groups](#):



```
aws codebuild list-shared-report-groups
```

- Untuk melihat ARN laporan dalam grup laporan, jalankan [list-reports-for-report-group](#) menggunakan kelompok laporan ARN:

```
aws codebuild list-reports-for-report-group --report-group-arn report-group-arn
```

- Untuk melihat informasi tentang kasus pengujian dalam laporan, jalankan [describe-test-cases](#) menggunakan laporan ARN:

```
aws codebuild describe-test-cases --report-arn report-arn
```

Outputnya terlihat seperti berikut ini:

```
{
 "testCases": [
 {
 "status": "FAILED",
 "name": "Test case 1",
 "expired": 1575916770.0,
 "reportArn": "report-arn",
 "prefix": "Cucumber tests for agent",
 "message": "A test message",
 "durationInNanoSeconds": 1540540,
 "testRawDataPath": "path-to-output-report-files"
 },
 {
 "status": "SUCCEEDED",
 "name": "Test case 2",
 "expired": 1575916770.0,
 "reportArn": "report-arn",
 "prefix": "Cucumber tests for agent",
 "message": "A test message",
 "durationInNanoSeconds": 1540540,
 "testRawDataPath": "path-to-output-report-files"
 }
]
}
```

## Izin grup bersama

### Izin untuk pemilik

Pemilik grup laporan dapat mengedit grup laporan dan menentukannya dalam proyek untuk menjalankan laporan.

### Izin untuk konsumen

Konsumen grup laporan dapat melihat grup laporan, laporannya, dan hasil uji untuk laporannya. Konsumen tidak dapat mengedit grup laporan atau laporannya, dan tidak dapat menggunakannya untuk membuat laporan.

## Bekerja dengan laporan

Laporan berisi hasil kasus uji yang ditentukan untuk satu kelompok laporan. Laporan pengujian dibuat selama menjalankan proyek build. Anda menentukan grup laporan, file kasus uji, dan perintah untuk menjalankan kasus pengujian dalam file buildspec. Setiap kali kasus uji dijalankan, laporan pengujian baru dibuat di grup laporan.

Laporan pengujian berakhir 30 hari setelah dibuat. Anda tidak dapat melihat laporan pengujian yang kedaluwarsa, tetapi Anda dapat mengekspor hasil pengujian ke file hasil pengujian mentah dalam bucket S3. File pengujian mentah yang diekspor tidak kedaluwarsa. Untuk informasi selengkapnya, lihat [Memperbarui grup laporan](#).

Status laporan pengujian dapat berupa salah satu dari yang berikut:

- **GENERATING:** Jalankan kasus uji masih berlangsung.
- **DELETING:** Laporan pengujian sedang dihapus. Ketika laporan pengujian dihapus, kasus pengujianya juga dihapus. File data hasil pengujian mentah yang diekspor ke bucket S3 tidak dihapus.
- **INCOMPLETE:** Laporan pengujian belum selesai. Status ini dapat dikembalikan karena salah satu alasan berikut:
  - Masalah dengan konfigurasi grup laporan yang menentukan kasus uji laporan ini. Misalnya, jalur ke kasus pengujian di bawah grup laporan dalam file buildspec mungkin salah.
  - Pengguna IAM yang menjalankan build tidak memiliki izin untuk menjalankan pengujian. Untuk informasi selengkapnya, lihat [Bekerja dengan izin laporan pengujian](#).
  - Build tidak selesai karena kesalahan yang tidak terkait dengan pengujian.

- SUCCEEDED: Semua kasus uji berhasil.
- FAILED: Beberapa kasus uji tidak berhasil.

Setiap kasus uji mengembalikan status. Status untuk kasus uji dapat berupa salah satu dari yang berikut:

- SUCCEEDED: Kasus uji lulus.
- FAILED: Kasus uji gagal.
- ERROR: Kasus uji menghasilkan kesalahan yang tidak terduga.
- SKIPPED: Kasus uji tidak berjalan.
- UNKNOWN: Kasus uji mengembalikan status selain SUCCEEDED, FAILED, ERROR, atau SKIPPED.

Laporan pengujian dapat memiliki maksimal 500 hasil kasus uji. Jika lebih dari 500 kasus uji dijalankan, CodeBuild prioritaskan pengujian dengan status FAILED dan potong hasil kasus uji.

## Bekerja dengan izin laporan pengujian

Topik ini menjelaskan informasi penting tentang izin yang terkait dengan pelaporan pengujian.

Topik

- [Buat peran untuk laporan pengujian](#)
- [Izin untuk operasi pelaporan pengujian](#)
- [Contoh izin pelaporan uji](#)


## Buat peran untuk laporan pengujian

Untuk menjalankan laporan pengujian, dan memperbarui proyek agar menyertakan laporan pengujian, peran IAM Anda memerlukan izin berikut. Izin ini disertakan dalam kebijakan AWS terkelola yang telah ditentukan sebelumnya. Jika ingin menambahkan pelaporan pengujian ke proyek build yang sudah ada, Anda harus menambahkan izin ini sendiri.

- CreateReportGroup
- CreateReport
- UpdateReport

- BatchPutTestCases

Untuk menjalankan laporan cakupan kode, peran IAM Anda juga harus menyertakan BatchPutCodeCoverages izin.

 Note

BatchPutTestCases, CreateReport, UpdateReport, dan BatchPutCodeCoverages bukan izin publik. Anda tidak dapat memanggil AWS CLI perintah atau metode SDK yang sesuai untuk izin ini.

Untuk memastikan Anda memiliki izin ini, Anda dapat melampirkan kebijakan berikut ke peran IAM Anda:

```
{
 "Effect": "Allow",
 "Resource": [
 "*"
],
 "Action": [
 "codebuild:CreateReportGroup",
 "codebuild:CreateReport",
 "codebuild:UpdateReport",
 "codebuild:BatchPutTestCases",
 "codebuild:BatchPutCodeCoverages"
]
}
```

Kami menyarankan Anda membatasi kebijakan ini hanya untuk grup laporan yang harus Anda gunakan. Berikut ini membatasi izin hanya untuk grup laporan dengan dua ARN dalam kebijakan:

```
{
 "Effect": "Allow",
 "Resource": [
 "arn:aws:codebuild:your-region:your-aws-account-id:report-group/report-group-name-1",
 "arn:aws:codebuild:your-region:your-aws-account-id:report-group/report-group-name-2"
],
}
```

```
"Action": [
 "codebuild:CreateReportGroup",
 "codebuild:CreateReport",
 "codebuild:UpdateReport",
 "codebuild:BatchPutTestCases",
 "codebuild:BatchPutCodeCoverages"
]
}
```

Berikut ini membatasi izin untuk hanya melaporkan grup yang dibuat dengan menjalankan build proyek bernama: `my-project`

```
{
 "Effect": "Allow",
 "Resource": [
 "arn:aws:codebuild:your-region:your-aws-account-id:report-group/my-project-*"
],
 "Action": [
 "codebuild:CreateReportGroup",
 "codebuild:CreateReport",
 "codebuild:UpdateReport",
 "codebuild:BatchPutTestCases",
 "codebuild:BatchPutCodeCoverages"
]
}
```

#### Note

Peran CodeBuild layanan yang ditentukan dalam proyek digunakan untuk izin untuk mengunggah ke bucket S3.

## Izin untuk operasi pelaporan pengujian

Anda dapat menentukan izin untuk operasi CodeBuild API pelaporan pengujian berikut:

- `BatchGetReportGroups`
- `BatchGetReports`
- `CreateReportGroup`
- `DeleteReportGroup`

- DeleteReport
- DescribeTestCases
- ListReportGroups
- ListReports
- ListReportsForReportGroup
- UpdateReportGroup

Untuk informasi selengkapnya, lihat [AWS CodeBuild referensi izin](#).

## Contoh izin pelaporan uji

Untuk informasi tentang kebijakan sampel yang terkait dengan pelaporan pengujian, lihat berikut ini:

- [Izinkan pengguna mengubah grup laporan](#)
- [Izinkan pengguna membuat grup laporan](#)
- [Memungkinkan pengguna untuk menghapus laporan](#)
- [Izinkan pengguna menghapus grup laporan](#)
- [Memungkinkan pengguna untuk mendapatkan informasi tentang grup laporan](#)
- [Memungkinkan pengguna untuk mendapatkan informasi tentang laporan](#)
- [Memungkinkan pengguna untuk mendapatkan daftar grup laporan](#)
- [Memungkinkan pengguna untuk mendapatkan daftar laporan](#)
- [Memungkinkan pengguna untuk mendapatkan daftar laporan untuk grup laporan](#)
- [Memungkinkan pengguna untuk mendapatkan daftar kasus uji untuk laporan](#)

## Lihat laporan pengujian

Anda dapat melihat detail tentang laporan pengujian, seperti informasi tentang kasus uji, nomor lulus dan gagal, dan berapa lama waktu yang dibutuhkan untuk menjalankannya. Anda dapat melihat laporan pengujian yang dikelompokkan berdasarkan build run, grup laporan, atau AWS akun Anda. Pilih laporan pengujian di konsol untuk melihat detail dan hasil kasus pengujiannya.

Anda dapat melihat laporan pengujian yang tidak kedaluwarsa. Laporan pengujian kedaluwarsa 30 hari setelah dibuat. Anda tidak dapat melihat laporan kedaluwarsa di CodeBuild.

## Topik

- [Melihat laporan pengujian untuk build](#)
- [Melihat laporan pengujian untuk grup laporan](#)
- [Melihat laporan pengujian di AWS akun Anda](#)

## Melihat laporan pengujian untuk build

Untuk melihat laporan pengujian untuk build

1. Buka AWS CodeBuild konsol di <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Temukan build yang ingin Anda lihat. Jika Anda mengetahui proyek yang menjalankan build yang membuat laporan pengujian:
  1. Di panel navigasi, pilih Build projects, lalu pilih project dengan build yang menjalankan laporan pengujian yang ingin Anda lihat.
  2. Pilih Riwayat build, lalu pilih build yang menjalankan membuat laporan yang ingin Anda lihat.

Anda juga dapat menemukan build dalam riwayat build untuk AWS akun Anda:

1. Di panel navigasi, pilih Riwayat build, lalu pilih build yang membuat laporan pengujian yang ingin Anda lihat.
3. Di halaman build, pilih Laporan, lalu pilih laporan pengujian untuk melihat detailnya.

## Melihat laporan pengujian untuk grup laporan

Untuk melihat laporan pengujian dalam grup laporan

1. Buka AWS CodeBuild konsol di <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Di panel navigasi, pilih Laporkan grup.
3. Pilih grup laporan yang berisi laporan pengujian yang ingin Anda lihat.
4. Pilih laporan pengujian untuk melihat detailnya.

## Melihat laporan pengujian di AWS akun Anda

Untuk melihat laporan pengujian di AWS akun Anda

1. Buka AWS CodeBuild konsol di <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Di panel navigasi, pilih Laporan riwayat.
3. Pilih laporan pengujian untuk melihat detailnya.

## Pelaporan pengujian dengan kerangka kerja pengujian

Topik di bagian ini menunjukkan cara mengatur pelaporan pengujian AWS CodeBuild untuk berbagai kerangka pengujian.

Topik

- [Siapkan pelaporan tes dengan Jasmine](#)
- [Siapkan pelaporan tes dengan Jest](#)
- [Siapkan pelaporan pengujian dengan pytest](#)
- [Siapkan pelaporan pengujian dengan RSpec](#)

## Siapkan pelaporan tes dengan Jasmine

Prosedur berikut menunjukkan cara mengonfigurasi pelaporan tes di AWS CodeBuild dengan [Kerangka pengujian JasmineBDD](#).

Prosedur ini membutuhkan prasyarat berikut:

- Anda masih memiliki proyek CodeBuild.
- Proyek Anda adalah proyek Node.js yang diatur untuk menggunakan kerangka pengujian Jasmine.

Tambahkan [jasmine-reporters](#) paket ke devDependencies bagian dari proyek Anda package.json file. Paket ini memiliki koleksi kelas reporter JavaScript yang dapat digunakan dengan Jasmine.

```
npm install --save-dev jasmine-reporters
```

Jika belum ada, tambahkan testscript untuk proyek Anda package.json file. Parameter testscript memastikan bahwa Jasmine disebut ketika npm test dijalankan.



```
{
 "scripts": {
 "test": "npm run jasmine"
 }
}
```

CodeBuild mendukung wartawan uji Jasmine berikut:

### JUnitXmlReporter

Digunakan untuk menghasilkan laporan diJUnitXmlformat.

### NUnitXmlReporter

Digunakan untuk menghasilkan laporan diNUnitXmlformat.

Sebuah proyek Node.js dengan Jasmine akan, secara default, memiliki specs sub-direktori, yang berisi konfigurasi Jasmine dan test script.

Untuk mengkonfigurasi Jasmine untuk menghasilkan laporan diJUnitXMLformat, instantiateJUnitXmlReporter dengan menambahkan kode berikut untuk tes Anda.

```
var reporters = require('jasmine-reporters');

var junitReporter = new reporters.JUnitXmlReporter({
 savePath: <test report directory>,
 filePrefix: <report filename>,
 consolidateAll: true
});

jasmine.getEnv().addReporter(junitReporter);
```

Untuk mengkonfigurasi Jasmine untuk menghasilkan laporan diNUnitXMLformat, instantiateNUnitXmlReporter dengan menambahkan kode berikut untuk tes Anda.

```
var reporters = require('jasmine-reporters');

var nunitReporter = new reporters.NUnitXmlReporter({
 savePath: <test report directory>,
 filePrefix: <report filename>,
 consolidateAll: true
});
```

```
});

jasmine.getEnv().addReporter(nunitReporter)
```

Laporan pengujian diekspor ke file yang ditentukan oleh `<test report directory>/<report filename>`.

Dalam `buildspec.yml` file, menambahkan/memperbarui bagian berikut.

```
version: 0.2

phases:
 pre_build:
 commands:
 - npm install
 build:
 commands:
 - npm build
 - npm test

reports:
 jasmine_reports:
 files:
 - <report filename>
 file-format: JUNITXML
 base-directory: <test report directory>
```

Jika Anda menggunakan `NunitXml` format laporan, mengubah `file-format` nilai berikut.

```
file-format: NUNITXML
```

## Siapkan pelaporan tes dengan Jest

Prosedur berikut menunjukkan cara mengonfigurasi pelaporan tes di AWS CodeBuild dengan [Kerangka pengujian bercanda](#).

Prosedur ini membutuhkan prasyarat berikut:

- Anda memiliki proyek CodeBuild.
- Proyek Anda adalah proyek Node.js yang diatur untuk menggunakan kerangka pengujian Jest.

Tambahkan `jest-junit` paket ke `devDependencies` bagian dari proyek Anda `package.json` file. CodeBuild menggunakan paket ini untuk menghasilkan laporan di `JunitXml` format.

```
npm install --save-dev jest-junit
```

Jika belum ada, tambahkan `testscript` untuk proyek Anda `package.json` file. Parameter `testscript` memastikan bahwa Jest dipanggil ketika `npm test` dijalankan.

```
{
 "scripts": {
 "test": "jest"
 }
}
```

Konfigurasi Jest untuk menggunakan `JunitXmlReporter` dengan menambahkan yang berikut ini ke file konfigurasi Jest Anda. Jika proyek Anda tidak memiliki file konfigurasi Jest, buat file bernama `jest.config.js` di akar proyek Anda dan tambahkan yang berikut ini. Laporan pengujian diekspor ke file yang ditentukan oleh `<test report directory>/<report filename>`.

```
module.exports = {
 reporters: [
 'default',
 ['jest-junit', {
 outputDirectory: <test report directory>,
 outputName: <report filename>,
 }]
]
};
```

Dalam `buildspec.yml` file, tambahkan/memperbarui bagian berikut.

```
version: 0.2

phases:
 pre_build:
 commands:
 - npm install
 build:
 commands:
 - npm build
```

```
- npm test

reports:
 jest_reports:
 files:
 - <report filename>
 file-format: JUNITXML
 base-directory: <test report directory>
```

## Siapkan pelaporan pengujian dengan pytest

Prosedur berikut menunjukkan cara mengonfigurasi pelaporan tes di AWS CodeBuild dengan [kerangka pengujian pytest](#).

Prosedur ini membutuhkan prasyarat berikut:

- Anda masih memiliki proyek CodeBuild.
- Proyek Anda adalah proyek Python yang diatur untuk menggunakan kerangka pengujian pytest.

Tambahkan entri berikut ke salah satu `buildspec.yml` file. Kode ini secara otomatis menemukan tes di direktori saat ini dan ekspor laporan pengujian ke file yang ditentukan oleh `<test report directory>/<report filename>`. Laporan menggunakan `JunitXml` format.

```
- python -m pytest --junitxml=<test report directory>/<report filename>
```

Dalam `buildspec.yml` file, menambahkan/memperbarui bagian berikut.

```
version: 0.2

phases:
 install:
 runtime-versions:
 python: 3.7
 commands:
 - pip3 install pytest
 build:
 commands:
 - python -m pytest --junitxml=<test report directory>/<report filename>
```

```
reports:
 pytest_reports:
 files:
 - <report filename>
 base-directory: <test report directory>
 file-format: JUNITXML
```

## Siapkan pelaporan pengujian dengan RSpec

Prosedur berikut menunjukkan cara mengatur pelaporan tes diAWS CodeBuilddengan[Kerangka pengujian RSpec](#).

Prosedur ini membutuhkan prasyarat berikut:

- Anda masih memiliki proyek CodeBuild.
- Proyek Anda adalah proyek Ruby yang diatur untuk menggunakan kerangka pengujian RSpec.

Tambahkan/perbarui hal berikut di`buildspec.yml`File. Kode ini menjalankan tes di`<test source directory>`direktori dan ekspor laporan tes ke file yang ditentukan oleh`<test report directory>/<report filename>`. Laporan menggunakanJunitXmlformat.

```
version: 0.2

phases:
 install:
 runtime-versions:
 ruby: 2.6
 pre_build:
 commands:
 - gem install rspec
 - gem install rspec_junit_formatter
 build:
 commands:
 - rspec <test source directory>/* --format RspecJunitFormatter --out <test report directory>/<report filename>
reports:
 rspec_reports:
 files:
 - <report filename>
 base-directory: <test report directory>
 file-format: JUNITXML
```

## Laporan cakupan kode

CodeBuild memungkinkan Anda untuk menghasilkan laporan cakupan kode untuk pengujian Anda. Laporan cakupan kode berikut disediakan:

### Cakupan baris

Cakupan garis mengukur berapa banyak pernyataan yang dicakup oleh pengujian Anda. Pernyataan adalah instruksi tunggal, tidak termasuk komentar atau kondisional.

```
line coverage = (total lines covered)/(total number of lines)
```

### Cakupan cabang

Cakupan cabang mengukur berapa banyak cabang yang dicakup oleh pengujian Anda dari setiap cabang yang mungkin dari struktur kontrol, seperti if atau case.

```
branch coverage = (total branches covered)/(total number of branches)
```

Format file laporan cakupan kode berikut ini didukung:

- XML
- SimpleCov JSON<sup>1</sup>
- XML Semanggi
- XML

<sup>1</sup> CodeBuild menerima laporan cakupan kode JSON yang dihasilkan oleh [simplecov](#), tidak [simplecov-json](#).

## Membuat laporan cakupan kode

Untuk membuat laporan cakupan kode, Anda menjalankan proyek pembangunan yang dikonfigurasi dengan setidaknya satu grup laporan cakupan kode dalam file buildspec. CodeBuild akan menafsirkan hasil cakupan kode dan memberikan laporan cakupan kode untuk dijalankan. Laporan pengujian baru dihasilkan untuk setiap build berikutnya yang menggunakan file buildspec yang sama.

### Membuat laporan pengujian

1. Membuat proyek pembangunan. Untuk informasi, lihat [Membuat proyek pembangunan diAWS CodeBuild](#).

2. Konfigurasi file buildspec proyek Anda dengan informasi laporan pengujian:
  - a. Tambahkan `reports` bagian dan tentukan nama untuk grup laporan Anda. CodeBuild membuat grup laporan untuk Anda menggunakan nama proyek Anda dan nama yang Anda tentukan dalam format `project-name-report-group-name-in-buildspec`. Jika Anda sudah memiliki grup laporan yang ingin Anda gunakan, tentukan ARN. Jika Anda menggunakan nama alih-alih ARN, CodeBuild membuat grup laporan baru. Untuk informasi selengkapnya, lihat [Reports syntax in the buildspec file](#).
  - b. Di bawah grup laporan, tentukan lokasi file yang berisi hasil cakupan kode. Jika Anda menggunakan lebih dari satu grup laporan, tentukan lokasi file hasil untuk setiap grup laporan. Laporan cakupan kode baru dibuat setiap kali proyek build Anda berjalan. Untuk informasi selengkapnya, lihat [Tentukan file uji](#).

Ini adalah contoh yang menghasilkan laporan cakupan kode untuk file hasil XML JaCoCo terletak di `test-results/jacoco-coverage-report.xml`.

```
reports:
 jacoco-report:
 files:
 - 'test-results/jacoco-coverage-report.xml'
 file-format: 'JACOCOXML'
```

- c. Di `commands` bagian `build` atau `post_build` urutan, menentukan perintah yang menjalankan analisis cakupan kode. Untuk informasi selengkapnya, lihat [Tentukan perintah uji](#).
3. Jalankan membangun proyek build. Untuk informasi selengkapnya, lihat [Jalankan build di AWS CodeBuild](#).
4. Saat build selesai, pilih build run baru Riwayat pembangunan pada halaman proyek Anda. Pilih Laporan untuk melihat laporan cakupan kode. Untuk informasi selengkapnya, lihat [Melihat laporan pengujian untuk build](#).

# Pencatatan dan pemantauan di AWS CodeBuild

Pemantauan adalah bagian penting dari pemeliharaan keandalan, ketersediaan, dan performa AWS CodeBuild serta solusi AWS Anda. Anda harus mengumpulkan data pemantauan dari semua bagian Anda AWS sehingga Anda dapat dengan lebih mudah melakukan debug kegagalan multi-titik, jika ada yang terjadi. AWS menyediakan alat berikut untuk memantau sumber daya CodeBuild Anda dan membangun dan merespons potensi insiden.

## Topik

- [Mencatat panggilan API AWS CodeBuild dengan AWS CloudTrail](#)
- [Memantau AWS CodeBuild](#)

## Mencatat panggilan API AWS CodeBuild dengan AWS CloudTrail

AWS CodeBuild terintegrasi dengan AWS CloudTrail, layanan yang menyediakan catatan tindakan yang diambil oleh pengguna, peran, atau AWS layanan di CodeBuild. CloudTrail menangkap semua panggilan API untuk CodeBuild sebagai peristiwa, termasuk panggilan dari CodeBuild konsol dan panggilan kode ke CodeBuild API. Jika membuat jejak, Anda dapat mengaktifkan pengiriman CloudTrail peristiwa berkelanjutan ke bucket Amazon S3, termasuk peristiwa untuk CodeBuild. Jika Anda tidak membuat konfigurasi jejak, Anda masih dapat melihat kejadian terbaru dalam konsol CloudTrail di Riwayat peristiwa. Menggunakan informasi yang dikumpulkan oleh CloudTrail, Anda dapat menentukan permintaan yang dibuat CodeBuild, alamat IP asal permintaan tersebut dibuat, siapa yang membuat permintaan, kapan permintaan dibuat, dan detail lainnya.

Untuk mempelajari lebih lanjut CloudTrail, lihat [Panduan AWS CloudTrail Pengguna](#).

## AWS CodeBuild informasi dalam CloudTrail

CloudTrail diaktifkan di AWS akun Anda saat Anda membuat akun. Ketika aktivitas terjadi di CodeBuild, aktivitas tersebut dicatat dalam CloudTrail peristiwa bersama peristiwa AWS layanan lainnya di Riwayat peristiwa. Anda dapat melihat, mencari, dan mengunduh peristiwa terbaru di akun AWS Anda. Untuk informasi selengkapnya, lihat [Melihat CloudTrail peristiwa dengan riwayat peristiwa](#) di Panduan AWS CloudTrail Pengguna.

Untuk catatan berkelanjutan tentang peristiwa di AWS akun Anda, termasuk peristiwa untuk CodeBuild, buat jejak. Jejak memungkinkan CloudTrail untuk mengirim file log ke bucket S3. Secara



default, saat Anda membuat jejak di konsol, jejak tersebut berlaku untuk semua wilayah. Jejak mencatat peristiwa dari semua wilayah di AWS partisi dan mengirimkan file log ke bucket Amazon S3 yang Anda tentukan. Anda dapat mengonfigurasi AWS layanan lainnya untuk menganalisis lebih lanjut dan bertindak berdasarkan data peristiwa yang dikumpulkan di CloudTrail log. Untuk informasi selengkapnya, lihat :

- [Gambaran umum untuk membuat jejak](#)
- [CloudTrail Layanan yang didukung dan integrasi](#)
- [Mengonfigurasi notifikasi Amazon SNS untuk CloudTrail](#)
- [Menerima file CloudTrail log dari beberapa wilayah](#) dan [Menerima file CloudTrail log dari beberapa akun](#)

Semua CodeBuild tindakan dicatat oleh CloudTrail dan didokumentasikan dalam [Referensi CodeBuild API](#). Misalnya, panggilan ke `CreateProject` (dalam AWS CLI, `create-project`), `StartBuild` (dalam AWS CLI, `start-project`), dan `UpdateProject` (dalam AWS CLI, `update-project`) tindakan menghasilkan entri dalam file CloudTrail log.

Setiap entri peristiwa atau log berisi informasi tentang siapa yang membuat permintaan tersebut. Informasi identitas membantu Anda menentukan hal berikut:

- Baik permintaan tersebut dibuat dengan kredensial pengguna root atau .
- Baik permintaan tersebut dibuat dengan kredensial keamanan sementara untuk peran atau pengguna gabungan.
- Bahwa permintaan dibuat oleh layanan AWS lain.

Untuk informasi selengkapnya, lihat [elemen CloudTrail userIdentity](#) di Panduan AWS CloudTrail Pengguna.

## Memahami entri file log AWS CodeBuild

Penjejak adalah konfigurasi yang dapat membuat pengiriman peristiwa sebagai file log ke bucket Amazon S3 yang telah Anda tentukan. CloudTrail File log berisi satu atau beberapa entri log. Sebuah peristiwa mewakili permintaan tunggal dari sumber apa pun dan mencakup informasi tentang tindakan yang diminta, tanggal dan waktu tindakan, parameter permintaan, dan sebagainya. CloudTrail File log bukan jejak tumpukan yang diperintahkan dari panggilan API publik, sehingga berkas tersebut tidak muncul dalam urutan tertentu.

**Note**

Untuk melindungi informasi sensitif, berikut ini disembunyikan di CodeBuild log:

- AWSID kunci akses. Untuk informasi selengkapnya, lihat [Mengelola Kunci Akses untuk Pengguna IAM](#) di PanduanAWS Identity and Access Management Pengguna.
- String ditentukan menggunakan Parameter Store. Untuk informasi selengkapnya, lihat Panduan Konsol [Parameter Store](#) dan [Systems Manager Parameter Store Console di Panduan](#) Pengguna Amazon EC2 Systems Manager.
- String ditentukan menggunakanAWS Secrets Manager. Untuk informasi selengkapnya, lihat [Manajemen kunci](#).

Contoh berikut menunjukkan entri CloudTrail log yang menunjukkan pembuatan proyek build di CodeBuild.

```
{
 "eventVersion": "1.05",
 "userIdentity": {
 "type": "FederatedUser",
 "principalId": "account-ID:user-name",
 "arn": "arn:aws:sts::account-ID:federated-user/user-name",
 "accountId": "account-ID",
 "accessKeyId": "access-key-ID",
 "sessionContext": {
 "attributes": {
 "mfaAuthenticated": "false",
 "creationDate": "2016-09-06T17:59:10Z"
 },
 "sessionIssuer": {
 "type": "IAMUser",
 "principalId": "access-key-ID",
 "arn": "arn:aws:iam::account-ID:user/user-name",
 "accountId": "account-ID",
 "userName": "user-name"
 }
 }
 },
 "eventTime": "2016-09-06T17:59:11Z",
 "eventSource": "codebuild.amazonaws.com",
 "eventName": "CreateProject",
```

```
"awsRegion": "region-ID",
"sourceIPAddress": "127.0.0.1",
"userAgent": "user-agent",
"requestParameters": {
 "awsActId": "account-ID"
},
"responseElements": {
 "project": {
 "environment": {
 "image": "image-ID",
 "computeType": "BUILD_GENERAL1_SMALL",
 "type": "LINUX_CONTAINER",
 "environmentVariables": []
 },
 "name": "codebuild-demo-project",
 "description": "This is my demo project",
 "arn": "arn:aws:codebuild:region-ID:account-ID:project/codebuild-demo-
project:project-ID",
 "encryptionKey": "arn:aws:kms:region-ID:key-ID",
 "timeoutInMinutes": 10,
 "artifacts": {
 "location": "arn:aws:s3:::codebuild-region-ID-account-ID-output-bucket",
 "type": "S3",
 "packaging": "ZIP",
 "outputName": "MyOutputArtifact.zip"
 },
 "serviceRole": "arn:aws:iam::account-ID:role/CodeBuildServiceRole",
 "lastModified": "Sep 6, 2016 10:59:11 AM",
 "source": {
 "type": "GITHUB",
 "location": "https://github.com/my-repo.git"
 },
 "created": "Sep 6, 2016 10:59:11 AM"
 }
},
"requestID": "9d32b228-745b-11e6-98bb-23b67EXAMPLE",
"eventID": "581f7dd1-8d2e-40b0-aaaa-0dbf7EXAMPLE",
"eventType": "AwsApiCall",
"recipientAccountId": "account-ID"
}
```

# Memantau AWS CodeBuild

Anda dapat menggunakan Amazon CloudWatch untuk mengawasi build, melaporkan saat terjadi kesalahan, dan mengambil tindakan otomatis jika diperlukan. Anda dapat memantau build Anda di dua tingkat:

## Tingkat proyek

Metrik ini adalah untuk semua membangun dalam proyek yang ditentukan. Untuk melihat metrik proyek, tentukan `ProjectName` untuk dimensi di CloudWatch.

## AWS Tingkat akun

Metrik ini untuk semua build di akun. Untuk melihat metrik di AWS tingkat akun, jangan masukkan dimensi di CloudWatch. Membangun metrik pemanfaatan sumber daya tidak tersedia di AWS tingkat akun.

Metrik CloudWatch menunjukkan perilaku build Anda dari waktu ke waktu. Misalnya, Anda dapat memantau:

- Berapa banyak membangun yang dicoba dalam proyek pembangunan atau AWS akun dari waktu ke waktu.
- Berapa banyak membangun yang berhasil dalam membangun proyek atau AWS akun dari waktu ke waktu.
- Berapa banyak build yang gagal dalam proyek build atau AWS akun dari waktu ke waktu.
- Berapa banyak waktu CodeBuild yang dihabiskan untuk menjalankan build dalam proyek build atau AWS akun dari waktu ke waktu.
- Membangun pemanfaatan sumber daya untuk membangun atau seluruh membangun proyek. Metrik build sumber daya mencakup metrik seperti CPU, dan pemanfaatan penyimpanan.

Untuk informasi selengkapnya, lihat [Memantau metrik CodeBuild](#).

## Metrik CodeBuild

Metrik berikut dapat dilacak per AWS akun atau membangun proyek.

### BuildDuration

Mengukur durasi build BUILD fase.

Unit: Detik

Statistik CloudWatch valid: Rata-rata (direkomendasikan), Maksimum, Minimum

### Bangunan

Mengukur jumlah membangun dipicu.

Unit: Count

Statistik CloudWatch valid: Jumlah

### DownloadSourceUration

Mengukur durasi buildDOWNLOAD\_SOURCEfase.

Unit: Detik

Statistik CloudWatch valid: Rata-rata (direkomendasikan), Maksimum, Minimum

### Duration

Mengukur durasi semua membangun dari waktu ke waktu.

Unit: Detik

Statistik CloudWatch valid: Rata-rata (direkomendasikan), Maksimum, Minimum

### FailedBuilds

Mengukur jumlah build yang gagal karena kesalahan klien atau batas waktu.

Unit: Count

Statistik CloudWatch valid: Jumlah

### Finalisasi Durasi

Mengukur durasi buildFINALIZINGfase.

Unit: Detik

Statistik CloudWatch valid: Rata-rata (direkomendasikan), Maksimum, Minimum

### InstallDuration

Mengukur durasi buildINSTALLfase.

Unit: Detik

Statistik CloudWatch valid: Rata-rata (direkomendasikan), Maksimum, Minimum  
PostbuildDuration

Mengukur durasi buildPOST\_BUILDfase

Unit: Detik

Statistik CloudWatch valid: Rata-rata (direkomendasikan), Maksimum, Minimum  
PrebuildDuration

Mengukur durasi buildPRE\_BUILDfase.

Unit: Detik

Statistik CloudWatch valid: Rata-rata (direkomendasikan), Maksimum, Minimum  
ProvisioningDuration

Mengukur durasi buildPROVISIONINGfase.

Unit: Detik

Statistik CloudWatch valid: Rata-rata (direkomendasikan), Maksimum, Minimum  
Antrian Durasi

Mengukur durasi buildQUEUEDfase.

Unit: Detik

Statistik CloudWatch valid: Rata-rata (direkomendasikan), Maksimum, Minimum  
DisampaikanDurasi

Mengukur durasi buildSUBMITTEDfase.

Unit: Detik

Statistik CloudWatch valid: Rata-rata (direkomendasikan), Maksimum, Minimum  
SucceededBuilds

Mengukur jumlah keberhasilan membangun.

Unit: Count

Statistik CloudWatch valid: Jumlah

## UploadArtifaktaDurasi

Mengukur durasi build `UPLOAD_ARTIFACTS` fase.

Unit: Detik

Statistik CloudWatch valid: Rata-rata (direkomendasikan), Maksimum, Minimum

## Metrik pemanfaatan sumber daya CodeBuild CloudWatch

### Note

Metrik CodeBuild hanya tersedia di wilayah berikut:

- Wilayah Asia Pacific (Tokyo)
- Wilayah Asia Pasifik (Seoul)
- Wilayah Asia Pacific (Mumbai)
- Wilayah Asia Pasifik (Singapore)
- Wilayah Asia Pasifik (Sydney)
- Wilayah Canada (Central)
- Wilayah Eropa (Frankfurt)
- Wilayah Europe (Ireland)
- Wilayah Eropa (London)
- Wilayah Eropa (Paris)
- Wilayah South America (Sao Paulo)
- Wilayah Timur AS (Virginia U.)
- Wilayah US East (Ohio)
- Wilayah US West (N. California)
- Wilayah US West (Oregon)

Metrik pemanfaatan sumber daya berikut dapat dilacak.

### CPUUtilized

Jumlah unit CPU pengolahan dialokasikan digunakan oleh membangun wadah.

Unit: Unit CPU

Statistik CloudWatch valid: Rata-rata (direkomendasikan), Maksimum, Minimum  
CPUUtilizedPercent

Persentase pengolahan dialokasikan digunakan oleh membangun wadah.

Unit: Persen

Statistik CloudWatch valid: Rata-rata (direkomendasikan), Maksimum, Minimum  
MemoryUtilized

Jumlah megabyte memori yang digunakan oleh membangun wadah.

Unit: Megabyte

Statistik CloudWatch valid: Rata-rata (direkomendasikan), Maksimum, Minimum  
MemoryDipfaatkan Persen

Persentase memori yang dialokasikan oleh kontainer build.

Unit: Persen

Statistik CloudWatch valid: Rata-rata (direkomendasikan), Maksimum, Minimum  
StorageReadBytes

Kecepatan baca penyimpanan yang digunakan oleh kontainer build.

Unit: Byte/Detik

Statistik CloudWatch valid: Rata-rata (direkomendasikan), Maksimum, Minimum  
StorageWriteBytes

Kecepatan tulis penyimpanan yang digunakan oleh wadah build.

Unit: Byte/Detik

Statistik CloudWatch valid: Rata-rata (direkomendasikan), Maksimum, Minimum

## Dimensi CodeBuild

CodeBuild menyediakan dimensi metrik CloudWatch berikut. Jika tidak ada yang ditentukan, metrik adalah untuk saat ini AWSakun.



BuildId, buildNumber, ProjectName

Metrik disediakan untuk pengenalan build, nomor build, dan nama proyek.

ProjectName

Metrik disediakan untuk nama proyek.

## Alarm CodeBuild

Anda dapat menggunakan konsol CloudWatch untuk membuat alarm berdasarkan metrik CodeBuild sehingga Anda dapat bereaksi jika ada yang tidak beres dengan build Anda. Dua metrik yang paling berguna dengan alarm adalah:

- **FailedBuild.** Anda dapat membuat alarm yang dipicu ketika sejumlah build gagal terdeteksi dalam jumlah detik yang telah ditentukan. Di CloudWatch, Anda menentukan jumlah detik dan berapa banyak build gagal memicu alarm.
- **Duration.** Anda dapat membuat alarm yang dipicu saat build membutuhkan waktu lebih lama dari yang diharapkan. Anda menentukan berapa detik yang harus berlalu setelah build dimulai dan sebelum build selesai sebelum alarm dipicu.

Untuk informasi tentang cara membuat alarm untuk metrik CodeBuild, lihat [Monitoring dibangun dengan alarm CloudWatch](#). Untuk informasi selengkapnya tentang alarm, lihat [Membuat Amazon alarm CloudWatch](#) di dalam Panduan Pengguna Amazon CloudWatch.

## Memantau metrik CodeBuild

AWS CodeBuild memonitor fungsi atas nama Anda dan melaporkan metrik melalui Amazon CloudWatch. Metrik ini mencakup jumlah total build, build yang gagal, build yang berhasil, dan durasi build.

Anda dapat menggunakan konsol CodeBuild atau konsol CloudWatch untuk memantau metrik CodeBuild. Prosedur berikut menunjukkan cara mengakses metrik.

Topik

- [Akses metrik build \(konsol CodeBuild\)](#)
- [Akses metrik build \(Konsol Amazon CloudWatch\)](#)

## Akses metrik build (konsol CodeBuild)

### Note

Anda tidak dapat menyesuaikan metrik atau grafik yang digunakan untuk menampilkannya di konsol CodeBuild. Jika Anda ingin menyesuaikan layar, gunakan konsol Amazon CloudWatch untuk melihat metrik build Anda.

### Metrik tingkat akun

#### Mengakses AWS Metrik tingkat akun

1. Masuk ke AWS Management Console dan membuka AWS CodeBuild konsol di <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Di panel navigasi, pilih Metrik akun.

### Metrik tingkat proyek

#### Untuk mengakses metrik tingkat proyek

1. Masuk ke AWS Management Console dan membuka AWS CodeBuild konsol di <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Di panel navigasi, pilih Proyek pembangunan.
3. Dalam daftar proyek pembangunan, di Nama kolom, pilih proyek di mana metrik ingin Anda lihat.
4. Pilih tab Metrik.

## Akses metrik build (Konsol Amazon CloudWatch)

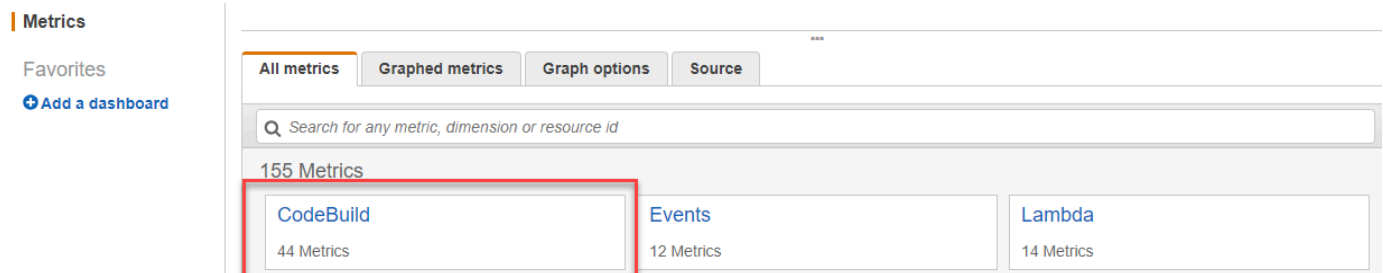
Anda dapat menyesuaikan metrik dan grafik yang digunakan untuk menampilkannya dengan konsol CloudWatch.

### Metrik tingkat akun

#### Mengakses metrik tingkat akun

1. Masuk ke AWS Management Console dan buka konsol CloudWatch di <https://console.aws.amazon.com/cloudwatch/>.

2. Di panel navigasi, pilih Metrik.
3. Pada Semua metrik tab, pilih CodeBuild.

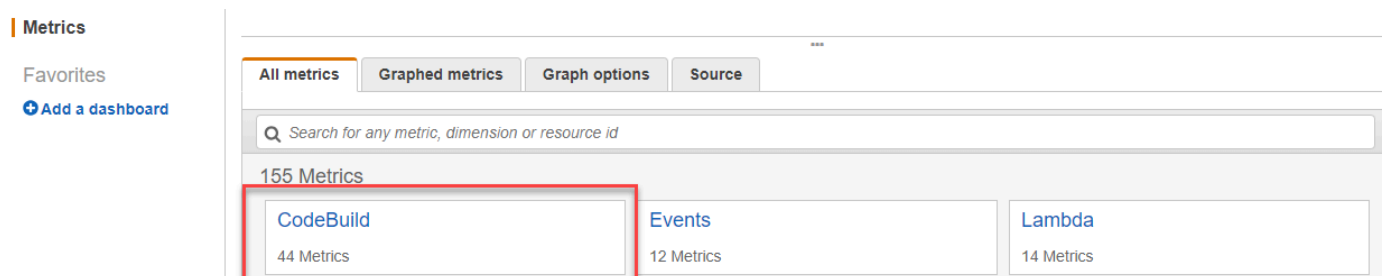


4. Silahkan Anda memilih Metrik Akun.
5. Pilih satu proyek atau beberapa metrik. Untuk setiap proyek, Anda dapat memilih SucceededBuilds, FailedBuilds, Build, dan Durasimetrik. Semua proyek yang dipilih dan kombinasi metrik ditampilkan dalam grafik pada halaman.

## Metrik tingkat proyek

Untuk mengakses metrik tingkat proyek

1. Masuk ke AWS Management Console dan buka konsol CloudWatch di <https://console.aws.amazon.com/cloudwatch/>.
2. Di panel navigasi, pilih Metrik.
3. Pada Semua metrik tab, pilih CodeBuild.



4. Pilih Dengan Proyek.
5. Pilih satu atau beberapa kombinasi proyek dan metrik. Untuk setiap proyek, Anda dapat memilih SucceededBuilds, FailedBuilds, Build, dan Durasimetrik. Semua proyek yang dipilih dan kombinasi metrik ditampilkan dalam grafik pada halaman.
6. (Opsional) Anda dapat menyesuaikan metrik dan grafik Anda. Misalnya, dari daftar tarik-turun di Statistik kolom, Anda dapat memilih statistik yang berbeda untuk ditampilkan. Atau dari menu tarik-turun di Periode kolom, Anda dapat memilih periode waktu yang berbeda untuk digunakan untuk memantau metrik.

Untuk informasi selengkapnya, lihat [Metrik grafik](#) dan [Lihat metrik yang tersedia](#) di Panduan Pengguna Amazon CloudWatch.

## Pemantauan metrik pemanfaatan sumber daya CodeBuild

AWS CodeBuild memonitor pemanfaatan sumber daya membangun atas nama Anda dan melaporkan metrik melalui Amazon CloudWatch. Ini termasuk metrik seperti CPU, memori, dan pemanfaatan penyimpanan.

### Note

Metrik pemanfaatan sumber daya CodeBuild hanya direkam untuk build yang berjalan selama lebih dari satu menit.

Anda dapat menggunakan konsol CodeBuild atau konsol CloudWatch untuk memantau metrik pemanfaatan sumber daya untuk CodeBuild.

### Note

Metrik pemanfaatan sumber daya CodeBuild hanya tersedia di wilayah berikut:

- Wilayah Asia Pacific (Tokyo)
- Wilayah Asia Pasifik (Seoul)
- Wilayah Asia Pacific (Mumbai)
- Wilayah Asia Pasifik (Singapore)
- Wilayah Asia Pasifik (Sydney)
- Wilayah Canada (Central)
- Wilayah Eropa (Frankfurt)
- Wilayah Europe (Ireland)
- Wilayah Eropa (London)
- Wilayah Eropa (Paris)
- Wilayah South America (Sao Paulo)
- Wilayah Timur AS (Virginia U.)
- Wilayah US East (Ohio)

- Wilayah US West (N. California)
- Wilayah US West (Oregon)

Prosedur berikut menunjukkan cara mengakses metrik pemanfaatan sumber daya Anda.

Topik

- [Mengakses metrik pemanfaatan sumber daya \(CodeBuild konsol\)](#)
- [Mengakses metrik pemanfaatan sumber daya \(konsol Amazon CloudWatch\)](#)

## Mengakses metrik pemanfaatan sumber daya (CodeBuild konsol)

### Note

Anda tidak dapat menyesuaikan metrik atau grafik yang digunakan untuk menampilkannya di konsol CodeBuild. Jika Anda ingin menyesuaikan layar, gunakan konsol Amazon CloudWatch untuk melihat metrik build Anda.

Metrik pemanfaatan sumber daya tingkat proyek

Untuk mengakses metrik pemanfaatan sumber daya tingkat proyek

1. Masuk keAWS Management Console dan bukaAWS CodeBuildkonsol di<https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Di panel navigasi, pilihProyek pembangunan.
3. Dalam daftar proyek pembangunan, diNamakolom, pilih proyek yang ingin Anda lihat metrik pemanfaatan untuk.
4. Pilih tab Metrik. Metrik pemanfaatan sumber daya ditampilkan dalamPemanfaatan sumber daya metrikbagian.
5. Untuk melihat metrik pemanfaatan sumber daya tingkat proyek di konsol CloudWatch, pilihLihat di CloudWatchdiPemanfaatan sumber daya metrikbagian.

## Metrik pemanfaatan sumber daya tingkat bangunan

Untuk mengakses metrik pemanfaatan sumber daya tingkat bangunan

1. Masuk ke AWS Management Console dan buka AWS CodeBuild konsol di <https://console.aws.amazon.com/codesuite/codebuild/home>.
2. Di panel navigasi, pilih Riwayat pembangunan.
3. Dalam daftar membangun, di Jalankan pembangunankolom, pilih build yang ingin Anda lihat metrik pemanfaatan.
4. Pilih Pemanfaatan sumber daya tab.
5. Untuk melihat metrik pemanfaatan sumber daya tingkat bangunan di konsol CloudWatch, pilih Lihat di CloudWatch di Pemanfaatan sumber daya metrik bagian.

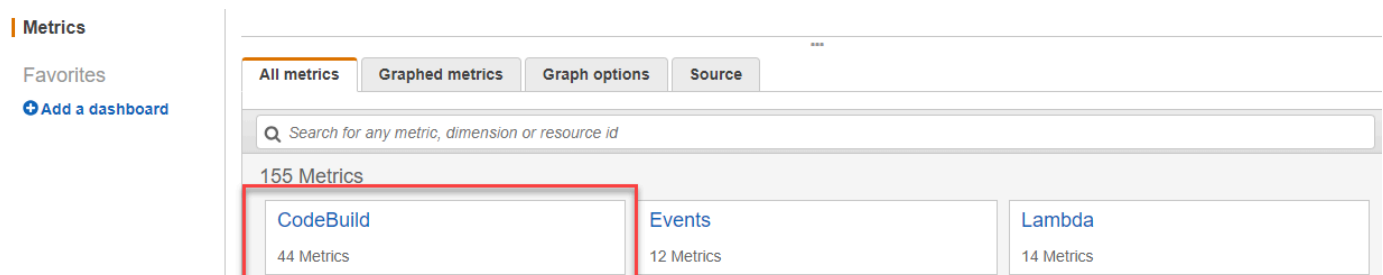
## Mengakses metrik pemanfaatan sumber daya (konsol Amazon CloudWatch)

Konsol Amazon CloudWatch dapat digunakan untuk mengakses metrik pemanfaatan sumber daya CodeBuild.

## Metrik pemanfaatan sumber daya tingkat proyek

Untuk mengakses metrik pemanfaatan sumber daya tingkat proyek

1. Masuk ke AWS Management Console dan buka konsol CloudWatch di <https://console.aws.amazon.com/cloudwatch/>.
2. Di panel navigasi, pilih Metrik.
3. Pada Semua metrik tab, pilih CodeBuild.



4. Pilih Proyek.
5. Pilih satu atau lebih kombinasi proyek dan metrik untuk ditambahkan ke grafik. Semua proyek yang dipilih dan kombinasi metrik ditampilkan dalam grafik pada halaman.

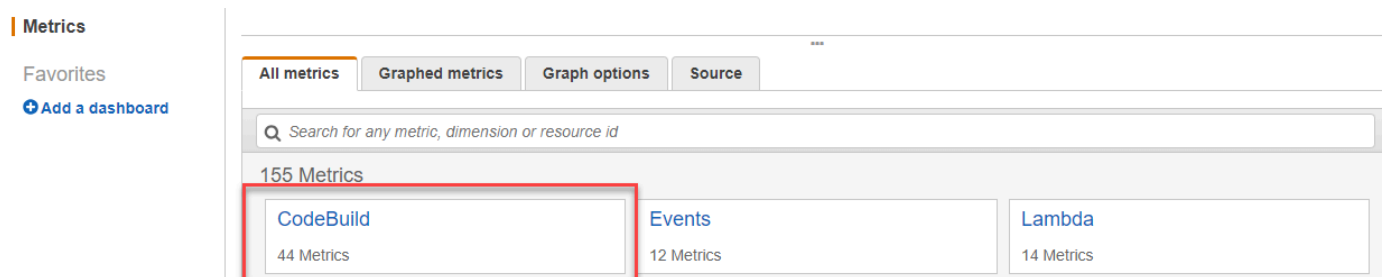
- (Opsional) Anda dapat menyesuaikan metrik dan grafik dari Metrik grafik tab. Misalnya, dari daftar tarik-turun di Statistik kolom, Anda dapat memilih statistik yang berbeda untuk ditampilkan. Atau dari menu tarik-turun di Periode kolom, Anda dapat memilih periode waktu yang berbeda untuk digunakan untuk memantau metrik.

Untuk informasi selengkapnya, lihat [Membuat grafik metrik](#) dan [Melihat metrik yang tersedia](#) di Panduan Pengguna Amazon CloudWatch.

## Metrik pemanfaatan sumber daya tingkat bangunan

Untuk mengakses metrik pemanfaatan sumber daya tingkat bangunan

- Masuk ke AWS Management Console dan buka konsol CloudWatch di <https://console.aws.amazon.com/cloudwatch/>.
- Di panel navigasi, pilih Metrik.
- Pada Semua metrik tab, pilih CodeBuild.



- Pilih BuildId, buildNumber, ProjectName.
- Pilih satu atau lebih kombinasi build dan metrik untuk ditambahkan ke grafik. Semua kombinasi build dan metrik yang dipilih ditampilkan dalam grafik pada halaman.
- (Opsional) Anda dapat menyesuaikan metrik dan grafik dari Metrik grafik tab. Misalnya, dari daftar tarik-turun di Statistik kolom, Anda dapat memilih statistik yang berbeda untuk ditampilkan. Atau dari menu tarik-turun di Periode kolom, Anda dapat memilih periode waktu yang berbeda untuk digunakan untuk memantau metrik.

Untuk informasi selengkapnya, lihat [Membuat grafik metrik](#) dan [Melihat metrik yang tersedia](#) di Panduan Pengguna Amazon CloudWatch.

## Monitoring dibangun dengan alarm CloudWatch

Anda dapat membuat alarm CloudWatch untuk build Anda. Alarm melihat metrik tunggal selama periode waktu yang Anda tentukan dan melakukan satu atau beberapa tindakan berdasarkan nilai metrik relatif terhadap ambang batas tertentu selama sejumlah periode waktu. Menggunakan fungsi alarm CloudWatch asli, Anda dapat menentukan salah satu tindakan yang didukung oleh CloudWatch ketika ambang batas terlampaui. Misalnya, Anda dapat menentukan bahwa pemberitahuan Amazon SNS dikirim ketika lebih dari tiga build di akun Anda gagal dalam waktu lima belas menit.

### Membuat alarm CloudWatch untuk metrik CodeBuild

1. Masuk ke AWS Management Console dan buka konsol CloudWatch di <https://console.aws.amazon.com/cloudwatch/>.
2. Di panel navigasi, pilih Alarm.
3. Pilih Buat Alarm.
4. Di bawah Metrik CloudWatch berdasarkan Kategori, pilih Metrik CodeBuild. Jika Anda tahu Anda hanya ingin metrik tingkat proyek, pilih Dengan Proyek. Jika Anda tahu Anda hanya ingin metrik tingkat akun, pilih Metrik Akun.
5. Pada Buat alarm, jika belum dipilih, pilih Pilih Metrik.
6. Pilih metrik yang ingin Anda buat alarm. Pilihannya adalah Dengan Proyek atau Metrik Akun.
7. Pilih Selanjutnya atau Tentukan alarm dan kemudian membuat alarm Anda. Untuk informasi selengkapnya, lihat [Membuat alarm Amazon CloudWatch](#) di Panduan Pengguna Amazon CloudWatch. Untuk informasi selengkapnya tentang pengaturan notifikasi Amazon SNS saat alarm dipicu, lihat [Mengatur notifikasi Amazon SNS](#) di Panduan Developer Amazon SNS.
8. Pilih Buat Alarm.



# Keamanan di AWS CodeBuild

Keamanan cloud di AWS adalah prioritas tertinggi. Sebagai AWS pelanggan, Anda mendapat manfaat dari pusat data dan arsitektur jaringan yang dibangun untuk memenuhi persyaratan organisasi yang paling sensitif terhadap keamanan.

Keamanan dan kepatuhan adalah tanggung jawab bersama antara Anda AWS dan Anda. Model bersama ini dapat membantu meringankan beban operasional Anda: AWS mengoperasikan, mengelola, dan mengontrol komponen dari sistem operasi host dan lapisan virtualisasi hingga keamanan fisik fasilitas layanan. Anda bertanggung jawab dan mengelola sistem operasi tamu (termasuk pembaruan dan patch keamanan) dan perangkat lunak aplikasi terkait lainnya. Anda juga bertanggung jawab atas konfigurasi firewall grup keamanan yang AWS disediakan. Tanggung jawab Anda bervariasi dengan layanan yang Anda gunakan, integrasi layanan tersebut ke dalam lingkungan TI Anda, dan hukum dan peraturan yang berlaku. Karena itu, Anda harus mempertimbangkan dengan cermat layanan yang digunakan organisasi Anda. Untuk informasi selengkapnya, lihat [Model tanggung jawab bersama](#).

Untuk mempelajari cara mengamankan CodeBuild sumber daya Anda, lihat topik berikut.

## Topik

- [Perlindungan data di AWS CodeBuild](#)
- [Manajemen identitas dan akses di AWS CodeBuild](#)
- [Validasi kepatuhan untuk AWS CodeBuild](#)
- [Ketahanan di AWS CodeBuild](#)
- [Keamanan infrastruktur di AWS CodeBuild](#)
- [Akses penyedia sumber Anda di CodeBuild](#)
- [Pencegahan confused deputy lintas layanan](#)

## Perlindungan data di AWS CodeBuild

[Model tanggung jawab AWS bersama model](#) berlaku untuk perlindungan data di AWS CodeBuild. Seperti yang dijelaskan dalam model AWS ini, bertanggung jawab untuk melindungi infrastruktur global yang menjalankan semua AWS Cloud. Anda bertanggung jawab untuk memelihara kendali atas isi yang dihost pada infrastruktur ini. Anda juga bertanggung jawab atas tugas-tugas konfigurasi dan manajemen keamanan untuk Layanan AWS yang Anda gunakan. Lihat informasi yang lebih

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

Untuk tujuan perlindungan data, kami menyarankan Anda melindungi Akun AWS kredensial dan mengatur pengguna individu dengan AWS IAM Identity Center atau AWS Identity and Access Management (IAM). Dengan cara itu, setiap pengguna hanya diberi izin yang diperlukan untuk memenuhi tanggung jawab tugasnya. Kami juga menyarankan supaya Anda mengamankan data dengan cara-cara berikut:

- Gunakan autentikasi multi-faktor (MFA) pada setiap akun.
- Gunakan SSL/TLS untuk berkomunikasi dengan sumber daya. AWS Kami mensyaratkan TLS 1.2 dan menganjurkan TLS 1.3.
- Siapkan API dan pencatatan aktivitas pengguna dengan AWS CloudTrail.
- Gunakan solusi AWS enkripsi, bersama dengan semua kontrol keamanan default di dalamnya Layanan AWS.
- Gunakan layanan keamanan terkelola lanjut seperti Amazon Macie, yang membantu menemukan dan mengamankan data sensitif yang disimpan di Amazon S3.
- Jika Anda memerlukan modul kriptografi tervalidasi FIPS 140-2 saat mengakses AWS melalui antarmuka baris perintah atau API, gunakan titik akhir FIPS. Lihat informasi yang lebih lengkap tentang titik akhir FIPS yang tersedia di [Standar Pemrosesan Informasi Federal \(FIPS\) 140-2](#).

Kami sangat merekomendasikan agar Anda tidak pernah memasukkan informasi identifikasi yang sensitif, seperti nomor rekening pelanggan Anda, ke dalam tanda atau bidang isian bebas seperti bidang Nama. Ini termasuk saat Anda bekerja dengan CodeBuild atau lainnya Layanan AWS menggunakan konsol, API AWS CLI, atau AWS SDK. Data apa pun yang Anda masukkan ke dalam tanda atau bidang isian bebas yang digunakan untuk nama dapat digunakan untuk log penagihan atau log diagnostik. Saat Anda memberikan URL ke server eksternal, kami sangat menganjurkan supaya Anda tidak menyertakan informasi kredensial di dalam URL untuk memvalidasi permintaan Anda ke server itu.

Untuk melindungi informasi sensitif, berikut ini disembunyikan di CodeBuild log:

- String ditentukan menggunakan Parameter Store dalam variabel lingkungan CodeBuild proyek atau bagian `env/parameter-store` buildspec. Untuk informasi selengkapnya, lihat [panduan konsol Systems Manager Parameter Store dan Systems Manager Parameter Store di Panduan Pengguna Amazon EC2 Systems Manager](#).

- String ditentukan menggunakan variabel AWS Secrets Manager lingkungan CodeBuild proyek atau bagian `env/secrets-manager` `buildspec`. Untuk informasi selengkapnya, lihat [Manajemen kunci](#).

Untuk informasi selengkapnya tentang perlindungan data, lihat postingan blog [model tanggung jawab bersama AWS dan Peraturan Perlindungan Data Umum \(GDPR\)](#) di Blog Keamanan AWS .

## Topik

- [Enkripsi data](#)
- [Manajemen kunci](#)
- [Privasi lalu lintas](#)

## Enkripsi data

Enkripsi adalah bagian penting dari CodeBuild keamanan. Beberapa enkripsi, seperti untuk data dalam perjalanan, disediakan secara default dan tidak mengharuskan Anda untuk melakukan apa pun. Enkripsi lain, seperti untuk data saat istirahat, Anda dapat mengonfigurasi saat membuat proyek atau membangun.

- Enkripsi data saat istirahat - Artefak build, seperti cache, log, file data laporan pengujian mentah yang diekspor, dan hasil build, dienkripsi secara default menggunakan Kunci yang dikelola AWS. Jika Anda tidak ingin menggunakan kunci KMS ini, Anda harus membuat dan mengkonfigurasi kunci yang dikelola pelanggan. Untuk informasi lebih lanjut [Membuat Kunci KMS](#) dan [konsep Layanan Manajemen AWS Kunci](#) di Panduan AWS Key Management Service Pengguna.
- Anda dapat menyimpan pengenal kunci AWS KMS yang CodeBuild digunakan untuk mengenkripsi artefak keluaran build dalam variabel lingkungan. `CODEBUILD_KMS_KEY_ID` Lihat informasi yang lebih lengkap di [Variabel lingkungan di lingkungan build](#)
- Anda dapat menentukan kunci terkelola pelanggan saat membuat proyek build. Untuk informasi selengkapnya, lihat [Set the Encryption Key Using the Console](#) dan [Mengatur kunci enkripsi menggunakan CLI](#).

Volume Amazon Elastic Block Store dari armada build Anda dienkripsi secara default menggunakan Kunci yang dikelola AWS

- Enkripsi data dalam transit - Semua komunikasi antara pelanggan dan dan antara CodeBuild CodeBuild dan dependensi hilirnya dilindungi menggunakan koneksi TLS yang ditandatangani menggunakan proses penandatanganan Signature Version 4. Semua CodeBuild titik akhir menggunakan sertifikat SHA-256 yang dikelola oleh AWS Private Certificate Authority Untuk

informasi selengkapnya, lihat [Proses penandatanganan Signature Version 4](#) dan [Tentang ACM PCA](#).

- Enkripsi artefak build - Peran CodeBuild layanan yang terkait dengan proyek build memerlukan akses ke kunci KMS untuk mengenkripsi artefak keluaran build-nya. Secara default, CodeBuild gunakan Kunci yang dikelola AWS untuk Amazon S3 di akun Anda AWS . Jika Anda tidak ingin menggunakan ini Kunci yang dikelola AWS, Anda harus membuat dan mengkonfigurasi kunci yang dikelola pelanggan. Untuk informasi selengkapnya, lihat [Membuat kunci dikelola pelanggan](#) dan [Membuat kunci](#) di Panduan AWS KMS Pengembang.

## Manajemen kunci

Anda dapat melindungi konten Anda dari penggunaan yang tidak sah melalui enkripsi. Simpan kunci enkripsi Anda AWS Secrets Manager, lalu berikan izin peran CodeBuild layanan yang terkait dengan proyek build untuk mendapatkan kunci enkripsi dari akun Secrets Manager Anda. Untuk informasi lebih lanjut, lihat [Membuat dan mengkonfigurasi kunci yang dikelola pelanggan CodeBuild](#), [Membuat proyek pembangunan di AWS CodeBuild](#), [Jalankan build di AWS CodeBuild](#), dan [Tutorial: Menyimpan dan mengambil rahasia](#).

Gunakan variabel `CODEBUILD_KMS_KEY_ID` lingkungan dalam perintah build untuk mendapatkan pengenal AWS KMS kunci. Untuk informasi selengkapnya, lihat [Variabel lingkungan di lingkungan build](#).

Anda dapat menggunakan Secrets Manager untuk melindungi kredensial ke registri pribadi yang menyimpan image Docker yang digunakan untuk lingkungan runtime Anda. Untuk informasi selengkapnya, lihat [Registri pribadi dengan AWS Secrets Manager sampel untuk CodeBuild](#).

## Privasi lalu lintas

Anda dapat meningkatkan keamanan build Anda dengan mengonfigurasi CodeBuild untuk menggunakan titik akhir VPC antarmuka. Untuk melakukan ini, Anda tidak memerlukan gateway internet, perangkat NAT, atau gateway pribadi virtual. Hal ini juga tidak diperlukan untuk mengkonfigurasi PrivateLink, meskipun dianjurkan. Untuk informasi selengkapnya, lihat [Gunakan titik akhir VPC](#). Untuk informasi selengkapnya tentang PrivateLink dan titik akhir VPC, lihat [AWS PrivateLink](#) dan [Mengakses AWS](#) layanan melalui PrivateLink

# Manajemen identitas dan akses di AWS CodeBuild

Akses ke AWS CodeBuild membutuhkan kredensial. Kredensial tersebut harus memiliki izin untuk mengakses AWS sumber daya, seperti menyimpan dan mengambil artefak build di bucket S3 dan melihat Log Amazon untuk build. CloudWatch Bagian berikut menjelaskan bagaimana Anda dapat menggunakan [AWS Identity and Access Management](#) (IAM) dan CodeBuild untuk membantu mengamankan akses ke sumber daya Anda:

## Ikhtisar mengelola izin akses ke sumber daya Anda AWS CodeBuild

Setiap AWS sumber daya dimiliki oleh AWS akun, dan izin untuk membuat atau mengakses sumber daya diatur oleh kebijakan izin. Administrator akun dapat melampirkan kebijakan izin ke identitas IAM (yaitu, pengguna, grup, dan peran).

### Note

Administrator akun (atau pengguna administrator) adalah pengguna dengan hak akses administrator. Untuk informasi selengkapnya tentang administrator, lihat [Praktik Terbaik IAM](#) dalam Panduan Pengguna IAM.

Ketika memberikan izin, Anda memutuskan orang yang mendapatkan izin, sumber daya yang dapat diakses, dan tindakan yang dapat dilakukan di sumber daya tersebut.


### Topik

- [AWS CodeBuild sumber daya dan operasi](#)
- [Memahami kepemilikan sumber daya](#)
- [Mengelola akses ke sumber daya](#)
- [Menentukan elemen kebijakan: Tindakan, efek, dan penanggung jawab](#)

## AWS CodeBuild sumber daya dan operasi

Pada tahun AWS CodeBuild, sumber daya utama adalah proyek pembangunan. Dalam sebuah kebijakan, Anda menggunakan Amazon Resource Name (ARN) untuk mengidentifikasi sumber daya tempat kebijakan berlaku. Build juga merupakan sumber daya dan memiliki ARN yang terkait dengannya. Untuk informasi selengkapnya, lihat [Nama Sumber Daya Amazon \(ARN\) dan Ruang Nama AWS Layanan](#) di Referensi Umum Amazon Web Services

| Jenis sumber daya                                                                                  | Format ARN                                                                                                 |
|----------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------|
| Membangun proyek                                                                                   | <code>arn:aws:codebuild: <i>region-ID</i> :<i>account-ID</i> :project/<i>project-name</i></code>           |
| Membangun                                                                                          | <code>arn:aws:codebuild: <i>region-ID</i> :<i>account-ID</i> :build/<i>build-ID</i></code>                 |
| Kelompok laporan                                                                                   | <code>arn:aws:codebuild: <i>region-ID</i> :<i>account-ID</i> :report-group/<i>report-group-name</i></code> |
| Laporan                                                                                            | <code>arn:aws:codebuild: <i>region-ID</i> :<i>account-ID</i> :report/<i>report-ID</i></code>               |
| Semua CodeBuild sumber daya                                                                        | <code>arn:aws:codebuild:*</code>                                                                           |
| Semua CodeBuild sumber daya yang dimiliki oleh akun yang ditentukan di AWS Wilayah yang ditentukan | <code>arn:aws:codebuild: <i>region-ID</i> :<i>account-ID</i> :*</code>                                     |

 Note

Sebagian besar AWS layanan memperlakukan titik dua (:) atau garis miring (/) sebagai karakter yang sama di ARN. Namun, CodeBuild menggunakan kecocokan yang tepat dalam pola dan aturan sumber daya. Pastikan untuk menggunakan karakter yang benar saat membuat pola peristiwa sehingga cocok dengan sintaks ARN di sumber daya.

Misalnya, Anda dapat menunjukkan proyek build (*myBuildProject*) tertentu dalam pernyataan Anda menggunakan ARN sebagai berikut:

```
"Resource": "arn:aws:codebuild:us-east-2:123456789012:project/myBuildProject"
```

Untuk menentukan semua sumber daya, atau jika tindakan API tidak mendukung ARN, gunakan karakter wildcard (\*) dalam Resource elemen sebagai berikut:

```
"Resource": "*"
```

Beberapa tindakan CodeBuild API menerima beberapa sumber daya (misalnya, `BatchGetProjects`). Untuk menentukan beberapa sumber daya dalam satu pernyataan, pisahkan ARN mereka dengan koma, sebagai berikut:

```
"Resource": [
 "arn:aws:codebuild:us-east-2:123456789012:project/myBuildProject",
 "arn:aws:codebuild:us-east-2:123456789012:project/myOtherBuildProject"
]
```

CodeBuild menyediakan satu set operasi untuk bekerja dengan CodeBuild sumber daya. Untuk daftar, lihat [AWS CodeBuild referensi izin](#).

## Memahami kepemilikan sumber daya

AWS Akun memiliki sumber daya yang dibuat di akun, terlepas dari siapa yang membuat sumber daya. Secara khusus, pemilik sumber daya adalah AWS akun [entitas utama](#) (yaitu, akun root, pengguna, atau peran IAM) yang mengautentikasi permintaan pembuatan sumber daya. Contoh berikut menggambarkan cara kerjanya:

- Jika Anda menggunakan kredensial akun root AWS akun Anda untuk membuat aturan, AWS akun Anda adalah pemilik sumber daya. CodeBuild
- Jika Anda membuat pengguna di AWS akun Anda dan memberikan izin untuk membuat CodeBuild sumber daya kepada pengguna tersebut, pengguna dapat membuat CodeBuild sumber daya. Namun, AWS akun Anda, tempat pengguna berada, memiliki CodeBuild sumber daya.
- Jika Anda membuat peran IAM di AWS akun Anda dengan izin untuk membuat CodeBuild sumber daya, siapa pun yang dapat mengambil peran tersebut dapat membuat CodeBuild sumber daya. AWS Akun Anda, tempat peran itu berada, memiliki CodeBuild sumber daya.

## Mengelola akses ke sumber daya

Kebijakan izin menjelaskan siapa yang memiliki akses ke sumber daya mana.

**Note**

Bagian ini membahas penggunaan IAM di AWS CodeBuild. Bagian ini tidak memberikan informasi yang mendetail tentang layanan IAM. Untuk dokumentasi IAM lengkap, lihat [Apa yang Dimaksud dengan IAM?](#) dalam Panduan Pengguna IAM. Untuk informasi tentang sintaksis dan deskripsi kebijakan IAM, lihat [Referensi Kebijakan IAM AWS](#) dalam Panduan Pengguna IAM.

Kebijakan yang terlampir pada identitas IAM disebut sebagai kebijakan berbasis identitas (kebijakan IAM). Kebijakan yang melekat pada sumber daya disebut sebagai kebijakan berbasis sumber daya. CodeBuild mendukung kebijakan berbasis identitas, dan kebijakan berbasis sumber daya untuk API baca saja tertentu untuk tujuan berbagi sumber daya lintas akun.

### Akses aman ke bucket S3

Kami sangat menyarankan agar Anda menyertakan izin berikut dalam peran IAM Anda untuk memverifikasi bucket S3 yang terkait dengan CodeBuild proyek Anda dimiliki oleh Anda atau seseorang yang Anda percayai. Izin ini tidak disertakan dalam kebijakan dan peran AWS terkelola. Anda harus menambahkannya sendiri.

- `s3:GetBucketAc1`
- `s3:GetBucketLocation`

Jika pemilik bucket S3 yang digunakan oleh project Anda berubah, Anda harus memverifikasi bahwa Anda masih memiliki bucket dan memperbarui izin dalam peran IAM Anda jika tidak. Lihat informasi yang lebih lengkap di [Menambahkan izin CodeBuild akses ke grup atau pengguna IAM](#) dan [Membuat peran CodeBuild layanan](#).

### Menentukan elemen kebijakan: Tindakan, efek, dan penanggung jawab

Untuk setiap AWS CodeBuild sumber daya, layanan mendefinisikan satu set operasi API. Untuk memberikan izin untuk operasi API ini, CodeBuild tentukan serangkaian tindakan yang dapat Anda tentukan dalam kebijakan. Beberapa operasi API dapat memerlukan izin untuk lebih dari satu tindakan untuk melakukan operasi API. Lihat informasi yang lebih lengkap di [AWS CodeBuild sumber daya dan operasi](#) dan [AWS CodeBuild referensi izin](#).

Berikut ini adalah elemen-elemen kebijakan dasar:



- Sumber daya – Anda menggunakan Amazon Resource Name (ARN) untuk mengidentifikasi sumber daya yang diberlakukan oleh kebijakan tersebut.
- Tindakan – Anda menggunakan kata kunci tindakan untuk mengidentifikasi operasi sumber daya yang ingin Anda izinkan atau tolak. Misalnya, izin `codebuild:CreateProject` memungkinkan pengguna melakukan operasi `CreateProject`.
- Pengaruh – Anda menetapkan pengaruh, baik mengizinkan atau menolak, ketika pengguna meminta tindakan tertentu. Jika Anda tidak secara eksplisit memberikan akses ke (mengizinkan) sumber daya, akses akan ditolak secara implisit. Anda juga dapat secara eksplisit menolak akses ke sumber daya. Anda dapat melakukan ini untuk memastikan pengguna tidak dapat mengakses sumber daya, meskipun kebijakan lain memberikan akses.
- Prinsipal – Dalam kebijakan berbasis identitas (kebijakan IAM), pengguna yang kebijakannya terlampir adalah prinsipal implisit. Untuk kebijakan berbasis sumber daya, Anda menentukan pengguna, akun, layanan, atau entitas lain yang ingin Anda terima izinnya.

Untuk mempelajari selengkapnya tentang sintaksis dan deskripsi kebijakan IAM, lihat [Referensi Kebijakan IAM AWS](#) dalam Panduan Pengguna IAM.

Untuk tabel yang menunjukkan semua tindakan CodeBuild API dan sumber daya yang diterapkan, lihat [AWS CodeBuild referensi izin](#).

## Menggunakan kebijakan berbasis identitas untuk AWS CodeBuild

Topik ini memberikan contoh kebijakan berbasis identitas yang menunjukkan bagaimana administrator akun dapat melampirkan kebijakan izin ke identitas IAM (yaitu, pengguna, grup, dan peran) dan dengan demikian memberikan izin untuk melakukan operasi AWS CodeBuild sumber daya.

### Important

Kami menyarankan Anda terlebih dahulu meninjau topik pengantar yang menjelaskan konsep dasar dan opsi yang tersedia untuk mengelola akses ke CodeBuild sumber daya Anda. Untuk informasi selengkapnya, lihat [Ikhtisar mengelola izin akses ke sumber daya Anda AWS CodeBuild](#).

## Topik

- [Izin yang diperlukan untuk menggunakan konsol AWS CodeBuild](#)
- [Izin diperlukan untuk terhubung AWS CodeBuild ke Amazon Elastic Container Registry](#)
- [Izin yang diperlukan untuk AWS CodeBuild konsol untuk terhubung ke penyedia sumber](#)
- [AWS kebijakan terkelola \(standar\) untuk AWS CodeBuild](#)
- [CodeBuild kebijakan dan pemberitahuan terkelola](#)
- [CodeBuild pembaruan kebijakan AWS terkelola](#)
- [Contoh kebijakan yang dikelola pelanggan](#)

Berikut ini menunjukkan contoh kebijakan izin yang memungkinkan pengguna mendapatkan informasi tentang proyek build hanya di us-east-2 wilayah 123456789012 untuk akun proyek build apa pun yang dimulai dengan nama: my

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": "codebuild:BatchGetProjects",
 "Resource": "arn:aws:codebuild:us-east-2:123456789012:project/my*"
 }
]
}
```

## Izin yang diperlukan untuk menggunakan konsol AWS CodeBuild

Pengguna yang menggunakan AWS CodeBuild konsol harus memiliki seperangkat izin minimum yang memungkinkan pengguna menjelaskan AWS sumber daya lain untuk AWS akun tersebut. Anda harus memiliki izin dari layanan berikut:

- AWS CodeBuild
- Amazon CloudWatch
- CodeCommit (jika Anda menyimpan kode sumber Anda di AWS CodeCommit repositori)
- Amazon Elastic Container Registry (Amazon ECR) (jika Anda menggunakan lingkungan build yang mengandalkan image Docker di repositori Amazon ECR)

**Note**

Per 26 Juli 2022, kebijakan IAM default telah diperbarui. Untuk informasi selengkapnya, lihat [Izin diperlukan untuk terhubung AWS CodeBuild ke Amazon Elastic Container Registry](#).

- Amazon Elastic Container Service (Amazon ECS) (jika Anda menggunakan lingkungan build yang mengandalkan image Docker di repositori Amazon ECR)
- AWS Identity and Access Management (IAM)
- AWS Key Management Service (AWS KMS)
- Amazon Simple Storage Service (Amazon S3)

Jika Anda membuat kebijakan IAM yang lebih ketat dari izin minimum yang diperlukan, konsol tidak akan berfungsi sebagaimana dimaksudkan.

## Izin diperlukan untuk terhubung AWS CodeBuild ke Amazon Elastic Container Registry

Per 26 Juli 2022, AWS CodeBuild telah memperbarui kebijakan IAM default untuk izin Amazon ECR. Izin berikut telah dihapus dari kebijakan default:

```
"ecr:PutImage",
"ecr:InitiateLayerUpload",
"ecr:UploadLayerPart",
"ecr:CompleteLayerUpload"
```

Untuk CodeBuild proyek yang dibuat sebelum 26 Juli 2022, kami sarankan Anda memperbarui kebijakan Anda dengan kebijakan ECR Amazon berikut:

```
"Action": [
 "ecr:BatchCheckLayerAvailability",
 "ecr:GetDownloadUrlForLayer",
 "ecr:BatchGetImage"
]
```

Untuk informasi selengkapnya tentang memperbarui kebijakan Anda, lihat [Menambahkan izin CodeBuild akses ke grup atau pengguna IAM](#).

## Izin yang diperlukan untuk AWS CodeBuild konsol untuk terhubung ke penyedia sumber

AWS CodeBuild Konsol menggunakan tindakan API berikut untuk terhubung ke penyedia sumber (misalnya, GitHub repositori).

- `codebuild:ListConnectedOAuthAccounts`
- `codebuild:ListRepositories`
- `codebuild:PersistOAuthToken`
- `codebuild:ImportSourceCredentials`

Anda dapat mengaitkan penyedia sumber (seperti GitHub repositori) dengan proyek build menggunakan konsol. AWS CodeBuild Untuk melakukannya, Anda harus terlebih dahulu menambahkan tindakan API sebelumnya ke kebijakan akses IAM yang terkait dengan pengguna yang Anda gunakan untuk mengakses konsol. AWS CodeBuild

Tindakan `ListConnectedOAuthAccounts`, `ListRepositories`, dan `PersistOAuthToken` API tidak dimaksudkan untuk dipanggil oleh kode Anda. Oleh karena itu, tindakan API ini tidak termasuk dalam AWS CLI dan AWS SDK.

## AWS kebijakan terkelola (standar) untuk AWS CodeBuild

AWS mengatasi banyak kasus penggunaan umum dengan menyediakan kebijakan IAM mandiri yang dibuat dan dikelola oleh AWS. Kebijakan AWS terkelola ini memberikan izin yang diperlukan untuk kasus penggunaan umum sehingga Anda dapat menghindari keharusan menyelidiki izin apa yang diperlukan. Kebijakan terkelola CodeBuild juga memberikan izin untuk melakukan operasi di layanan lain, seperti IAM, Amazon EC2, AWS CodeCommit, Amazon ECR, Amazon SNS, CloudWatch dan Amazon Events, sebagaimana diperlukan untuk tanggung jawab bagi pengguna yang telah diberikan kebijakan tersebut. Misalnya, `AWSCodeBuildAdminAccess` kebijakan tersebut adalah kebijakan pengguna tingkat administratif yang memungkinkan pengguna dengan kebijakan ini membuat dan mengelola aturan CloudWatch Acara untuk pembuatan proyek dan topik Amazon SNS untuk pemberitahuan tentang peristiwa terkait proyek (topik yang namanya diawali `arn:aws:codebuild:`), serta mengelola proyek dan grup laporan di CodeBuild Untuk informasi selengkapnya, lihat [Kebijakan Terkelola AWS](#) dalam Panduan Pengguna IAM.

Kebijakan AWS terkelola berikut, yang dapat Anda lampirkan ke pengguna di akun Anda, khusus untuk AWS CodeBuild.

## AWSCodeBuildAdminAccess

Menyediakan akses penuh untuk CodeBuild menyertakan izin untuk mengelola proyek CodeBuild pembangunan.

## AWSCodeBuildDeveloperAccess

Menyediakan akses ke CodeBuild tetapi tidak mengizinkan administrasi proyek build.

## AWSCodeBuildReadOnlyAccess

Menyediakan akses hanya-baca ke. CodeBuild

Untuk mengakses artefak keluaran build yang CodeBuild dibuat, Anda juga harus melampirkan kebijakan AWS terkelola bernama `AmazonS3ReadOnlyAccess`.

Untuk membuat dan mengelola peran CodeBuild layanan, Anda juga harus melampirkan kebijakan AWS terkelola bernama `IAMFullAccess`.

Anda juga dapat membuat kebijakan IAM khusus Anda sendiri untuk memberikan izin untuk CodeBuild tindakan dan sumber daya. Anda dapat menyematkan kebijakan khusus ini untuk pengguna atau grup yang memerlukan izin tersebut.

### Topik

- [AWSCodeBuildAdminAccess](#)
- [AWSCodeBuildDeveloperAccess](#)
- [AWSCodeBuildReadOnlyAccess](#)

## AWSCodeBuildAdminAccess

`AWSCodeBuildAdminAccess` Kebijakan ini menyediakan akses penuh ke CodeBuild, termasuk izin untuk mengelola proyek CodeBuild build. Terapkan kebijakan ini hanya kepada pengguna tingkat administratif untuk memberi mereka kontrol penuh atas CodeBuild proyek, grup laporan, dan sumber daya terkait di AWS akun Anda, termasuk kemampuan untuk menghapus proyek dan grup laporan.

`AWSCodeBuildAdminAccess` Kebijakan tersebut berisi pernyataan kebijakan berikut:

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
```

```

 "Action": [
 "codebuild:*",
 "codecommit:GetBranch",
 "codecommit:GetCommit",
 "codecommit:GetRepository",
 "codecommit:ListBranches",
 "codecommit:ListRepositories",
 "cloudwatch:GetMetricStatistics",
 "ec2:DescribeVpcs",
 "ec2:DescribeSecurityGroups",
 "ec2:DescribeSubnets",
 "ecr:DescribeRepositories",
 "ecr:ListImages",
 "elasticfilesystem:DescribeFileSystems",
 "events:DeleteRule",
 "events:DescribeRule",
 "events:DisableRule",
 "events:EnableRule",
 "events:ListTargetsByRule",
 "events:ListRuleNamesByTarget",
 "events:PutRule",
 "events:PutTargets",
 "events:RemoveTargets",
 "logs:GetLogEvents",
 "s3:GetBucketLocation",
 "s3:ListAllMyBuckets"
],
 "Effect": "Allow",
 "Resource": "*"
 },
 {
 "Action": [
 "logs>DeleteLogGroup"
],
 "Effect": "Allow",
 "Resource": "arn:aws:logs:*:*:log-group:/aws/codebuild/*:log-stream:*"
 },
 {
 "Effect": "Allow",
 "Action": [
 "ssm:PutParameter"
],
 "Resource": "arn:aws:ssm:*:*:parameter/CodeBuild/*"
 },
}

```

```
{
 "Effect": "Allow",
 "Action": [
 "ssm:StartSession"
],
 "Resource": "arn:aws:ecs:*:*:task/*/*"
},
{
 "Sid": "CodeStarConnectionsReadWriteAccess",
 "Effect": "Allow",
 "Action": [
 "codestar-connections:CreateConnection",
 "codestar-connections>DeleteConnection",
 "codestar-connections:UpdateConnectionInstallation",
 "codestar-connections:TagResource",
 "codestar-connections:UntagResource",
 "codestar-connections:ListConnections",
 "codestar-connections:ListInstallationTargets",
 "codestar-connections:ListTagsForResource",
 "codestar-connections:GetConnection",
 "codestar-connections:GetIndividualAccessToken",
 "codestar-connections:GetInstallationUrl",
 "codestar-connections:PassConnection",
 "codestar-connections:StartOAuthHandshake",
 "codestar-connections:UseConnection"
],
 "Resource": "arn:aws:codestar-connections:*:*:connection/*"
},
{
 "Sid": "CodeStarNotificationsReadWriteAccess",
 "Effect": "Allow",
 "Action": [
 "codestar-notifications:CreateNotificationRule",
 "codestar-notifications:DescribeNotificationRule",
 "codestar-notifications:UpdateNotificationRule",
 "codestar-notifications>DeleteNotificationRule",
 "codestar-notifications:Subscribe",
 "codestar-notifications:Unsubscribe"
],
 "Resource": "*",
 "Condition": {
 "StringLike": {
 "codestar-notifications:NotificationsForResource": "arn:aws:codebuild:*"
 }
 }
}
```

```
 }
 },
 {
 "Sid": "CodeStarNotificationsListAccess",
 "Effect": "Allow",
 "Action": [
 "codestar-notifications:ListNotificationRules",
 "codestar-notifications:ListEventTypes",
 "codestar-notifications:ListTargets",
 "codestar-notifications:ListTagsForResource"
],
 "Resource": "*"
 },
 {
 "Sid": "CodeStarNotificationsSNSTopicCreateAccess",
 "Effect": "Allow",
 "Action": [
 "sns:CreateTopic",
 "sns:SetTopicAttributes"
],
 "Resource": "arn:aws:sns:*:*:codestar-notifications*"
 },
 {
 "Sid": "SNSTopicListAccess",
 "Effect": "Allow",
 "Action": [
 "sns:ListTopics",
 "sns:GetTopicAttributes"
],
 "Resource": "*"
 },
 {
 "Sid": "CodeStarNotificationsChatbotAccess",
 "Effect": "Allow",
 "Action": [
 "chatbot:DescribeSlackChannelConfigurations",
 "chatbot:ListMicrosoftTeamsChannelConfigurations"
],
 "Resource": "*"
 }
]
}
```



## AWSCodeBuildDeveloperAccess

`AWSCodeBuildDeveloperAccess` kebijakan ini memungkinkan akses ke semua fungsionalitas CodeBuild dan memproyeksikan dan melaporkan sumber daya terkait grup. Kebijakan ini tidak mengizinkan pengguna untuk menghapus CodeBuild proyek atau grup laporan, atau sumber daya terkait di AWS layanan lain, seperti CloudWatch Acara. Kami merekomendasikan bahwa Anda menerapkan kebijakan ini untuk sebagian besar pengguna.

`AWSCodeBuildDeveloperAccess` kebijakan tersebut berisi pernyataan kebijakan berikut:

```
{
 "Statement": [
 {
 "Action": [
 "codebuild:StartBuild",
 "codebuild:StopBuild",
 "codebuild:StartBuildBatch",
 "codebuild:StopBuildBatch",
 "codebuild:RetryBuild",
 "codebuild:RetryBuildBatch",
 "codebuild:BatchGet*",
 "codebuild:GetResourcePolicy",
 "codebuild:DescribeTestCases",
 "codebuild:DescribeCodeCoverages",
 "codebuild:List*",
 "codecommit:GetBranch",
 "codecommit:GetCommit",
 "codecommit:GetRepository",
 "codecommit:ListBranches",
 "cloudwatch:GetMetricStatistics",
 "events:DescribeRule",
 "events:ListTargetsByRule",
 "events:ListRuleNamesByTarget",
 "logs:GetLogEvents",
 "s3:GetBucketLocation",
 "s3:ListAllMyBuckets"
],
 "Effect": "Allow",
 "Resource": "*"
 },
 {
 "Effect": "Allow",
 "Action": [
```

```

 "ssm:PutParameter"
],
 "Resource": "arn:aws:ssm:*:*:parameter/CodeBuild/*"
},
{
 "Effect": "Allow",
 "Action": [
 "ssm:StartSession"
],
 "Resource": "arn:aws:ecs:*:*:task/*/*"
},
{
 "Sid": "CodeStarConnectionsUserAccess",
 "Effect": "Allow",
 "Action": [
 "codestar-connections:ListConnections",
 "codestar-connections:GetConnection"
],
 "Resource": "arn:aws:codestar-connections:*:*:connection/*"
},
{
 "Sid": "CodeStarNotificationsReadWriteAccess",
 "Effect": "Allow",
 "Action": [
 "codestar-notifications:CreateNotificationRule",
 "codestar-notifications:DescribeNotificationRule",
 "codestar-notifications:UpdateNotificationRule",
 "codestar-notifications:Subscribe",
 "codestar-notifications:Unsubscribe"
],
 "Resource": "*",
 "Condition": {
 "StringLike": {
 "codestar-notifications:NotificationsForResource": "arn:aws:codebuild:*"
 }
 }
},
{
 "Sid": "CodeStarNotificationsListAccess",
 "Effect": "Allow",
 "Action": [
 "codestar-notifications:ListNotificationRules",
 "codestar-notifications:ListEventTypes",
 "codestar-notifications:ListTargets",

```

```

 "codestar-notifications:ListTagsForResource"
],
 "Resource": "*"
},
{
 "Sid": "SNSTopicListAccess",
 "Effect": "Allow",
 "Action": [
 "sns:ListTopics",
 "sns:GetTopicAttributes"
],
 "Resource": "*"
},
{
 "Sid": "CodeStarNotificationsChatbotAccess",
 "Effect": "Allow",
 "Action": [
 "chatbot:DescribeSlackChannelConfigurations",
 "chatbot:ListMicrosoftTeamsChannelConfigurations"
],
 "Resource": "*"
}
],
"Version": "2012-10-17"
}

```

## AWSCodeBuildReadOnlyAccess

AWSCodeBuildReadOnlyAccessKebijakan ini memberikan akses hanya-baca ke CodeBuild dan sumber daya terkait di layanan lain. AWS Terapkan kebijakan ini kepada pengguna yang dapat melihat dan menjalankan build, melihat proyek, dan melihat grup laporan, tetapi tidak dapat membuat perubahan apa pun padanya.

AWSCodeBuildReadOnlyAccessKebijakan tersebut berisi pernyataan kebijakan berikut:

```

{
 "Statement": [
 {
 "Action": [
 "codebuild:BatchGet*",
 "codebuild:GetResourcePolicy",
 "codebuild:List*",
 "codebuild:DescribeTestCases",

```

```

 "codebuild:DescribeCodeCoverages",
 "codecommit:GetBranch",
 "codecommit:GetCommit",
 "codecommit:GetRepository",
 "cloudwatch:GetMetricStatistics",
 "events:DescribeRule",
 "events:ListTargetsByRule",
 "events:ListRuleNamesByTarget",
 "logs:GetLogEvents"
],
 "Effect": "Allow",
 "Resource": "*"
},
{
 "Sid": "CodeStarConnectionsUserAccess",
 "Effect": "Allow",
 "Action": [
 "codestar-connections:ListConnections",
 "codestar-connections:GetConnection"
],
 "Resource": "arn:aws:codestar-connections:*:*:connection/*"
},
{
 "Sid": "CodeStarNotificationsPowerUserAccess",
 "Effect": "Allow",
 "Action": [
 "codestar-notifications:DescribeNotificationRule"
],
 "Resource": "*",
 "Condition": {
 "StringLike": {
 "codestar-notifications:NotificationsForResource": "arn:aws:codebuild:*"
 }
 }
},
{
 "Sid": "CodeStarNotificationsListAccess",
 "Effect": "Allow",
 "Action": [
 "codestar-notifications:ListNotificationRules",
 "codestar-notifications:ListEventTypes",
 "codestar-notifications:ListTargets"
],
 "Resource": "*"
}

```

```

 }
],
 "Version": "2012-10-17"
}

```

## CodeBuild kebijakan dan pemberitahuan terkelola

CodeBuild mendukung pemberitahuan, yang dapat memberi tahu pengguna tentang perubahan penting untuk membangun proyek. Kebijakan terkelola untuk CodeBuild menyertakan pernyataan kebijakan untuk fungsionalitas notifikasi. Untuk informasi selengkapnya, lihat [Apa itu notifikasi?](#).

Izin yang terkait dengan notifikasi dalam kebijakan terkelola akses penuh

Kebijakan terkelola `AWSCodeBuildFullAccess` mencakup pernyataan berikut untuk mengizinkan akses penuh ke notifikasi. Pengguna dengan kebijakan terkelola ini juga dapat membuat dan mengelola topik Amazon SNS untuk notifikasi, berlangganan dan berhenti berlangganan topik, membuat daftar topik untuk dipilih sebagai target untuk aturan notifikasi, dan membuat daftar klien AWS Chatbot yang dikonfigurasi untuk Slack.

```

{
 "Sid": "CodeStarNotificationsReadWriteAccess",
 "Effect": "Allow",
 "Action": [
 "codestar-notifications:CreateNotificationRule",
 "codestar-notifications:DescribeNotificationRule",
 "codestar-notifications:UpdateNotificationRule",
 "codestar-notifications>DeleteNotificationRule",
 "codestar-notifications:Subscribe",
 "codestar-notifications:Unsubscribe"
],
 "Resource": "*",
 "Condition" : {
 "StringLike" : {"codestar-notifications:NotificationsForResource" :
"arn:aws:codebuild:*"}
 }
},
{
 "Sid": "CodeStarNotificationsListAccess",
 "Effect": "Allow",
 "Action": [
 "codestar-notifications:ListNotificationRules",
 "codestar-notifications:ListTargets",

```

```

 "codestar-notifications:ListTagsForResource",
 "codestar-notifications:ListEventTypes"
],
 "Resource": "*"
 },
 {
 "Sid": "CodeStarNotificationsSNSTopicCreateAccess",
 "Effect": "Allow",
 "Action": [
 "sns:CreateTopic",
 "sns:SetTopicAttributes"
],
 "Resource": "arn:aws:sns:*:*:codestar-notifications*"
 },
 {
 "Sid": "SNSTopicListAccess",
 "Effect": "Allow",
 "Action": [
 "sns:ListTopics"
],
 "Resource": "*"
 },
 {
 "Sid": "CodeStarNotificationsChatbotAccess",
 "Effect": "Allow",
 "Action": [
 "chatbot:DescribeSlackChannelConfigurations",
 "chatbot:ListMicrosoftTeamsChannelConfigurations"
],
 "Resource": "*"
 }
}

```

Izin yang terkait dengan notifikasi dalam kebijakan terkelola hanya-baca

Kebijakan terkelola `AWSCodeBuildReadOnlyAccess` mencakup pernyataan berikut untuk mengizinkan akses penuh ke notifikasi. Pengguna yang menerapkan kebijakan terkelola ini dapat melihat pemberitahuan untuk sumber daya, tetapi tidak dapat membuat, mengelola, atau berlangganan ke sumber daya tersebut.

```

{
 "Sid": "CodeStarNotificationsPowerUserAccess",
 "Effect": "Allow",
 "Action": [

```

```

 "codestar-notifications:DescribeNotificationRule"
],
 "Resource": "*",
 "Condition" : {
 "StringLike" : {"codestar-notifications:NotificationsForResource" :
"arn:aws:codebuild:*"}
 }
},
{
 "Sid": "CodeStarNotificationsListAccess",
 "Effect": "Allow",
 "Action": [
 "codestar-notifications:ListNotificationRules",
 "codestar-notifications:ListEventTypes",
 "codestar-notifications:ListTargets"
],
 "Resource": "*"
}

```

Izin terkait notifikasi dalam kebijakan terkelola lainnya

Kebijakan terkelola `AWSCodeBuildDeveloperAccess` mencakup pernyataan berikut untuk mengizinkan pengguna membuat, mengedit, dan berlangganan notifikasi. Mereka tidak dapat menghapus aturan notifikasi atau mengelola tag untuk sumber daya.

```

{
 "Sid": "CodeStarNotificationsReadWriteAccess",
 "Effect": "Allow",
 "Action": [
 "codestar-notifications:CreateNotificationRule",
 "codestar-notifications:DescribeNotificationRule",
 "codestar-notifications:UpdateNotificationRule",
 "codestar-notifications:Subscribe",
 "codestar-notifications:Unsubscribe"
],
 "Resource": "*",
 "Condition" : {
 "StringLike" : {"codestar-notifications:NotificationsForResource" :
"arn:aws:codebuild*"}
 }
},
{
 "Sid": "CodeStarNotificationsListAccess",

```

```
"Effect": "Allow",
"Action": [
 "codestar-notifications:ListNotificationRules",
 "codestar-notifications:ListTargets",
 "codestar-notifications:ListTagsForResource",
 "codestar-notifications:ListEventTypes"
],
"Resource": "*"
},
{
 "Sid": "SNSTopicListAccess",
 "Effect": "Allow",
 "Action": [
 "sns:ListTopics"
],
 "Resource": "*"
},
{
 "Sid": "CodeStarNotificationsChatbotAccess",
 "Effect": "Allow",
 "Action": [
 "chatbot:DescribeSlackChannelConfigurations",
 "chatbot:ListMicrosoftTeamsChannelConfigurations"
],
 "Resource": "*"
}
```

Untuk informasi selengkapnya tentang IAM dan notifikasi, lihat [Identity and Access Management untuk AWS CodeStar Pemberitahuan](#).

## CodeBuild pembaruan kebijakan AWS terkelola

Lihat detail tentang pembaruan kebijakan AWS terkelola CodeBuild sejak layanan ini mulai melacak perubahan ini. Untuk peringatan otomatis tentang perubahan pada halaman ini, berlangganan RSS feed on. [AWS CodeBuild Riwayat dokumen Panduan Pengguna](#)

| Perubahan                             | Deskripsi                                                                    | Tanggal     |
|---------------------------------------|------------------------------------------------------------------------------|-------------|
| AWSCodeBuildAdminAccess dan AWSCodeBu | CodeBuild menambahkan izin ke kebijakan ini untuk mendukung jenis notifikasi | 16 Mei 2023 |



| Perubahan                                                                | Deskripsi                                                                                                                                                                                            | Tanggal      |
|--------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------|
| <a href="#">BuildDeveloperAccess</a><br>— Perbarui ke kebijakan yang ada | tambahan yang digunakan AWS Chatbot.<br><br>AWSCodeBuildDeveloperAccess Kebijakan AWSCodeBuildAdminAccess dan telah diubah untuk menambahkan izin, chatbot:ListMicrosoftTeamsChannelConfigurations . |              |
| CodeBuild mulai melacak perubahan                                        | CodeBuild mulai melacak perubahan untuk kebijakan yang AWS dikelola.                                                                                                                                 | Mei 16, 2021 |

## Contoh kebijakan yang dikelola pelanggan

Di bagian ini, Anda dapat menemukan contoh kebijakan pengguna yang memberikan izin untuk AWS CodeBuild tindakan. Kebijakan ini berfungsi saat Anda menggunakan CodeBuild API, AWS SDK, atau AWS CLI. Saat menggunakan konsol, Anda harus memberikan izin tambahan khusus konsol. Untuk informasi, lihat [Izin yang diperlukan untuk menggunakan konsol AWS CodeBuild](#).

Anda dapat menggunakan contoh kebijakan IAM berikut untuk membatasi CodeBuild akses bagi pengguna dan peran Anda.

### Topik

- [Memungkinkan pengguna untuk mendapatkan informasi tentang proyek build](#)
- [Memungkinkan pengguna untuk mendapatkan informasi tentang grup laporan](#)
- [Memungkinkan pengguna untuk mendapatkan informasi tentang laporan](#)
- [Izinkan pengguna membuat proyek pembangunan](#)
- [Izinkan pengguna membuat grup laporan](#)
- [Izinkan pengguna menghapus grup laporan](#)
- [Memungkinkan pengguna untuk menghapus laporan](#)

- [Izinkan pengguna untuk menghapus proyek build](#)
- [Izinkan pengguna untuk mendapatkan daftar nama proyek build](#)
- [Memungkinkan pengguna untuk mengubah informasi tentang proyek build](#)
- [Izinkan pengguna mengubah grup laporan](#)
- [Memungkinkan pengguna untuk mendapatkan informasi tentang build](#)
- [Izinkan pengguna mendapatkan daftar ID build untuk proyek build](#)
- [Izinkan pengguna mendapatkan daftar ID build](#)
- [Memungkinkan pengguna untuk mendapatkan daftar grup laporan](#)
- [Memungkinkan pengguna untuk mendapatkan daftar laporan](#)
- [Memungkinkan pengguna untuk mendapatkan daftar laporan untuk grup laporan](#)
- [Memungkinkan pengguna untuk mendapatkan daftar kasus uji untuk laporan](#)
- [Izinkan pengguna untuk mulai menjalankan build](#)
- [Izinkan pengguna untuk mencoba menghentikan build](#)
- [Izinkan pengguna mencoba menghapus build](#)
- [Izinkan pengguna untuk mendapatkan informasi tentang gambar Docker yang dikelola oleh CodeBuild](#)
- [Izinkan CodeBuild akses ke AWS layanan yang diperlukan untuk membuat antarmuka jaringan VPC](#)
- [Gunakan pernyataan penolakan untuk mencegah terputusnya sambungan AWS CodeBuild dari penyedia sumber](#)

Memungkinkan pengguna untuk mendapatkan informasi tentang proyek build

Contoh pernyataan kebijakan berikut memungkinkan pengguna untuk mendapatkan informasi tentang proyek build di us-east-2 Wilayah 123456789012 untuk memperhitungkan setiap proyek build yang dimulai dengan namamy:

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": "codebuild:BatchGetProjects",
 "Resource": "arn:aws:codebuild:us-east-2:123456789012:project/my*"
 }
]
}
```

```
 }
]
}
```

Memungkinkan pengguna untuk mendapatkan informasi tentang grup laporan

Contoh pernyataan kebijakan berikut memungkinkan pengguna untuk mendapatkan informasi tentang grup laporan di us-east-2 Wilayah untuk akun123456789012:

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": "codebuild:BatchGetReportGroups",
 "Resource": "arn:aws:codebuild:us-east-2:123456789012:report-group/*"
 }
]
}
```

Memungkinkan pengguna untuk mendapatkan informasi tentang laporan

Contoh pernyataan kebijakan berikut memungkinkan pengguna untuk mendapatkan informasi tentang laporan di us-east-2 Wilayah untuk akun123456789012:

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": "codebuild:BatchGetReports",
 "Resource": "arn:aws:codebuild:us-east-2:123456789012:report-group/*"
 }
]
}
```

Izinkan pengguna membuat proyek pembangunan

Contoh pernyataan kebijakan berikut memungkinkan pengguna untuk membuat proyek build dengan nama apa pun tetapi hanya di us-east-2 Wilayah untuk akun 123456789012 dan hanya menggunakan peran CodeBuild layanan yang ditentukan:

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": "codebuild:CreateProject",
 "Resource": "arn:aws:codebuild:us-east-2:123456789012:project/*"
 },
 {
 "Effect": "Allow",
 "Action": "iam:PassRole",
 "Resource": "arn:aws:iam::123456789012:role/CodeBuildServiceRole"
 }
]
}
```

Pernyataan kebijakan contoh berikut memungkinkan pengguna untuk membuat proyek build dengan nama apa pun tetapi hanya di us-east-2 Wilayah untuk akun 123456789012 dan hanya menggunakan peran CodeBuild layanan yang ditentukan. Ini juga memberlakukan bahwa pengguna hanya dapat menggunakan peran layanan yang ditentukan dengan AWS CodeBuild dan bukan AWS layanan lain.

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": "codebuild:CreateProject",
 "Resource": "arn:aws:codebuild:us-east-2:123456789012:project/*"
 },
 {
 "Effect": "Allow",
 "Action": "iam:PassRole",
 "Resource": "arn:aws:iam::123456789012:role/CodeBuildServiceRole",
 "Condition": {
 "StringEquals": {"iam:PassedToService": "codebuild.amazonaws.com"}
 }
 }
]
}}
```

## Izinkan pengguna membuat grup laporan

Contoh pernyataan kebijakan berikut memungkinkan pengguna untuk membuat grup laporan di us-east-2 Wilayah untuk akun123456789012:

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": "codebuild:CreateReportGroup",
 "Resource": "arn:aws:codebuild:us-east-2:123456789012:report-group/*"
 }
]
}
```

## Izinkan pengguna menghapus grup laporan

Contoh pernyataan kebijakan berikut memungkinkan pengguna untuk menghapus grup laporan di us-east-2 Wilayah untuk akun123456789012:

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": "codebuild>DeleteReportGroup",
 "Resource": "arn:aws:codebuild:us-east-2:123456789012:report-group/*"
 }
]
}
```

## Memungkinkan pengguna untuk menghapus laporan

Contoh pernyataan kebijakan berikut memungkinkan pengguna untuk menghapus laporan di us-east-2 Wilayah untuk akun123456789012:

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
```

```
 "Effect": "Allow",
 "Action": "codebuild:DeleteReport",
 "Resource": "arn:aws:codebuild:us-east-2:123456789012:report-group/*"
 }
]
```

Izinkan pengguna untuk menghapus proyek build

Contoh pernyataan kebijakan berikut memungkinkan pengguna untuk menghapus proyek build di us-east-2 Wilayah untuk akun 123456789012 untuk setiap proyek build yang dimulai dengan namamy:

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": "codebuild:DeleteProject",
 "Resource": "arn:aws:codebuild:us-east-2:123456789012:project/my*"
 }
]
}
```

Izinkan pengguna untuk mendapatkan daftar nama proyek build

Contoh pernyataan kebijakan berikut memungkinkan pengguna untuk mendapatkan daftar nama proyek build untuk akun yang sama:

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": "codebuild:ListProjects",
 "Resource": "*"
 }
]
}
```

## Memungkinkan pengguna untuk mengubah informasi tentang proyek build

Contoh pernyataan kebijakan berikut memungkinkan pengguna untuk mengubah informasi tentang proyek build dengan nama apa pun tetapi hanya di us-east-2 Wilayah untuk akun 123456789012 dan hanya menggunakan peran AWS CodeBuild layanan yang ditentukan:

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": "codebuild:UpdateProject",
 "Resource": "arn:aws:codebuild:us-east-2:123456789012:project/*"
 },
 {
 "Effect": "Allow",
 "Action": "iam:PassRole",
 "Resource": "arn:aws:iam::123456789012:role/CodeBuildServiceRole"
 }
]
}
```

## Izinkan pengguna mengubah grup laporan

Contoh pernyataan kebijakan berikut memungkinkan pengguna untuk mengubah grup laporan di us-east-2 Wilayah untuk akun123456789012:

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": "codebuild:UpdateReportGroup",
 "Resource": "arn:aws:codebuild:us-east-2:123456789012:report-group/*"
 }
]
}
```

## Memungkinkan pengguna untuk mendapatkan informasi tentang build

Contoh pernyataan kebijakan berikut memungkinkan pengguna untuk mendapatkan informasi tentang build di us-east-2 Wilayah 123456789012 untuk akun proyek build bernama my-build-project danmy-other-build-project:

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": "codebuild:BatchGetBuilds",
 "Resource": [
 "arn:aws:codebuild:us-east-2:123456789012:project/my-build-project",
 "arn:aws:codebuild:us-east-2:123456789012:project/my-other-build-project"
]
 }
]
}
```

## Izinkan pengguna mendapatkan daftar ID build untuk proyek build

Contoh pernyataan kebijakan berikut memungkinkan pengguna untuk mendapatkan daftar ID build di us-east-2 Region 123456789012 untuk akun proyek build bernama my-build-project danmy-other-build-project:

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": "codebuild:ListBuildsForProject",
 "Resource": [
 "arn:aws:codebuild:us-east-2:123456789012:project/my-build-project",
 "arn:aws:codebuild:us-east-2:123456789012:project/my-other-build-project"
]
 }
]
}
```



## Izinkan pengguna mendapatkan daftar ID build

Contoh pernyataan kebijakan berikut memungkinkan pengguna mendapatkan daftar semua ID build untuk akun yang sama:

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": "codebuild:ListBuilds",
 "Resource": "*"
 }
]
}
```

## Memungkinkan pengguna untuk mendapatkan daftar grup laporan

Contoh pernyataan kebijakan berikut memungkinkan pengguna untuk mendapatkan daftar grup laporan di us-east-2 Wilayah untuk akun123456789012:

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": "codebuild:ListReportGroups",
 "Resource": "*"
 }
]
}
```

## Memungkinkan pengguna untuk mendapatkan daftar laporan

Contoh pernyataan kebijakan berikut memungkinkan pengguna untuk mendapatkan daftar laporan di us-east-2 Wilayah untuk akun123456789012:

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
```

```
 "Action": "codebuild:ListReports",
 "Resource": "*"
 }
]
```

Memungkinkan pengguna untuk mendapatkan daftar laporan untuk grup laporan

Contoh pernyataan kebijakan berikut memungkinkan pengguna untuk mendapatkan daftar laporan untuk grup laporan di us-east-2 Wilayah untuk akun123456789012:

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": "codebuild:ListReportsForReportGroup",
 "Resource": "arn:aws:codebuild:us-east-2:123456789012:report-group/*"
 }
]
}
```

Memungkinkan pengguna untuk mendapatkan daftar kasus uji untuk laporan

Contoh pernyataan kebijakan berikut memungkinkan pengguna untuk mendapatkan daftar kasus uji untuk laporan di us-east-2 Wilayah untuk akun123456789012:

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": "codebuild:DescribeTestCases",
 "Resource": "arn:aws:codebuild:us-east-2:123456789012:report-group/*"
 }
]
}
```

Izinkan pengguna untuk mulai menjalankan build

Contoh pernyataan kebijakan berikut memungkinkan pengguna menjalankan build di us-east-2 Region 123456789012 untuk akun proyek build yang dimulai dengan nama: my

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": "codebuild:StartBuild",
 "Resource": "arn:aws:codebuild:us-east-2:123456789012:project/my*"
 }
]
}
```

Izinkan pengguna untuk mencoba menghentikan build

Contoh pernyataan kebijakan berikut memungkinkan pengguna untuk mencoba menghentikan menjalankan build hanya di `us-east-2` wilayah `123456789012` untuk memperhitungkan setiap proyek build yang dimulai dengan nama: `my`

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": "codebuild:StopBuild",
 "Resource": "arn:aws:codebuild:us-east-2:123456789012:project/my*"
 }
]
}
```

Izinkan pengguna mencoba menghapus build

Contoh pernyataan kebijakan berikut memungkinkan pengguna untuk mencoba menghapus build hanya di `us-east-2` Region untuk akun `123456789012` untuk setiap proyek build yang dimulai dengan nama: `my`

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": "codebuild:BatchDeleteBuilds",
 "Resource": "arn:aws:codebuild:us-east-2:123456789012:project/my*"
 }
]
}
```

```
 }
]
}
```

Izinkan pengguna untuk mendapatkan informasi tentang gambar Docker yang dikelola oleh CodeBuild

Contoh pernyataan kebijakan berikut memungkinkan pengguna untuk mendapatkan informasi tentang semua gambar Docker yang dikelola oleh CodeBuild:

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": "codebuild:ListCuratedEnvironmentImages",
 "Resource": "*"
 }
]
}
```

Izinkan CodeBuild akses ke AWS layanan yang diperlukan untuk membuat antarmuka jaringan VPC

Contoh pernyataan kebijakan berikut memberikan AWS CodeBuild izin untuk membuat antarmuka jaringan di VPC dengan dua subnet:

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": [
 "ec2:CreateNetworkInterface",
 "ec2:DescribeDhcpOptions",
 "ec2:DescribeNetworkInterfaces",
 "ec2>DeleteNetworkInterface",
 "ec2:DescribeSubnets",
 "ec2:DescribeSecurityGroups",
 "ec2:DescribeVpcs"
],
 "Resource": "*"
 },
],
}
```

```
{
 "Effect": "Allow",
 "Action": [
 "ec2:CreateNetworkInterfacePermission"
],
 "Resource": "arn:aws:ec2:region:account-id:network-interface/*",
 "Condition": {
 "StringEquals": {
 "ec2:AuthorizedService": "codebuild.amazonaws.com"
 },
 "ArnEquals": {
 "ec2:Subnet": [
 "arn:aws:ec2:region:account-id:subnet/subnet-id-1",
 "arn:aws:ec2:region:account-id:subnet/subnet-id-2"
]
 }
 }
}
```

Gunakan pernyataan penolakan untuk mencegah terputusnya sambungan AWS CodeBuild dari penyedia sumber

Contoh pernyataan kebijakan berikut menggunakan pernyataan penolakan untuk mencegah terputus AWS CodeBuild dari penyedia sumber. Ini menggunakan `codebuild:DeleteOAuthToken`, yang merupakan kebalikan dari `codebuild:PersistOAuthToken` dan `codebuild:ImportSourceCredentials`, untuk terhubung dengan penyedia sumber. Untuk informasi selengkapnya, lihat [Izin yang diperlukan untuk AWS CodeBuild konsol untuk terhubung ke penyedia sumber](#).

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Deny",
 "Action": "codebuild:DeleteOAuthToken",
 "Resource": "*"
 }
]
}
```

## AWS CodeBuild referensi izin

Anda dapat menggunakan kunci kondisi AWS-wide dalam AWS CodeBuild kebijakan Anda untuk menyatakan kondisi. Untuk daftar, lihat [Kunci yang Tersedia](#) di Panduan Pengguna IAM.

Anda menentukan tindakan di bidang `Action` kebijakan. Untuk menentukan tindakan, gunakan prefiks `codebuild:` diikuti dengan nama operasi API (misalnya `codebuild:CreateProject` dan `codebuild:StartBuild`). Untuk menetapkan beberapa tindakan dalam satu pernyataan, pisahkan dengan koma (misalnya, `"Action": [ "codebuild:CreateProject", "codebuild:StartBuild" ]`).

### Menggunakan Karakter Wildcard

Anda menentukan ARN, dengan atau tanpa karakter wildcard (\*), sebagai nilai sumber daya dalam bidang `Resource` kebijakan. Anda bisa menggunakan wildcard untuk menentukan beberapa tindakan atau sumber daya. Misalnya, `codebuild:*` menentukan semua CodeBuild tindakan dan `codebuild:Batch*` menentukan semua CodeBuild tindakan yang dimulai dengan kata. `Batch` Contoh berikut memberikan akses ke semua proyek build dengan nama yang dimulai dengan `my`:

```
arn:aws:codebuild:us-east-2:123456789012:project/my*
```

### CodeBuild Operasi API dan izin yang diperlukan untuk tindakan

#### BatchDeleteBuilds

Tindakan: `codebuild:BatchDeleteBuilds`

Diperlukan untuk menghapus build.

Sumber daya: `arn:aws:codebuild:region-ID:account-ID:project/project-name`

#### BatchGetBuilds

Tindakan: `codebuild:BatchGetBuilds`

Diperlukan untuk mendapatkan informasi tentang build.

Sumber daya: `arn:aws:codebuild:region-ID:account-ID:project/project-name`

#### BatchGetProjects

Tindakan: `codebuild:BatchGetProjects`

Diperlukan untuk mendapatkan informasi tentang membangun proyek.

Sumber daya: `arn:aws:codebuild:region-ID:account-ID:project/project-name`

#### BatchGetReportGroups

Tindakan: `codebuild:BatchGetReportGroups`

Diperlukan untuk mendapatkan informasi tentang grup laporan.

Sumber daya: `arn:aws:codebuild:region-ID:account-ID:report-group/report-group-name`

#### BatchGetReports

Tindakan: `codebuild:BatchGetReports`

Diperlukan untuk mendapatkan informasi tentang laporan.

Sumber daya: `arn:aws:codebuild:region-ID:account-ID:report-group/report-group-name`

#### BatchPutTestCases<sup>1</sup>

Tindakan: `codebuild:BatchPutTestCases`

Diperlukan untuk membuat atau memperbarui laporan pengujian.

Sumber daya: `arn:aws:codebuild:region-ID:account-ID:report-group/report-group-name`

#### CreateProject

Tindakan: `codebuild>CreateProject`, `iam:PassRole`

Diperlukan untuk membuat proyek pembangunan.

Sumber Daya:

- `arn:aws:codebuild:region-ID:account-ID:project/project-name`
- `arn:aws:iam::account-ID:role/role-name`

#### CreateReport<sup>1</sup>

Tindakan: `codebuild>CreateReport`

Diperlukan untuk membuat laporan pengujian.

Sumber daya: `arn:aws:codebuild:region-ID:account-ID:report-group/report-group-name`

### CreateReportGroup

Tindakan: `codebuild:CreateReportGroup`

Diperlukan untuk membuat grup laporan.

Sumber daya: `arn:aws:codebuild:region-ID:account-ID:report-group/report-group-name`

### CreateWebhook

Tindakan: `codebuild:CreateWebhook`

Diperlukan untuk membuat webhook.

Sumber daya: `arn:aws:codebuild:region-ID:account-ID:project/project-name`

### DeleteProject

Tindakan: `codebuild>DeleteProject`

Diperlukan untuk menghapus CodeBuild proyek.

Sumber daya: `arn:aws:codebuild:region-ID:account-ID:project/project-name`

### DeleteReport

Tindakan: `codebuild>DeleteReport`

Diperlukan untuk menghapus laporan.

Sumber daya: `arn:aws:codebuild:region-ID:account-ID:report-group/report-group-name`

### DeleteReportGroup

Tindakan: `codebuild>DeleteReportGroup`

Diperlukan untuk menghapus grup laporan.



Sumber daya: `arn:aws:codebuild:region-ID:account-ID:report-group/report-group-name`

### DeleteSourceCredentials

Tindakan: `codebuild>DeleteSourceCredentials`

Diperlukan untuk menghapus sekumpulan `SourceCredentialsInfo` objek yang berisi informasi tentang kredensial untuk repositori GitHub, Server GitHub Perusahaan, atau Bitbucket.

Sumber daya: \*

### DeleteWebhook

Tindakan: `codebuild>DeleteWebhook`

Diperlukan untuk membuat webhook.

Sumber daya: `arn:aws:codebuild:region-ID:account-ID:project/project-name`

### DescribeTestCases

Tindakan: `codebuild:DescribeTestCases`

Diperlukan untuk mengembalikan daftar kasus uji paginasi.

Sumber daya: `arn:aws:codebuild:region-ID:account-ID:report-group/report-group-name`

### ImportSourceCredentials

Tindakan: `codebuild:ImportSourceCredentials`

Diperlukan untuk mengimpor sekumpulan `SourceCredentialsInfo` objek yang berisi informasi tentang kredensial untuk repositori GitHub, Server GitHub Perusahaan, atau Bitbucket.

Sumber daya: \*

### InvalidateProjectCache

Tindakan: `codebuild:InvalidateProjectCache`

Diperlukan untuk mengatur ulang cache untuk sebuah proyek.

Sumber daya: `arn:aws:codebuild:region-ID:account-ID:project/project-name`

## ListBuildBatches

Tindakan: `codebuild:ListBuildBatches`

Diperlukan untuk mendapatkan daftar ID batch build.

Sumber daya: \*

## ListBuildBatchesForProject

Tindakan: `codebuild:ListBuildBatchesForProject`

Diperlukan untuk mendapatkan daftar ID batch build untuk proyek tertentu.

Sumber daya: `arn:aws:codebuild:region-ID:account-ID:project/project-name`

## ListBuilds

Tindakan: `codebuild:ListBuilds`

Diperlukan untuk mendapatkan daftar ID build.

Sumber daya: \*

## ListBuildsForProject

Tindakan: `codebuild:ListBuildsForProject`

Diperlukan untuk mendapatkan daftar ID build untuk proyek build.

Sumber daya: `arn:aws:codebuild:region-ID:account-ID:project/project-name`

## ListCuratedEnvironmentImages

Tindakan: `codebuild:ListCuratedEnvironmentImages`

Diperlukan untuk mendapatkan informasi tentang semua gambar Docker yang dikelola oleh AWS CodeBuild.

Sumber daya: \* (diperlukan, tetapi tidak mengacu pada sumber daya yang dapat dialamatkan AWS )

## ListProjects

Tindakan: `codebuild:ListProjects`

Diperlukan untuk mendapatkan daftar nama proyek build.

Sumber daya: \*

### ListReportGroups

Tindakan: `codebuild:ListReportGroups`

Diperlukan untuk mendapatkan daftar grup laporan.

Sumber daya: \*

### ListReports

Tindakan: `codebuild:ListReports`

Diperlukan untuk mendapatkan daftar laporan.

Sumber daya: \*

### ListReportsForReportGroup

Tindakan: `codebuild:ListReportsForReportGroup`

Diperlukan untuk mendapatkan daftar laporan untuk grup laporan.

Sumber daya: `arn:aws:codebuild:region-ID:account-ID:report-group/report-group-name`

### RetryBuild

Tindakan: `codebuild:RetryBuild`

Diperlukan untuk mencoba lagi build.

Sumber daya: `arn:aws:codebuild:region-ID:account-ID:project/project-name`

### StartBuild

Tindakan: `codebuild:StartBuild`

Diperlukan untuk mulai menjalankan build.

Sumber daya: `arn:aws:codebuild:region-ID:account-ID:project/project-name`

## StopBuild

Tindakan: `codebuild:StopBuild`

Diperlukan untuk mencoba berhenti menjalankan build.

Sumber daya: `arn:aws:codebuild:region-ID:account-ID:project/project-name`

## UpdateProject

Tindakan: `codebuild:UpdateProject`, `iam:PassRole`

Diperlukan untuk mengubah informasi tentang build.

Sumber Daya:

- `arn:aws:codebuild:region-ID:account-ID:project/project-name`
- `arn:aws:iam::account-ID:role/role-name`

## UpdateProjectVisibility

Tindakan: `codebuild:UpdateProjectVisibility`, `iam:PassRole`

Diperlukan untuk mengubah visibilitas publik dari bangunan proyek.

Sumber Daya:

- `arn:aws:codebuild:region-ID:account-ID:project/project-name`
- `arn:aws:iam::account-ID:role/role-name`

## UpdateReport<sup>1</sup>

Tindakan: `codebuild:UpdateReport`

Diperlukan untuk membuat atau memperbarui laporan pengujian.

Sumber daya: `arn:aws:codebuild:region-ID:account-ID:report-group/report-group-name`

## UpdateReportGroup

Tindakan: `codebuild:UpdateReportGroup`

Diperlukan untuk memperbarui grup laporan.

Sumber daya: `arn:aws:codebuild:region-ID:account-ID:report-group/report-group-name`

## UpdateWebhook

Tindakan: `codebuild:UpdateWebhook`

Diperlukan untuk memperbarui webhook.

Sumber daya: `arn:aws:codebuild:region-ID:account-ID:project/project-name`

<sup>1</sup> Digunakan hanya untuk izin. Tidak ada API untuk tindakan ini.

## Menggunakan tag untuk mengontrol akses ke AWS CodeBuild sumber daya

Kondisi dalam pernyataan kebijakan IAM adalah bagian dari sintaks yang dapat Anda gunakan untuk menentukan izin untuk CodeBuild tindakan berbasis proyek. Anda dapat membuat kebijakan yang mengizinkan atau menolak tindakan pada proyek berdasarkan tag yang terkait dengan proyek tersebut, lalu menerapkan kebijakan tersebut ke grup IAM yang Anda konfigurasi untuk mengelola pengguna. Untuk informasi tentang menerapkan tag ke proyek menggunakan konsol atau AWS CLI, lihat [Membuat proyek pembangunan di AWS CodeBuild](#). Untuk informasi tentang menerapkan tag menggunakan CodeBuild SDK, lihat [CreateProject](#) dan [Tag](#) di Referensi CodeBuild API. Untuk informasi tentang penggunaan tag untuk mengontrol akses ke AWS sumber daya, lihat [Mengontrol Akses ke AWS Sumber Daya Menggunakan Tag Sumber Daya](#) di Panduan Pengguna IAM.

### Example Contoh 1: Batasi tindakan CodeBuild proyek berdasarkan tag sumber daya

Contoh berikut menyangkal semua `BatchGetProjects` tindakan pada proyek yang ditandai dengan kunci `Environment` dengan nilai kunci `Production` Administrator pengguna harus melampirkan kebijakan IAM ini selain kebijakan pengguna terkelola untuk pengguna yang tidak sah. Kunci `aws:ResourceTag` kondisi digunakan untuk mengontrol akses ke sumber daya berdasarkan tag mereka.

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Deny",
 "Action": [
 "codebuild:BatchGetProjects"
],
 },
],
}
```

```
 "Resource": "*",
 "Condition": {
 "ForAnyValue:StringEquals": {
 "aws:ResourceTag/Environment": "Production"
 }
 }
 }
]
```

## Example Contoh 2: Batasi tindakan CodeBuild proyek berdasarkan tag permintaan

Kebijakan berikut menolak izin pengguna untuk `CreateProject` tindakan jika permintaan berisi tag dengan kunci `Environment` dan nilai `Production` kunci. Selain itu, kebijakan mencegah pengguna yang tidak sah ini memodifikasi proyek dengan menggunakan kunci `aws:TagKeys` kondisi untuk tidak mengizinkan `UpdateProject` jika permintaan berisi tag dengan kunci `Environment`. Administrator harus melampirkan kebijakan IAM ini selain kebijakan pengguna terkelola kepada pengguna yang tidak berwenang untuk melakukan tindakan ini. Kunci `aws:RequestTag` kondisi digunakan untuk mengontrol tag mana yang dapat diteruskan dalam permintaan IAM.

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Deny",
 "Action": [
 "codebuild:CreateProject"
],
 "Resource": "*",
 "Condition": {
 "ForAnyValue:StringEquals": {
 "aws:RequestTag/Environment": "Production"
 }
 }
 },
 {
 "Effect": "Deny",
 "Action": [
 "codebuild:UpdateProject"
],
 "Resource": "*",
 "Condition": {
```

```

 "ForAnyValue:StringEquals": {
 "aws:TagKeys": ["Environment"]
 }
 }
}
]
}

```

### Example Contoh 3: Tolak atau izinkan tindakan pada grup laporan berdasarkan tag sumber daya

Anda dapat membuat kebijakan yang mengizinkan atau menolak tindakan pada CodeBuild sumber daya (proyek dan grup laporan) berdasarkan AWS tag yang terkait dengan sumber daya tersebut, lalu menerapkan kebijakan tersebut ke grup IAM yang Anda konfigurasi untuk mengelola pengguna. *Misalnya, Anda dapat membuat kebijakan yang menolak semua CodeBuild tindakan pada grup laporan apa pun dengan kunci AWS tag Status dan nilai kunci Secret, lalu menerapkan kebijakan tersebut ke grup IAM yang Anda buat untuk pengembang umum (Pengembang).* Anda kemudian perlu memastikan bahwa pengembang yang bekerja pada grup laporan yang diberi tag tersebut bukan anggota grup *Pengembang* umum tersebut, melainkan milik grup IAM berbeda yang tidak memiliki kebijakan restriktif yang diterapkan (). SecretDeveIopers

Contoh berikut menyangkal semua CodeBuild tindakan pada grup laporan yang ditandai dengan kunci Status dan nilai kunci: Secret

```

{
 "Version": "2012-10-17",
 "Statement" : [
 {
 "Effect" : "Deny",
 "Action" : [
 "codebuild:BatchGetReportGroups",
 "codebuild:CreateReportGroup",
 "codebuild>DeleteReportGroup",
 "codebuild>ListReportGroups",
 "codebuild>ListReportsForReportGroup",
 "codebuild:UpdateReportGroup"
]
 "Resource" : "*",
 "Condition" : {
 "StringEquals" : "aws:ResourceTag/Status": "Secret"
 }
 }
]
}

```

```

 }
]
}

```

Example Contoh 4: Batasi CodeBuild tindakan AWSCodeBuildDeveloperAccess berdasarkan tag sumber daya

Anda dapat membuat kebijakan yang mengizinkan CodeBuild tindakan pada semua grup laporan dan proyek yang tidak ditandai dengan tag tertentu. Misalnya, kebijakan berikut memungkinkan [AWSCodeBuildDeveloperAccess](#) izin yang setara untuk semua grup laporan dan proyek kecuali yang ditandai dengan tag yang ditentukan:

```

{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": [
 "codebuild:StartBuild",
 "codebuild:StopBuild",
 "codebuild:BatchGet*",
 "codebuild:GetResourcePolicy",
 "codebuild:DescribeTestCases",
 "codebuild:List*",
 "codecommit:GetBranch",
 "codecommit:GetCommit",
 "codecommit:GetRepository",
 "codecommit:ListBranches",
 "cloudwatch:GetMetricStatistics",
 "events:DescribeRule",
 "events:ListTargetsByRule",
 "events:ListRuleNamesByTarget",
 "logs:GetLogEvents",
 "s3:GetBucketLocation",
 "s3:ListAllMyBuckets"
],
 "Resource": "*",
 "Condition": {
 "StringNotEquals": {
 "aws:ResourceTag/Status": "Secret",
 "aws:ResourceTag/Team": "Saanvi"
 }
 }
 }
]
}

```



```
 }
]
}
```

## Menampilkan sumber daya di konsol

AWS CodeBuild Konsol memerlukan `ListRepositories` izin untuk menampilkan daftar repositori untuk AWS akun Anda di AWS Wilayah tempat Anda masuk. Konsol juga termasuk fungsi Pergi ke sumber daya untuk secara cepat melakukan pencarian sensitif huruf besar/kecil untuk sumber daya. Pencarian ini dilakukan di AWS akun Anda di AWS Wilayah tempat Anda masuk. Sumber daya berikut ditampilkan di seluruh layanan berikut:

- AWS CodeBuild: Bangun proyek
- AWS CodeCommit: Repositori
- AWS CodeDeploy: Aplikasi
- AWS CodePipeline: Alur

Untuk melakukan pencarian ini di sumber daya di semua layanan, Anda harus memiliki izin berikut:

- CodeBuild: `ListProjects`
- CodeCommit: `ListRepositories`
- CodeDeploy: `ListApplications`
- CodePipeline: `ListPipelines`

Hasil tidak dikembalikan untuk sumber daya layanan jika Anda tidak memiliki izin untuk layanan tersebut. Bahkan jika Anda memiliki izin untuk melihat sumber daya, beberapa sumber daya tidak dikembalikan jika ada eksplisit Deny untuk melihat sumber daya tersebut.

## Validasi kepatuhan untuk AWS CodeBuild

Auditor pihak ketiga menilai keamanan dan kepatuhan AWS CodeBuild sebagai bagian dari beberapa program AWS kepatuhan. Program ini mencakup SOC, PCI, FedRAMP, HIPAA, dan lainnya.

Untuk daftar AWS layanan dalam lingkup program kepatuhan tertentu, lihat [AWS layanan dalam lingkup oleh program kepatuhan](#). Untuk informasi umum, lihat [Program kepatuhan AWS](#).

Anda dapat mengunduh laporan audit pihak ketiga menggunakan AWS Artifact. Untuk informasi selengkapnya, lihat [Mengunduh laporan di AWS Artifak](#).

Tanggung jawab kepatuhan Anda saat menggunakan CodeBuild ditentukan oleh sensitivitas data Anda, tujuan kepatuhan perusahaan Anda, dan hukum dan peraturan yang berlaku. Jika penggunaan CodeBuild Anda tunduk pada kepatuhan terhadap standar seperti HIPAA, PCI, atau FedRAMP, menyediakan sumber daya untuk membantu: AWS

- [Panduan memulai cepat keamanan dan kepatuhan — Panduan](#) penerapan ini membahas pertimbangan arsitektur dan memberikan langkah-langkah untuk menerapkan lingkungan dasar yang berfokus pada keamanan dan kepatuhan. AWS
- [Arsitektur untuk Whitepaper Keamanan dan Kepatuhan HIPAA — Whitepaper](#) ini menjelaskan bagaimana perusahaan dapat menggunakan untuk membuat aplikasi yang sesuai dengan HIPAA. AWS
- [AWS sumber daya kepatuhan](#) - Kumpulan buku kerja dan panduan ini mungkin berlaku untuk industri dan lokasi Anda.
- [AWS Config](#) AWS Layanan ini menilai seberapa baik konfigurasi sumber daya Anda mematuhi praktik internal, pedoman industri, dan peraturan.
- [AWS Security Hub](#)— Pantau penggunaan Anda AWS CodeBuild karena berkaitan dengan praktik terbaik keamanan dengan menggunakan [AWS Security Hub](#). Hub Keamanan menggunakan kontrol keamanan untuk mengevaluasi konfigurasi sumber daya dan standar keamanan guna membantu Anda mematuhi berbagai kerangka kerja kepatuhan. Untuk informasi selengkapnya tentang penggunaan Security Hub guna mengevaluasi CodeBuild sumber daya, lihat [AWS CodeBuild kontrol](#) di Panduan AWS Security Hub Pengguna.

## Ketahanan di AWS CodeBuild

Infrastruktur AWS global dibangun di sekitar AWS Wilayah dan Zona Ketersediaan. AWS Wilayah menyediakan beberapa Availability Zone yang terpisah secara fisik dan terisolasi, yang terhubung dengan latensi rendah, throughput tinggi, dan jaringan yang sangat redundan. Dengan Zona Ketersediaan, Anda dapat merancang dan mengoperasikan aplikasi dan basis data yang melakukan secara otomatis pinda saat gagal/failover di antara zona-zona tanpa terputus. Zona Ketersediaan lebih sangat tersedia, lebih toleran kesalahan, dan lebih dapat diskalakan daripada infrastruktur pusat data tunggal atau multi tradisional.

Untuk informasi selengkapnya tentang AWS Wilayah dan Availability Zone, lihat [infrastruktur AWS global](#).

# Keamanan infrastruktur di AWS CodeBuild

Sebagai layanan terkelola, AWS CodeBuild dilindungi oleh keamanan jaringan AWS global. Untuk informasi tentang layanan AWS keamanan dan cara AWS melindungi infrastruktur, lihat [Keamanan AWS Cloud](#). Untuk mendesain AWS lingkungan Anda menggunakan praktik terbaik untuk keamanan infrastruktur, lihat [Perlindungan Infrastruktur dalam Kerangka Kerja](#) yang AWS Diarsiteksikan dengan Baik Pilar Keamanan.

Anda menggunakan panggilan API yang AWS dipublikasikan untuk mengakses CodeBuild melalui jaringan. Klien harus mendukung hal-hal berikut:

- Keamanan Lapisan Pengangkutan (TLS). Kami mensyaratkan TLS 1.2 dan menganjurkan TLS 1.3.
- Sandi cocok dengan sistem kerahasiaan maju sempurna (perfect forward secrecy, PFS) seperti DHE (Ephemeral Diffie-Hellman) atau ECDHE (Elliptic Curve Ephemeral Diffie-Hellman). Sebagian besar sistem modern seperti Java 7 dan versi lebih baru mendukung mode-mode ini.

Selain itu, permintaan harus ditandatangani dengan menggunakan ID kunci akses dan kunci akses rahasia yang terkait dengan pengguna utama IAM. Atau Anda dapat menggunakan [AWS Security Token Service](#) (AWS STS) untuk menghasilkan kredensial keamanan sementara untuk menandatangani permintaan.

## Akses penyedia sumber Anda di CodeBuild

Untuk GitHub atau Server GitHub Perusahaan, Anda menggunakan token akses pribadi untuk mengakses penyedia sumber. Untuk Bitbucket, Anda menggunakan kata sandi aplikasi untuk mengakses penyedia sumber.

Topik

- [GitHub dan token akses Server GitHub Perusahaan](#)
- [Kata sandi aplikasi Bitbucket](#)

## GitHub dan token akses Server GitHub Perusahaan

### Prasyarat token akses

Sebelum memulai, Anda harus menambahkan cakupan izin yang tepat ke token GitHub akses Anda.

Untuk GitHub, token akses pribadi Anda harus memiliki cakupan berikut.

- `repo`: Memberikan kontrol penuh atas repositori pribadi.
- `repo:status`: Memberikan akses baca/tulis ke status komit repositori publik dan pribadi.
- `admin:repo_hook`: Memberikan kontrol penuh atas kait repositori. Cakupan ini tidak diperlukan jika token Anda memiliki `repo` ruang lingkup.

Untuk informasi selengkapnya, lihat [Memahami cakupan untuk aplikasi OAuth](#) di situs web. GitHub

## Connect GitHub dengan token akses (konsol)

Untuk menggunakan konsol untuk menghubungkan proyek Anda GitHub menggunakan token akses, lakukan hal berikut saat Anda membuat proyek. Untuk informasi, lihat [Buat proyek build \(konsol\)](#).

1. Untuk penyedia Sumber, pilih GitHub.
2. Untuk Repositori, pilih Connect with a GitHub personal access token.
3. Dalam token akses GitHub pribadi, masukkan token akses GitHub pribadi Anda.
4. Pilih Simpan token.

## Connect GitHub dengan token akses (CLI)

Ikuti langkah-langkah ini untuk menggunakan AWS CLI untuk menghubungkan proyek Anda GitHub dengan menggunakan token akses. Untuk informasi tentang menggunakan AWS CLI with AWS CodeBuild, lihat [Referensi baris perintah](#).

1. Jalankan perintah `import-source-credentials`:

```
aws codebuild import-source-credentials --generate-cli-skeleton
```

Data berformat JSON muncul di output. Salin data ke file (misalnya, `import-source-credentials.json`) di lokasi di komputer lokal atau contoh di AWS CLI mana diinstal. Ubah data yang disalin sebagai berikut, dan simpan hasil Anda.

```
{
 "serverType": "server-type",
 "authType": "auth-type",
 "shouldOverwrite": "should-overwrite",
 "token": "token",
```

```
"username": "username"
}
```

Ganti yang berikut ini:

- ***server-type***: Nilai yang dibutuhkan. Penyedia sumber yang digunakan untuk kredensi ini. Nilai yang valid adalah GITHUB atau GITHUB\_ENTERPRISE.
  - ***auth-type***: *Nilai* yang diperlukan. Jenis otentikasi yang digunakan untuk terhubung ke repositori Server GitHub atau GitHub Enterprise. Nilai yang valid termasuk PERSONAL\_ACCESS\_TOKEN dan BASIC\_AUTH. Anda tidak dapat menggunakan CodeBuild API untuk membuat koneksi OAUTH. Anda harus menggunakan CodeBuild konsol sebagai gantinya.
  - ***shoud-overwrite***: Nilai opsional. Setel `false` untuk mencegah penimpaan kredensial sumber repositori. Setel `true` untuk menimpa kredensial sumber repositori. Nilai default-nya adalah `true`.
  - ***token***: Nilai yang dibutuhkan. Untuk GitHub atau Server GitHub Perusahaan, ini adalah token akses pribadi.
  - ***nama pengguna***: Nilai opsional. Parameter ini diabaikan untuk GitHub dan penyedia sumber Server GitHub Perusahaan.
2. Untuk menghubungkan akun Anda dengan token akses, beralih ke direktori yang berisi `import-source-credentials.json` file yang Anda simpan di langkah 1 dan jalankan `import-source-credentials` perintah lagi.

```
aws codebuild import-source-credentials --cli-input-json file://import-source-credentials.json
```

Data berformat JSON muncul di output dengan Amazon Resource Name (ARN).

```
{
 "arn": "arn:aws:codebuild:region:account-id:token/server-type"
}
```

#### Note

Jika Anda menjalankan `import-source-credentials` perintah dengan jenis server dan jenis autentikasi yang sama untuk kedua kalinya, token akses yang disimpan diperbarui.

Setelah akun Anda terhubung dengan token akses, Anda dapat menggunakannya `create-project` untuk membuat CodeBuild proyek Anda. Untuk informasi selengkapnya, lihat [Buat proyek build \(AWS CLI\)](#).

3. Untuk melihat token akses yang terhubung, jalankan `list-source-credentials` perintah.

```
aws codebuild list-source-credentials
```

`sourceCredentialsInfos` objek berformat JSON muncul di output:

```
{
 "sourceCredentialsInfos": [
 {
 "authType": "auth-type",
 "serverType": "server-type",
 "arn": "arn"
 }
]
}
```

`sourceCredentialsObject` ini berisi daftar informasi kredensial sumber yang terhubung:

- `authType` ini adalah jenis otentikasi yang digunakan oleh kredensial. Ini dapat berupa `OAUTH`, `BASIC_AUTH`, atau `PERSONAL_ACCESS_TOKEN`.
  - `serverType` ini adalah jenis penyedia sumber. Ini dapat berupa `GITHUB`, `GITHUB_ENTERPRISE`, atau `BITBUCKET`.
  - `arn` itu adalah ARN dari token.
4. Untuk memutuskan sambungan dari penyedia sumber dan menghapus token aksesnya, jalankan `delete-source-credentials` perintah dengan ARN-nya.

```
aws codebuild delete-source-credentials --arn arn-of-your-credentials
```

Data berformat JSON dikembalikan dengan ARN dari kredensial yang dihapus.

```
{
 "arn": "arn:aws:codebuild:region:account-id:token/server-type"
}
```

# Kata sandi aplikasi Bitbucket

## Prasyarat kata sandi aplikasi

Sebelum memulai, Anda harus menambahkan cakupan izin yang tepat ke kata sandi aplikasi Bitbucket Anda.

Untuk Bitbucket, kata sandi aplikasi Anda harus memiliki cakupan berikut.

- `repository:read`: Memberikan akses baca ke semua repositori yang dapat diakses oleh pengguna otorisasi.
- `pullrequest:read`: Memberikan akses baca ke permintaan tarik. Jika project Anda memiliki webhook Bitbucket, kata sandi aplikasi Anda harus memiliki cakupan ini.
- `webhook`: Memberikan akses ke webhook. Jika project Anda memiliki operasi webhook, kata sandi aplikasi Anda harus memiliki cakupan ini.

Untuk informasi selengkapnya, lihat [Cakupan untuk Bitbucket Cloud REST API](#) dan [OAuth di Bitbucket Cloud di situs web Bitbucket](#).

## Hubungkan Bitbucket dengan kata sandi aplikasi (konsol)

Untuk menggunakan konsol untuk menghubungkan proyek Anda ke Bitbucket menggunakan kata sandi aplikasi, lakukan hal berikut saat Anda membuat proyek. Untuk informasi, lihat [Buat proyek build \(konsol\)](#).

1. Untuk penyedia Sumber, pilih Bitbucket.

### Note

CodeBuild tidak mendukung Bitbucket Server.

2. Untuk Repositori, pilih Connect with a Bitbucket app password.
3. Di nama pengguna Bitbucket, masukkan nama pengguna Bitbucket Anda.
4. Dalam kata sandi aplikasi Bitbucket, masukkan kata sandi aplikasi Bitbucket Anda.
5. Pilih Simpan kredensial Bitbucket.

## Connect Bitbucket dengan password aplikasi (CLI)

Ikuti langkah-langkah berikut untuk menggunakan AWS CLI untuk menghubungkan proyek Anda ke Bitbucket menggunakan kata sandi aplikasi. Untuk informasi tentang menggunakan AWS CLI with AWS CodeBuild, lihat [Referensi baris perintah](#).

1. Jalankan perintah `import-source-credentials`:

```
aws codebuild import-source-credentials --generate-cli-skeleton
```

Data berformat JSON muncul di output. Salin data ke file (misalnya, `import-source-credentials.json`) di lokasi di komputer lokal atau contoh di AWS CLI mana diinstal. Ubah data yang disalin sebagai berikut, dan simpan hasil Anda.

```
{
 "serverType": "BITBUCKET",
 "authType": "auth-type",
 "shouldOverwrite": "should-overwrite",
 "token": "token",
 "username": "username"
}
```

Ganti yang berikut ini:

- *auth-type*: *Nilai* yang diperlukan. Jenis otentikasi yang digunakan untuk terhubung ke repositori Bitbucket. Nilai yang valid termasuk `PERSONAL_ACCESS_TOKEN` dan `BASIC_AUTH`. Anda tidak dapat menggunakan CodeBuild API untuk membuat koneksi OAUTH. Anda harus menggunakan CodeBuild konsol sebagai gantinya.
  - *shoud-overwrite*: Nilai opsional. Setel `false` untuk mencegah penimpaan kredenal sumber repositori. Setel `true` untuk menimpa kredenal sumber repositori. Nilai default-nya adalah `true`.
  - *token*: Nilai yang dibutuhkan. Untuk Bitbucket, ini adalah kata sandi aplikasi.
  - *nama pengguna*: Nilai opsional. Nama pengguna Bitbucket kapan `authType` adalah `BASIC_AUTH`. Parameter ini diabaikan untuk jenis penyedia sumber atau koneksi lainnya.
2. Untuk menghubungkan akun Anda dengan kata sandi aplikasi, alihkan ke direktori yang berisi `import-source-credentials.json` file yang Anda simpan di langkah 1 dan jalankan `import-source-credentials` perintah lagi.



```
aws codebuild import-source-credentials --cli-input-json file://import-source-credentials.json
```

Data berformat JSON muncul di output dengan Amazon Resource Name (ARN).

```
{
 "arn": "arn:aws:codebuild:region:account-id:token/server-type"
}
```

### Note

Jika Anda menjalankan `import-source-credentials` perintah dengan jenis server dan jenis autentikasi yang sama untuk kedua kalinya, token akses yang disimpan diperbarui.

Setelah akun Anda terhubung dengan kata sandi aplikasi, Anda dapat menggunakannya `create-project` untuk membuat CodeBuild proyek Anda. Untuk informasi selengkapnya, lihat [Buat proyek build \(AWS CLI\)](#).

3. Untuk melihat kata sandi aplikasi yang terhubung, jalankan `list-source-credentials` perintah.

```
aws codebuild list-source-credentials
```

`sourceCredentialsInfos` objek berformat JSON muncul di output:

```
{
 "sourceCredentialsInfos": [
 {
 "authType": "auth-type",
 "serverType": "BITBUCKET",
 "arn": "arn"
 }
]
}
```

`sourceCredentialsObject` ini berisi daftar informasi kredensial sumber yang terhubung:

- `authType` ini adalah jenis otentikasi yang digunakan oleh kredensial. Ini dapat berupa `OAUTH`, `BASIC_AUTH`, atau `PERSONAL_ACCESS_TOKEN`.

- `arn` itu adalah ARN dari token.
4. Untuk memutuskan sambungan dari penyedia sumber dan menghapus kata sandi aplikasinya, jalankan `delete-source-credentials` perintah dengan ARN-nya.

```
aws codebuild delete-source-credentials --arn arn-of-your-credentials
```

Data berformat JSON dikembalikan dengan ARN dari kredensial yang dihapus.

```
{
 "arn": "arn:aws:codebuild:region:account-id:token/server-type"
}
```

## Pencegahan confused deputy lintas layanan

Masalah confused deputy adalah masalah keamanan saat entitas yang tidak memiliki izin untuk melakukan suatu tindakan dapat memaksa entitas yang lebih berhak untuk melakukan tindakan tersebut. Pada tahun AWS, peniruan lintas layanan dapat mengakibatkan masalah wakil yang membingungkan. Peniruan identitas lintas layanan dapat terjadi ketika satu layanan (layanan pemanggil) memanggil layanan lain (layanan yang dipanggil). Layanan pemanggil dapat dimanipulasi menggunakan izinnya untuk bertindak pada sumber daya pelanggan lain dengan cara yang seharusnya tidak dilakukannya kecuali bila memiliki izin untuk mengakses. Untuk mencegah hal ini, AWS menyediakan alat yang membantu Anda melindungi data untuk semua layanan dengan pengguna utama layanan yang telah diberi akses ke sumber daya di akun Anda.

Sebaiknya gunakan kunci konteks kondisi [aws:SourceAccount](#) global [aws:SourceArn](#) dan dalam kebijakan sumber daya untuk membatasi izin yang AWS CodeBuild memberikan layanan lain ke sumber daya. Gunakan `aws:SourceArn` jika Anda hanya ingin satu sumber daya dikaitkan dengan akses lintas layanan. Gunakan `aws:SourceAccount` jika Anda ingin mengizinkan sumber daya apa pun di akun tersebut dikaitkan dengan penggunaan lintas layanan.

Cara paling efektif untuk melindungi dari masalah confused deputy adalah dengan menggunakan kunci konteks kondisi global `aws:SourceArn` dengan ARN sumber daya penuh. Jika Anda tidak mengetahui ARN lengkap sumber daya atau jika Anda menentukan beberapa sumber daya, gunakan kunci kondisi konteks `aws:SourceArn` global dengan karakter wildcard (\*) untuk bagian ARN yang tidak diketahui. Misalnya, `arn:aws:codebuild:*:123456789012:*`.

Jika `aws:SourceArn` nilainya tidak berisi ID akun, seperti ARN bucket Amazon S3, Anda harus menggunakan kedua kunci konteks kondisi global untuk membatasi izin.

Nilai `aws:SourceArn` harus menjadi CodeBuild proyek ARN.

Contoh berikut menunjukkan bagaimana Anda dapat menggunakan kunci konteks kondisi `aws:SourceAccount` global `aws:SourceArn` dan CodeBuild untuk mencegah masalah wakil yang membingungkan.

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Principal": {
 "Service": "codebuild.amazonaws.com"
 },
 "Action": "sts:AssumeRole",
 "Condition": {
 "StringEquals": {
 "aws:SourceArn": "arn:aws:codebuild:region-ID:account-ID:project/project-name"
 }
 }
 }
]
}
```

# Topik lanjutan

Bagian ini mencakup beberapa topik lanjutan yang berguna bagi AWS CodeBuild pengguna yang lebih berpengalaman.

## Topik

- [Pengaturan lanjutan](#)
- [Referensi baris perintah untuk AWS CodeBuild](#)
- [AWS Referensi SDK dan alat AWS CodeBuild](#)
- [Tentukan AWS CodeBuild titik akhir](#)
- [Gunakan AWS CodePipeline dengan AWS CodeBuild untuk menguji kode dan menjalankan build](#)
- [Gunakan AWS CodeBuild dengan Jenkins](#)
- [Gunakan AWS CodeBuild dengan Codecov](#)
- [Gunakan AWS CodeBuild aplikasi tanpa server](#)

## Pengaturan lanjutan

Jika Anda mengikuti langkah-langkah [Memulai menggunakan konsol](#) untuk mengakses AWS CodeBuild untuk pertama kalinya, kemungkinan besar Anda tidak memerlukan informasi dalam topik ini. Namun, saat Anda terus menggunakan CodeBuild, Anda mungkin ingin melakukan hal-hal seperti memberi grup IAM dan pengguna di organisasi Anda akses CodeBuild, memodifikasi peran layanan yang ada di IAM atau AWS KMS keys untuk mengakses CodeBuild, atau menyiapkan workstation organisasi Anda untuk mengakses CodeBuild. AWS CLI Topik ini menjelaskan cara menyelesaikan langkah persiapan terkait.

Kami berasumsi Anda sudah memiliki AWS akun. Namun, jika Anda belum memilikinya, buka <http://aws.amazon.com>, pilih Masuk ke Konsol, dan ikuti petunjuk online.

## Topik

- [Menambahkan izin CodeBuild akses ke grup atau pengguna IAM](#)
- [Membuat peran CodeBuild layanan](#)
- [Membuat dan mengkonfigurasi kunci yang dikelola pelanggan CodeBuild](#)
- [Instal dan konfigurasi AWS CLI](#)

## Menambahkan izin CodeBuild akses ke grup atau pengguna IAM

Untuk mengakses AWS CodeBuild dengan grup atau pengguna IAM, Anda harus menambahkan izin akses. Bagian ini menjelaskan cara melakukan ini dengan konsol IAM atau AWS CLI.

Jika Anda akan mengakses CodeBuild dengan akun AWS root Anda (tidak disarankan) atau pengguna administrator di AWS akun Anda, maka Anda tidak perlu mengikuti instruksi ini.

Untuk informasi tentang akun AWS root dan pengguna administrator, lihat [Pengguna Akun AWS root](#) dan [Membuat pengguna Akun AWS root Pertama Anda dan Grup](#) di Panduan Pengguna.

Untuk menambahkan izin CodeBuild akses ke grup IAM atau pengguna (konsol)

1. Buka konsol IAM di <https://console.aws.amazon.com/iam/>.

Anda seharusnya sudah masuk ke AWS Management Console dengan menggunakan salah satu dari berikut ini:

- Akun AWS root Anda. Ini tidak disarankan. Untuk informasi selengkapnya, lihat [Pengguna Akun AWS root](#) di Panduan Pengguna.
- Pengguna administrator di AWS akun Anda. Untuk informasi selengkapnya, lihat [Membuat pengguna Akun AWS root Pertama Anda dan Grup](#) di Panduan Pengguna.
- Pengguna di AWS akun Anda dengan izin untuk melakukan serangkaian tindakan minimum berikut:

```
iam:AttachGroupPolicy
iam:AttachUserPolicy
iam:CreatePolicy
iam>ListAttachedGroupPolicies
iam>ListAttachedUserPolicies
iam>ListGroups
iam>ListPolicies
iam>ListUsers
```

Untuk informasi selengkapnya, [lihat Ringkasan Kebijakan IAM](#) di Panduan Pengguna.

2. Di panel navigasi, pilih Kebijakan.
3. Untuk menambahkan kumpulan izin AWS CodeBuild akses khusus ke grup IAM atau pengguna IAM, lanjutkan ke langkah 4 dalam prosedur ini.

Untuk menambahkan seperangkat izin CodeBuild akses default ke grup IAM atau pengguna IAM, pilih Jenis Kebijakan, AWSDikelola, lalu lakukan hal berikut:

- Untuk menambahkan izin akses penuh ke CodeBuild, centang kotak bernama `AWSCodeBuildAdminAccess`, pilih Tindakan Kebijakan, lalu pilih Lampirkan. Centang kotak di samping grup atau pengguna IAM target, lalu pilih Lampirkan Kebijakan. Ulangi ini untuk kebijakan bernama `AmazonS3ReadOnlyAccess` dan `IAMFullAccess`.
- Untuk menambahkan izin akses ke semua hal CodeBuild kecuali administrasi proyek build, centang kotak bernama `AWSCodeBuildDeveloperAccess`, pilih Tindakan Kebijakan, lalu pilih Lampirkan. Centang kotak di samping grup atau pengguna IAM target, lalu pilih Lampirkan Kebijakan. Ulangi ini untuk kebijakan bernama `AmazonS3ReadOnlyAccess`.
- Untuk menambahkan izin akses hanya-baca CodeBuild, pilih kotak bernama `AWSCodeBuildReadOnlyAccess`. Centang kotak di samping grup atau pengguna IAM target, lalu pilih Lampirkan Kebijakan. Ulangi ini untuk kebijakan bernama `AmazonS3ReadOnlyAccess`.

Anda sekarang telah menambahkan seperangkat izin CodeBuild akses default ke grup atau pengguna IAM. Lewati langkah-langkah lainnya dalam prosedur ini.

4. Pilih Buat Kebijakan.
5. Pada halaman Buat Kebijakan, di samping Buat Kebijakan Anda Sendiri, pilih Pilih.
6. Pada halaman Tinjau kebijakan, untuk nama kebijakan, masukkan nama untuk kebijakan tersebut (misalnya `CodeBuildAccessPolicy`). Jika Anda menggunakan nama yang berbeda, pastikan untuk menggunakannya di seluruh prosedur ini.
7. Untuk Dokumen Kebijakan, masukkan yang berikut, lalu pilih Buat Kebijakan.

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Sid": "CodeBuildAccessPolicy",
 "Effect": "Allow",
 "Action": [
 "codebuild:*"
],
 "Resource": "*"
 },
 {
```

```

 "Sid": "CodeBuildRolePolicy",
 "Effect": "Allow",
 "Action": [
 "iam:PassRole"
],
 "Resource": "arn:aws:iam::account-ID:role/role-name"
},
{
 "Sid": "CloudWatchLogsAccessPolicy",
 "Effect": "Allow",
 "Action": [
 "logs:FilterLogEvents",
 "logs:GetLogEvents"
],
 "Resource": "*"
},
{
 "Sid": "S3AccessPolicy",
 "Effect": "Allow",
 "Action": [
 "s3:CreateBucket",
 "s3:GetObject",
 "s3:List*",
 "s3:PutObject"
],
 "Resource": "*"
},
{
 "Sid": "S3BucketIdentity",
 "Effect": "Allow",
 "Action": [
 "s3:GetBucketAcl",
 "s3:GetBucketLocation"
],
 "Resource": "*"
}
]
}

```

### Note

Kebijakan ini memungkinkan akses ke semua CodeBuild tindakan dan ke sejumlah besar AWS sumber daya. Untuk membatasi izin terhadap CodeBuild tindakan tertentu,

ubah nilai `codebuild:*` dalam pernyataan CodeBuild kebijakan. Untuk informasi selengkapnya, lihat [Pengelolaan identitas dan akses](#). Untuk membatasi akses ke AWS sumber daya tertentu, ubah nilai `Resource` objek. Untuk informasi selengkapnya, lihat [Pengelolaan identitas dan akses](#).

CodeBuildRolePolicyPernyataan ini diperlukan untuk memungkinkan membangun proyek yang akan dibuat atau dimodifikasi.

8. Di panel navigasi, pilih Grup atau Pengguna.
9. Dalam daftar grup atau pengguna, pilih nama grup IAM atau pengguna IAM yang ingin Anda tambahkan izin CodeBuild akses.
10. Untuk grup, pada halaman pengaturan grup, pada tab Izin, luaskan Kebijakan Terkelola, lalu pilih Lampirkan Kebijakan.

Untuk pengguna, pada halaman pengaturan pengguna, pada tab Izin, pilih Tambahkan izin.

11. Untuk grup, pada halaman Lampirkan Kebijakan, pilih CodeBuildAccessPolicy, lalu pilih Lampirkan Kebijakan.

Untuk pengguna, pada halaman Tambahkan izin, pilih Lampirkan kebijakan yang ada secara langsung. Pilih CodeBuildAccessPolicy, pilih Berikutnya: Tinjau, lalu pilih Tambahkan izin.

Untuk menambahkan izin CodeBuild akses ke grup IAM atau pengguna (AWS CLI)

1. Pastikan Anda telah mengkonfigurasi AWS CLI dengan kunci AWS akses dan kunci akses AWS rahasia yang sesuai dengan salah satu entitas IAM, seperti yang dijelaskan dalam prosedur sebelumnya. Untuk informasi selengkapnya, lihat [Mengatur dengan AWS Command Line Interface di Panduan AWS Command Line Interface Pengguna](#).
2. Untuk menambahkan kumpulan izin AWS CodeBuild akses khusus ke grup IAM atau pengguna IAM, lewati ke langkah 3 dalam prosedur ini.

Untuk menambahkan seperangkat izin CodeBuild akses default ke grup IAM atau pengguna IAM, lakukan hal berikut:

Jalankan salah satu perintah berikut, tergantung apakah Anda ingin menambahkan izin ke grup IAM atau pengguna:

```
aws iam attach-group-policy --group-name group-name --policy-arn policy-arn
```



```
aws iam attach-user-policy --user-name user-name --policy-arn policy-arn
```

Anda harus menjalankan perintah tiga kali, mengganti nama *grup atau nama pengguna dengan nama* grup IAM atau nama pengguna, dan mengganti *policy-arn* satu kali untuk masing-masing kebijakan berikut Amazon Resource Names (ARN):

- Untuk menambahkan izin akses penuh ke CodeBuild, gunakan ARN kebijakan berikut:
  - `arn:aws:iam::aws:policy/AWSCodeBuildAdminAccess`
  - `arn:aws:iam::aws:policy/AmazonS3ReadOnlyAccess`
  - `arn:aws:iam::aws:policy/IAMFullAccess`
- Untuk menambahkan izin akses CodeBuild untuk semua hal kecuali membangun administrasi proyek, gunakan ARN kebijakan berikut:
  - `arn:aws:iam::aws:policy/AWSCodeBuildDeveloperAccess`
  - `arn:aws:iam::aws:policy/AmazonS3ReadOnlyAccess`
- Untuk menambahkan izin akses hanya-baca CodeBuild, gunakan ARN kebijakan berikut:
  - `arn:aws:iam::aws:policy/AWSCodeBuildReadOnlyAccess`
  - `arn:aws:iam::aws:policy/AmazonS3ReadOnlyAccess`

Anda sekarang telah menambahkan seperangkat izin CodeBuild akses default ke grup atau pengguna IAM. Lewati langkah-langkah lainnya dalam prosedur ini.

3. Dalam direktori kosong di workstation lokal atau contoh diAWS CLI mana diinstal, membuat file bernama `put-group-policy.json` atau `put-user-policy.json`. Jika Anda menggunakan nama file yang berbeda, pastikan untuk menggunakannya di seluruh prosedur ini.

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Sid": "CodeBuildAccessPolicy",
 "Effect": "Allow",
 "Action": [
 "codebuild:*"
],
 "Resource": "*"
 },
 {
 "Sid": "CodeBuildRolePolicy",
```

```

 "Effect": "Allow",
 "Action": [
 "iam:PassRole"
],
 "Resource": "arn:aws:iam::account-ID:role/role-name"
 },
 {
 "Sid": "CloudWatchLogsAccessPolicy",
 "Effect": "Allow",
 "Action": [
 "logs:FilterLogEvents",
 "logs:GetLogEvents"
],
 "Resource": "*"
 },
 {
 "Sid": "S3AccessPolicy",
 "Effect": "Allow",
 "Action": [
 "s3:CreateBucket",
 "s3:GetObject",
 "s3:List*",
 "s3:PutObject"
],
 "Resource": "*"
 },
 {
 "Sid": "S3BucketIdentity",
 "Effect": "Allow",
 "Action": [
 "s3:GetBucketAcl",
 "s3:GetBucketLocation"
],
 "Resource": "*"
 }
]
}

```

### Note

Kebijakan ini memungkinkan akses ke semua CodeBuild tindakan dan ke sejumlah besar AWS sumber daya. Untuk membatasi izin terhadap CodeBuild tindakan tertentu, ubah nilai `codebuild:*` dalam pernyataan CodeBuild kebijakan. Untuk informasi

selengkapnya, lihat [Pengelolaan identitas dan akses](#). Untuk membatasi akses keAWS sumber daya tertentu, ubah nilaiResource objek terkait. Untuk informasi lebih lanjut, lihat[Pengelolaan identitas dan akses](#) atau dokumentasi keamananAWS layanan tertentu. CodeBuildRolePolicyPernyataan ini diperlukan untuk memungkinkan membangun proyek yang akan dibuat atau dimodifikasi.

4. Beralih ke direktori tempat Anda menyimpan file, dan kemudian jalankan salah satu perintah berikut. Anda dapat menggunakan nilai yang berbeda untukCodeBuildGroupAccessPolicy danCodeBuildUserAccessPolicy. Jika Anda menggunakan nilai yang berbeda, pastikan untuk menggunakannya di sini.

Untuk grup IAM:

```
aws iam put-group-policy --group-name group-name --policy-name
CodeBuildGroupAccessPolicy --policy-document file://put-group-policy.json
```

Untuk pengguna:

```
aws iam put-user-policy --user-name user-name --policy-name
CodeBuildUserAccessPolicy --policy-document file://put-user-policy.json
```

Dalam perintah sebelumnya, ganti nama *grup atau nama pengguna* dengan nama grup IAM target atau pengguna.

## Membuat peran CodeBuild layanan

Anda memerlukan peranAWS CodeBuild layanan sehingga CodeBuild dapat berinteraksi denganAWS layanan dependen atas nama Anda. Anda dapat membuat peran CodeBuild layanan dengan menggunakan CodeBuild atauAWS CodePipeline konsol. Untuk informasi, lihat:

- [Buat proyek build \(konsol\)](#)
- [Membuat pipeline yang menggunakanCodeBuild \(CodePipelinekonsol\)](#)
- [Menambahkan aksiCodeBuild build ke pipeline \(CodePipelinekonsol\)](#)
- [Mengubah setelan proyek build \(konsol\)](#)

Jika Anda tidak berencana untuk menggunakan konsol ini, bagian ini menjelaskan cara membuat peran CodeBuild layanan dengan konsol IAM atau konsol AWS CLI.

### Important

CodeBuild menggunakan peran layanan untuk semua operasi yang dilakukan atas nama Anda. Jika peran tersebut mencakup izin yang seharusnya tidak dimiliki pengguna, Anda dapat secara tidak sengaja meningkatkan izin pengguna. Pastikan peran memberikan [keistimewaan paling rendah](#).

Peran layanan yang dijelaskan di halaman ini berisi kebijakan yang memberikan izin minimum yang diperlukan untuk digunakan CodeBuild. Anda mungkin perlu menambahkan izin tambahan, tergantung kasus penggunaan Anda.

Untuk membuat peran CodeBuild layanan (konsol)

1. Buka konsol IAM di <https://console.aws.amazon.com/iam/>.

Anda seharusnya sudah masuk ke konsol dengan menggunakan salah satu dari berikut ini:

- Akun AWS root Anda. Ini tidak disarankan. Untuk informasi selengkapnya, lihat [Pengguna Akun AWS root](#) di Panduan Pengguna.
- Pengguna administrator di AWS akun Anda. Untuk informasi selengkapnya, lihat [Membuat pengguna Akun AWS root Pertama Anda dan Grup](#) di Panduan Pengguna.
- Pengguna di AWS akun Anda dengan izin untuk melakukan serangkaian tindakan minimum berikut:

```
iam:AddRoleToInstanceProfile
iam:AttachRolePolicy
iam:CreateInstanceProfile
iam:CreatePolicy
iam:CreateRole
iam:GetRole
iam>ListAttachedRolePolicies
iam>ListPolicies
iam>ListRoles
iam:PassRole
iam:PutRolePolicy
iam:UpdateAssumeRolePolicy
```

Untuk informasi selengkapnya, [lihat Ringkasan Kebijakan IAM](#) di Panduan Pengguna.

2. Di panel navigasi, pilih Kebijakan.
3. Pilih Buat Kebijakan.
4. Pada halaman Buat Kebijakan, pilih JSON.
5. Untuk kebijakan JSON, masukkan yang berikut, lalu pilih Kebijakan Tinjauan:

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Sid": "CloudWatchLogsPolicy",
 "Effect": "Allow",
 "Action": [
 "logs:CreateLogGroup",
 "logs:CreateLogStream",
 "logs:PutLogEvents"
],
 "Resource": "*"
 },
 {
 "Sid": "CodeCommitPolicy",
 "Effect": "Allow",
 "Action": [
 "codecommit:GitPull"
],
 "Resource": "*"
 },
 {
 "Sid": "S3GetObjectPolicy",
 "Effect": "Allow",
 "Action": [
 "s3:GetObject",
 "s3:GetObjectVersion"
],
 "Resource": "*"
 },
 {
 "Sid": "S3PutObjectPolicy",
 "Effect": "Allow",
 "Action": [
 "s3:PutObject"
]
 }
]
}
```

```

],
 "Resource": "*"
 },
 {
 "Sid": "ECRPullPolicy",
 "Effect": "Allow",
 "Action": [
 "ecr:BatchCheckLayerAvailability",
 "ecr:GetDownloadUrlForLayer",
 "ecr:BatchGetImage"
],
 "Resource": "*"
 },
 {
 "Sid": "ECRAuthPolicy",
 "Effect": "Allow",
 "Action": [
 "ecr:GetAuthorizationToken"
],
 "Resource": "*"
 },
 {
 "Sid": "S3BucketIdentity",
 "Effect": "Allow",
 "Action": [
 "s3:GetBucketAcl",
 "s3:GetBucketLocation"
],
 "Resource": "*"
 }
]
}

```

#### Note

Kebijakan ini berisi pernyataan yang memungkinkan akses ke sejumlah besar AWS sumber daya. Untuk membatasi AWS CodeBuild untuk mengakses AWS sumber daya tertentu, mengubah nilai `Resource` array. Untuk informasi lebih lanjut, lihat dokumentasi keamanan untuk AWS layanan ini.

6. Pada halaman Kebijakan Tinjauan, untuk Nama Kebijakan, masukkan nama untuk kebijakan (misalnya, **CodeBuildServiceRolePolicy**), lalu pilih Buat kebijakan.

**Note**

Jika Anda menggunakan nama yang berbeda, pastikan untuk menggunakannya di seluruh prosedur ini.

7. Di panel navigasi, pilih Peran.
8. Pilih Buat peran.
9. Pada halaman Buat peran, dengan AWSLayanan sudah dipilih, pilih CodeBuild, lalu pilih Berikutnya:Izin.
10. Pada halaman Lampirkan kebijakan izin, pilih CodeBuildServiceRolePolicy, lalu pilih Berikutnya:Tinjauan.
11. Pada halaman Buat peran dan peninjauan, untuk nama peran, masukkan nama untuk peran (misalnya**CodeBuildServiceRole**), dan pilih Buat peran.

Untuk membuat peran CodeBuild layanan (AWS CLI)

1. Pastikan Anda telah mengkonfigurasiAWS CLI dengan kunciAWS akses dan kunci aksesAWS rahasia yang sesuai dengan salah satu entitas IAM, seperti yang dijelaskan dalam prosedur sebelumnya. Untuk informasi selengkapnya, lihat [Mengatur denganAWS Command Line Interface di](#) PanduanAWS Command Line Interface Pengguna.
2. Dalam direktori kosong pada workstation lokal atau contoh di manaAWS CLI diinstal, membuat dua file bernamacreate-role.json danput-role-policy.json. Jika Anda memilih nama file yang berbeda, pastikan untuk menggunakannya di seluruh prosedur ini.

create-role.json:

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Principal": {
 "Service": "codebuild.amazonaws.com"
 },
 "Action": "sts:AssumeRole"
 }
]
}
```

}

**Note**

Kami menyarankan Anda menggunakan `aws:SourceAccount` dan `aws:SourceArn` kondisi untuk melindungi diri Anda [dari masalah wakil yang bingung](#). Misalnya, Anda dapat mengedit kebijakan kepercayaan sebelumnya dengan blok kondisi berikut. `aws:SourceAccount` adalah pemilik CodeBuild proyek dan `aws:SourceArn` adalah CodeBuild proyek ARN.

Jika Anda ingin membatasi peran layanan Anda ke AWS akun, `create-role.json` mungkin terlihat mirip dengan ini:

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Principal": {
 "Service": "codebuild.amazonaws.com"
 },
 "Action": "sts:AssumeRole",
 "Condition": {
 "StringEquals": {
 "aws:SourceAccount": [
 "account-ID"
]
 }
 }
 }
]
}
```

Jika Anda ingin membatasi peran layanan Anda ke CodeBuild proyek tertentu, `create-role.json` mungkin terlihat mirip dengan ini:

```
{
 "Version": "2012-10-17",
 "Statement": [
```



```
{
 "Effect": "Allow",
 "Principal": {
 "Service": "codebuild.amazonaws.com"
 },
 "Action": "sts:AssumeRole",
 "Condition": {
 "StringEquals": {
 "aws:SourceArn": "arn:aws:codebuild:region-ID:account-ID:project/project-name"
 }
 }
}
```

#### Note

Jika Anda tidak tahu atau belum memutuskan nama untuk CodeBuild proyek Anda dan menginginkan pembatasan kebijakan kepercayaan pada pola ARN tertentu, Anda dapat mengganti bagian ARN tersebut dengan wildcard (\*). Setelah Anda membuat proyek Anda, Anda kemudian dapat memperbarui kebijakan kepercayaan.

put-role-policy.json:

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Sid": "CloudWatchLogsPolicy",
 "Effect": "Allow",
 "Action": [
 "logs:CreateLogGroup",
 "logs:CreateLogStream",
 "logs:PutLogEvents"
],
 "Resource": "*"
 },
 {
 "Sid": "CodeCommitPolicy",
```

```
"Effect": "Allow",
 "Action": [
 "codecommit:GitPull"
],
 "Resource": "*"
},
{
 "Sid": "S3GetObjectPolicy",
 "Effect": "Allow",
 "Action": [
 "s3:GetObject",
 "s3:GetObjectVersion"
],
 "Resource": "*"
},
{
 "Sid": "S3PutObjectPolicy",
 "Effect": "Allow",
 "Action": [
 "s3:PutObject"
],
 "Resource": "*"
},
{
 "Sid": "S3BucketIdentity",
 "Effect": "Allow",
 "Action": [
 "s3:GetBucketAcl",
 "s3:GetBucketLocation"
],
 "Resource": "*"
}
]
}
```

#### Note

Kebijakan ini berisi pernyataan yang memungkinkan akses ke sejumlah besar AWS sumber daya. Untuk membatasi AWS CodeBuild untuk mengakses AWS sumber daya tertentu, mengubah nilai `Resource` array. Untuk informasi lebih lanjut, lihat dokumentasi keamanan untuk AWS layanan ini.

3. Beralih ke direktori tempat Anda menyimpan file sebelumnya, dan kemudian jalankan dua perintah berikut, satu per satu, dalam urutan ini. Anda dapat menggunakan nilai yang berbeda untuk `CodeBuildServiceRole` dan `CodeBuildServiceRolePolicy`, tetapi pastikan untuk menggunakannya di sini.

```
aws iam create-role --role-name CodeBuildServiceRole --assume-role-policy-document
file://create-role.json
```

```
aws iam put-role-policy --role-name CodeBuildServiceRole --policy-name
CodeBuildServiceRolePolicy --policy-document file://put-role-policy.json
```

## Membuat dan mengkonfigurasi kunci yang dikelola pelanggan CodeBuild

AWS CodeBuild Untuk mengenkripsi artefak keluaran build-nya, diperlukan akses ke kunci KMS. Secara default, CodeBuild gunakan Kunci yang dikelola AWS untuk Amazon S3 di AWS akun Anda.

Jika Anda tidak ingin menggunakan Kunci yang dikelola AWS, Anda harus membuat dan mengkonfigurasi kunci dikelola pelanggan sendiri. Bagian ini menjelaskan cara melakukan ini dengan konsol IAM.

Untuk informasi tentang kunci yang dikelola pelanggan, lihat [AWS Key Management Service Konsep](#) dan [Membuat Kunci](#) di Panduan AWS KMS Pengembang.

Untuk mengonfigurasi kunci yang dikelola pelanggan untuk digunakan CodeBuild, ikuti petunjuk di bagian "Cara Memodifikasi Kebijakan Kunci" pada [Memodifikasi Kebijakan Kunci](#) dalam Panduan AWS KMS Pengembang. Kemudian tambahkan pernyataan berikut (antara **### BEGIN ADDING STATEMENTS HERE ###** dan **### END ADDING STATEMENTS HERE ###**) ke kebijakan kunci. Elipses (. . .) digunakan untuk singkatnya dan untuk membantu Anda menemukan tempat untuk menambahkan pernyataan. Jangan menghapus pernyataan apa pun, dan jangan menyetikkan elips ini ke dalam kebijakan kunci.

```
{
 "Version": "2012-10-17",
 "Id": "...",
 "Statement": [
 ### BEGIN ADDING STATEMENTS HERE ###
 {
 "Sid": "Allow access through Amazon S3 for all principals in the account that are
authorized to use Amazon S3",
```

```
"Effect": "Allow",
"Principal": {
 "AWS": "*"
},
"Action": [
 "kms:Encrypt",
 "kms:Decrypt",
 "kms:ReEncrypt*",
 "kms:GenerateDataKey*",
 "kms:DescribeKey"
],
"Resource": "*",
"Condition": {
 "StringEquals": {
 "kms:ViaService": "s3.region-ID.amazonaws.com",
 "kms:CallerAccount": "account-ID"
 }
}
},
{
 "Effect": "Allow",
 "Principal": {
 "AWS": "arn:aws:iam::account-ID:role/CodeBuild-service-role"
 },
 "Action": [
 "kms:Encrypt",
 "kms:Decrypt",
 "kms:ReEncrypt*",
 "kms:GenerateDataKey*",
 "kms:DescribeKey"
],
 "Resource": "*"
},
END ADDING STATEMENTS HERE
{
 "Sid": "Enable IAM User Permissions",
 ...
},
{
 "Sid": "Allow access for Key Administrators",
 ...
},
{
 "Sid": "Allow use of the key",
```

```
 ...
 },
 {
 "Sid": "Allow attachment of persistent resources",
 ...
 }
]
}
```

- **Region-ID** mewakili IDAWS wilayah tempat bucket Amazon S3 yang terkait dengan CodeBuild berada (misalnya,us-east-1).
- **Account-ID** mewakili ID dariAWS akun yang memiliki kunci yang dikelola pelanggan.
- **CodeBuild-service-role** mewakili nama peran CodeBuild layanan yang Anda buat atau identifikasi sebelumnya dalam topik ini.

#### Note

Untuk membuat atau mengonfigurasi kunci terkelola pelanggan melalui konsol IAM, Anda harus masuk terlebih dahulu keAWS Management Console dengan menggunakan salah satu langkah berikut:

- AkunAWS root Anda. Ini tidak disarankan. Untuk informasi selengkapnya, lihat [Pengguna Root Akun](#) di Panduan Pengguna.
- Pengguna administrator diAWS akun Anda. Untuk informasi selengkapnya, lihat [Membuat penggunaAkun AWS root Pertama Anda dan Grup](#) di Panduan Pengguna.
- Pengguna diAWS akun Anda dengan izin untuk membuat atau memodifikasi kunci yang dikelola pelanggan. Untuk informasi selengkapnya, lihat [Izin yang Diperlukan untuk MenggunakanAWS KMS Konsol](#) di PanduanAWS KMS Developer.

## Instal dan konfigurasiAWS CLI

Untuk mengaksesAWS CodeBuild, Anda dapat menggunakan CodeBuild konsol, CodePipeline konsol, atauAWS SDK.AWS CLI Untuk menginstal dan mengkonfigurasiAWS CLI, lihat [Mengatur denganAWS Command Line Interface diAWS Command Line Interface](#) User Guide.

1. Jalankan perintah berikut untuk mengonfirmasi apakah instalasi Anda AWS CLI mendukung CodeBuild:

```
aws codebuild list-builds
```

Jika berhasil, informasi yang serupa dengan yang berikut akan muncul di output:

```
{
 "ids": []
}
```

Kurung persegi kosong menunjukkan bahwa Anda belum menjalankan build apa pun.

2. Jika kesalahan adalah output, Anda harus menghapus versi Anda saat ini AWS CLI dan kemudian menginstal versi terbaru. Untuk informasi selengkapnya, lihat [Menghapus AWS CLI dan Menginstal AWS Command Line Interface](#) di Panduan AWS Command Line Interface Pengguna.

## Referensi baris perintah untuk AWS CodeBuild

Parameter AWS CLI menyediakan perintah untuk mengotomatisasi AWS CodeBuild. Gunakan informasi dalam topik ini sebagai suplemen untuk [AWS Command Line Interface Panduan Pengguna](#) dan [AWS CLI Referensi untuk AWS CodeBuild](#).

Bukan apa yang kau cari? Jika Anda ingin menggunakan AWS SDK untuk memanggil CodeBuild, lihat [AWS Referensi SDK dan alat](#).

Untuk menggunakan informasi dalam topik ini, Anda seharusnya sudah menginstal AWS CLI dan dikonfigurasi untuk digunakan dengan CodeBuild, seperti yang dijelaskan dalam [Instal dan konfigurasi AWS CLI](#).

Untuk menggunakan AWS CLI untuk menentukan endpoint untuk CodeBuild, lihat [Tentukan AWS CodeBuild endpoint \(AWS CLI\)](#).

Jalankan perintah ini untuk mendapatkan daftar perintah CodeBuild.

```
aws codebuild help
```

Jalankan perintah ini untuk mendapatkan informasi tentang perintah CodeBuild, di mana *perintah-nama* adalah nama perintah.

```
aws codebuild command-name help
```

Perintah CodeBuild meliputi:

- `batch-delete-builds`: Menghapus satu atau lebih build di CodeBuild. Untuk informasi selengkapnya, lihat [Menghapus build \(AWS CLI\)](#).
- `batch-get-builds`: Mendapat informasi tentang beberapa membangun di CodeBuild. Untuk informasi selengkapnya, lihat [Lihat detail build \(AWS CLI\)](#).
- `batch-get-projects`: Mendapat informasi tentang satu atau lebih proyek pembangunan yang ditentukan. Untuk informasi selengkapnya, lihat [Lihat detail proyek build \(AWS CLI\)](#).
- `create-project`: Membuat proyek pembangunan. Untuk informasi selengkapnya, lihat [Buat proyek build \(AWS CLI\)](#).
- `delete-project`: Menghapus proyek pembangunan. Untuk informasi selengkapnya, lihat [Menghapus proyek pembangunan \(AWS CLI\)](#).
- `list-builds`: Mendaftarkan Amazon Resource Names (ARN) untuk pembangunan di CodeBuild. Untuk informasi selengkapnya, lihat [Melihat daftar ID build \(AWS CLI\)](#).
- `list-builds-for-project`: Mendapat daftar ID build yang terkait dengan proyek build yang ditentukan. Untuk informasi selengkapnya, lihat [Melihat daftar ID build untuk proyek build \(AWS CLI\)](#).
- `list-curated-environment-images`: Mendapat daftar gambar Docker yang dikelola oleh CodeBuild yang dapat Anda gunakan untuk build Anda. Untuk informasi selengkapnya, lihat [Gambar Docker disediakan oleh CodeBuild](#).
- `list-projects`: Mendapat daftar nama proyek build. Untuk informasi selengkapnya, lihat [Melihat daftar nama proyek build \(AWS CLI\)](#).
- `start-build`: Mulai menjalankan membangun. Untuk informasi selengkapnya, lihat [Jalankan build \(AWS CLI\)](#).
- `stop-build`: Upaya untuk menghentikan build yang ditentukan agar tidak berjalan. Untuk informasi selengkapnya, lihat [Menghentikan membangun \(AWS CLI\)](#).
- `update-project`: Mengubah informasi tentang proyek pembangunan yang ditentukan. Untuk informasi selengkapnya, lihat [Mengubah setelan proyek pembangunan \(AWS CLI\)](#).

## AWS Referensi SDK dan alat AWS CodeBuild

Untuk menggunakan salah satu AWS SDK atau alat untuk mengotomatisasi AWS CodeBuild, lihat sumber daya berikut.

Jika Anda ingin menggunakan AWS CLI untuk menjalankan CodeBuild, lihat [Referensi baris perintah](#).

## Didukung AWS SDK dan alat untuk AWS CodeBuild

Berikut AWS SDK dan alat mendukung CodeBuild:

- Parameter [AWS SDK for C++](#). Untuk informasi selengkapnya, lihat [AWS::CodeBuild](#) bagian namespace dari AWS Referensi API SDK for C++.
- Parameter [AWS SDK for Go](#). Untuk informasi selengkapnya, lihat [CodeBuild](#) bagian AWS Referensi API SDK for Go.
- Parameter [AWS SDK for Java](#). Untuk informasi selengkapnya, lihat `com.amazonaws.services.codebuild` dan `com.amazonaws.services.codebuild.model` bagian [API SDK for Java](#).
- Parameter [AWS SDK for JavaScript di browser](#) dan [AWS SDK for JavaScript in Node.js Node.js](#). Untuk informasi selengkapnya, lihat [Kelas: AWSCodeBuild](#) bagian AWS Referensi API SDK for JavaScript.
- Parameter [AWS SDK for .NET](#). Untuk informasi selengkapnya, lihat [Amazon.codebuild](#) dan [Amazon.codebuild.Model](#) bagian namespace dari AWS Referensi API SDK for .NET.
- Parameter [AWS SDK for PHP](#). Untuk informasi selengkapnya, lihat [Namespace AWS\CodeBuild](#) bagian AWS Referensi SDK for PHP API.
- Parameter [AWS SDK for Python \(Boto3\)](#). Untuk informasi selengkapnya, lihat [CodeBuild](#) bagian Dokumentasi Boto 3.
- Parameter [AWS SDK for Ruby](#). Untuk informasi selengkapnya, lihat [Modul: AWS::CodeBuild](#) bagian AWS Referensi API SDK for Ruby.
- Parameter [AWS Tools for PowerShell](#). Untuk informasi selengkapnya, lihat [AWS CodeBuild](#) bagian AWS Referensi Tools for PowerShell Cmdlet.



## Tentukan AWS CodeBuild titik akhir

Anda dapat menggunakan AWS Command Line Interface (AWS CLI) atau salah satu AWS SDK untuk menentukan titik akhir yang digunakan oleh AWS CodeBuild. Ada titik akhir untuk setiap wilayah di mana CodeBuild tersedia. Selain endpoint regional, empat wilayah juga memiliki titik akhir Federal Information Processing Standards (FIPS). Untuk informasi selengkapnya tentang titik akhir FIPS, lihat [Gambaran Umum FIPS 140-2](#).

Menentukan titik akhir adalah opsional. Jika Anda tidak secara eksplisit memberi tahu titik akhir CodeBuild mana yang akan digunakan, layanan menggunakan titik akhir yang terkait dengan wilayah yang digunakan AWS akun Anda. CodeBuild tidak pernah default ke endpoint FIPS. Jika Anda ingin menggunakan titik akhir FIPS, Anda harus CodeBuild mengaitkannya menggunakan salah satu metode berikut.

### Note

Anda dapat menggunakan alias atau nama wilayah untuk menentukan titik akhir menggunakan AWS SDK. Jika Anda menggunakan AWS CLI, maka Anda harus menggunakan nama endpoint lengkap.

Untuk titik akhir yang dapat digunakan CodeBuild, lihat [CodeBuild wilayah dan titik akhir](#).

### Topik

- [Tentukan AWS CodeBuild endpoint \(AWS CLI\)](#)
- [Tentukan AWS CodeBuild titik akhir \(AWS SDK\)](#)

## Tentukan AWS CodeBuild endpoint (AWS CLI)

Anda dapat menggunakan AWS CLI untuk menentukan titik akhir melalui AWS CodeBuild yang diakses dengan menggunakan `--endpoint-url` argumen dalam CodeBuild perintah apapun. Misalnya, jalankan perintah ini untuk mendapatkan daftar nama pembuatan proyek menggunakan titik akhir Federal Information Processing Standards (FIPS) di Wilayah US East (N. Virginia):

```
aws codebuild list-projects --endpoint-url https://codebuild-fips.us-east-1.amazonaws.com
```

Sertakan `https://` pada awal titik akhir.

`--endpoint-url` AWS CLI Argumen tersedia untuk semua AWS layanan. Untuk informasi selengkapnya tentang ini dan AWS CLI argumen lainnya, lihat [Referensi AWS CLI Perintah](#).

## Tentukan AWS CodeBuild titik akhir (AWSSDK)

Anda dapat menggunakan AWS SDK untuk menentukan titik akhir yang AWS CodeBuild digunakan untuk diakses. Meskipun contoh ini menggunakan [AWSSDK for Java](#), Anda dapat menentukan titik akhir dengan AWS SDK lainnya.

Gunakan `withEndpointConfiguration` metode saat membangun AWS CodeBuild klien. Berikut adalah format untuk digunakan:

```
AWSCodeBuild awsCodeBuild = AWSCodeBuildClientBuilder.standard().
 withEndpointConfiguration(new AwsClientBuilder.EndpointConfiguration("endpoint",
 "region")).
 withCredentials(new AWSStaticCredentialsProvider(sessionCredentials)).
 build();
```

Untuk selengkapnya `AWSCodeBuildClientBuilder`, lihat [Kelas AWSCodeBuildClientBuilder](#).

Kredensial yang digunakan `withCredentials` harus bertipe `AWSCredentialsProvider`. Untuk informasi selengkapnya, lihat [Bekerja dengan AWS kredensial](#).

Jangan sertakan `https://` pada awal titik akhir.

Jika Anda ingin menentukan titik akhir non-FIPS, Anda dapat menggunakan wilayah alih-alih titik akhir yang sebenarnya. Misalnya, untuk menentukan titik akhir di wilayah AS Timur (Virginia N.), Anda dapat menggunakan `us-east-1` alih-alih nama endpoint lengkap, `codebuild.us-east-1.amazonaws.com`.

Jika Anda ingin menentukan titik akhir FIPS, Anda dapat menggunakan alias untuk menyederhanakan kode Anda. Hanya endpoint FIPS yang memiliki alias. Titik akhir lainnya harus ditentukan menggunakan wilayah atau nama lengkapnya.

Tabel berikut mencantumkan alias untuk masing-masing dari empat titik akhir FIPS yang tersedia:

| Nama wilayah                  | Wilayah   | Titik Akhir                            | Alias             |
|-------------------------------|-----------|----------------------------------------|-------------------|
| US East (Northern Virginia)   | as-east-1 | codebuild-fips.us-east-1.amazonaws.com | us-us-east-1-fips |
| Timur AS (Ohio)               | us-east-2 | codebuild-fips.us-east-2.amazonaws.com | us-us-east-2-fips |
| US West (Northern California) | us-west-1 | codebuild-fips.us-west-1.amazonaws.com | us-west-1-fips    |
| US West (Oregon)              | us-west-2 | codebuild-fips.us-west-2.amazonaws.com | us-west-2-fips    |

Untuk menentukan penggunaan titik akhir FIPS di wilayah US West (Oregon) menggunakan alias:

```
AWSCodeBuild awsCodeBuild = AWSCodeBuildClientBuilder.standard().
 withEndpointConfiguration(new AwsClientBuilder.EndpointConfiguration("us-west-2-
fips", "us-west-2")).
 withCredentials(new AWSStaticCredentialsProvider(sessionCredentials)).
 build();
```

Untuk menentukan penggunaan titik akhir non-FIPS di wilayah US East (N. Virginia):

```
AWSCodeBuild awsCodeBuild = AWSCodeBuildClientBuilder.standard().
 withEndpointConfiguration(new AwsClientBuilder.EndpointConfiguration("us-east-1",
"us-east-1")).
 withCredentials(new AWSStaticCredentialsProvider(sessionCredentials)).
 build();
```

Untuk menentukan penggunaan titik akhir non-FIPS di wilayah Asia Pasifik (Mumbai):

```

AWSCodeBuild awsCodeBuild = AWSCodeBuildClientBuilder.standard().
 withEndpointConfiguration(new AwsClientBuilder.EndpointConfiguration("ap-south-1",
"ap-south-1")).
 withCredentials(new AWSStaticCredentialsProvider(sessionCredentials)).
 build();

```

## Gunakan AWS CodePipeline dengan AWS CodeBuild untuk menguji kode dan menjalankan build

Anda dapat mengotomatiskan proses rilis Anda dengan menggunakan AWS CodePipeline untuk menguji kode Anda dan menjalankan build Anda dengan AWS CodeBuild

Tabel berikut mencantumkan tugas dan metode yang tersedia untuk melaksanakannya.

Menggunakan AWS SDK untuk menyelesaikan tugas-tugas ini berada di luar cakupan topik ini.

| Tugas                                                                                                      | Pendekatan yang tersedia                                                                                | Pendekatan yang dijelaskan dalam topik ini                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Buat pipeline pengiriman berkelanjutan (CD) dengan CodePipeline itu mengotomatiskan build dengan CodeBuild | <ul style="list-style-type: none"> <li>CodePipeline konsol</li> <li>AWS CLI</li> <li>SDK AWS</li> </ul> | <ul style="list-style-type: none"> <li><a href="#">Gunakan CodePipeline konsol</a></li> <li><a href="#">Gunakan AWS CLI</a></li> <li>Anda dapat menyesuaikan informasi dalam topik ini untuk menggunakan AWS SDK. Untuk informasi selengkapnya, lihat dokumentasi <code>create-pipeline</code> tindakan untuk bahasa pemrograman Anda di bagian <a href="#">SDK Tools for Amazon Web Services</a> atau lihat <a href="#">CreatePipeline</a> di Referensi AWS CodePipeline API.</li> </ul> |
| Tambahkan pengujian dan bangun otomatisasi CodeBuild dengan pipeline                                       | <ul style="list-style-type: none"> <li>CodePipeline konsol</li> <li>AWS CLI</li> <li>SDK AWS</li> </ul> | <ul style="list-style-type: none"> <li><a href="#">Gunakan CodePipeline konsol untuk menambahkan otomatisasi build</a></li> <li><a href="#">Gunakan CodePipeline konsol untuk menambahkan otomatisasi pengujian</a></li> <li>Untuk itu AWS CLI, Anda dapat mengadaptasi informasi dalam topik ini untuk membuat pipeline yang berisi</li> </ul>                                                                                                                                           |

| Tugas                    | Pendekatan yang tersedia | Pendekatan yang dijelaskan dalam topik ini                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|--------------------------|--------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| yang ada di CodePipeline |                          | <p>tindakan CodeBuild build atau tindakan pengujian. Untuk informasi selengkapnya, lihat <a href="#">Mengedit pipeline (AWS CLI)</a> dan <a href="#">referensi struktur CodePipeline pipa</a> di Panduan AWS CodePipeline Pengguna.</p> <ul style="list-style-type: none"> <li>Anda dapat menyesuaikan informasi dalam topik ini untuk menggunakan AWS SDK. Untuk informasi selengkapnya, lihat dokumentasi <code>update-pipeline</code> tindakan untuk bahasa pemrograman Anda melalui bagian <a href="#">SDK Tools for Amazon Web Services</a> atau lihat <a href="#">UpdatePipeline</a> di Referensi AWS CodePipeline API.</li> </ul> |

## Prasyarat

- Jawab pertanyaan di [Rencanakan pembangunan](#).
- Jika Anda menggunakan pengguna untuk mengakses CodePipeline alih-alih akun AWS root atau pengguna administrator, lampirkan kebijakan terkelola yang diberi nama `AWSCodePipelineFullAccess` ke pengguna (atau ke grup IAM tempat pengguna tersebut berada). Menggunakan akun AWS root tidak disarankan. Kebijakan ini memberikan izin kepada pengguna untuk membuat pipeline. CodePipeline Untuk informasi selengkapnya, lihat [Melampirkan kebijakan terkelola](#) di Panduan pengguna.

### Note

Entitas IAM yang melampirkan kebijakan ke pengguna (atau grup IAM yang menjadi milik pengguna) harus memiliki izin di IAM untuk melampirkan kebijakan. Untuk informasi selengkapnya, lihat [Mendelegasikan izin untuk mengelola pengguna, grup, dan kredensial IAM](#) di Panduan pengguna.


- Buat peran CodePipeline layanan, jika Anda belum memilikinya tersedia di AWS akun Anda. CodePipeline menggunakan peran layanan ini untuk berinteraksi dengan AWS layanan lain, termasuk AWS CodeBuild, atas nama Anda. Misalnya, untuk menggunakan peran AWS CLI untuk membuat CodePipeline layanan, jalankan `create-role` perintah IAM:

Untuk Linux, macOS, atau Unix:

```
aws iam create-role --role-name AWS-CodePipeline-CodeBuild-Service-Role
--assume-role-policy-document '{"Version":"2012-10-17","Statement":
{"Effect":"Allow","Principal":
{"Service":"codepipeline.amazonaws.com"},"Action":"sts:AssumeRole"}'}
```


Untuk Windows:

```
aws iam create-role --role-name AWS-CodePipeline-CodeBuild-Service-Role --assume-
role-policy-document "{\"Version\":\"2012-10-17\",\"Statement\":{\"Effect\":
\"Allow\",\"Principal\":{\"Service\":\"codepipeline.amazonaws.com\"},\"Action\":
\"sts:AssumeRole\"}}"
```

 Note

Entitas IAM yang membuat peran CodePipeline layanan ini harus memiliki izin di IAM untuk membuat peran layanan.

- Setelah membuat peran CodePipeline layanan atau mengidentifikasi peran yang sudah ada, Anda harus menambahkan kebijakan peran CodePipeline layanan default ke peran layanan seperti yang dijelaskan dalam [Meninjau kebijakan peran CodePipeline layanan default](#) di Panduan AWS CodePipeline Pengguna, jika belum menjadi bagian dari kebijakan untuk peran tersebut.

 Note

Entitas IAM yang menambahkan kebijakan peran CodePipeline layanan ini harus memiliki izin di IAM untuk menambahkan kebijakan peran layanan ke peran layanan.

- Buat dan unggah kode sumber ke jenis repositori yang didukung oleh CodeBuild dan CodePipeline, seperti, Amazon S3 CodeCommit, Bitbucket, atau GitHub. Kode sumber harus berisi file buildspec, tetapi Anda dapat mendeklarasikannya saat menentukan proyek build nanti dalam topik ini. Lihat informasi yang lebih lengkap di [Referensi Buildspec](#).

**⚠ Important**

Jika Anda berencana menggunakan pipeline untuk menerapkan kode sumber bawaan, artefak keluaran build harus kompatibel dengan sistem penerapan yang Anda gunakan.

- Untuk AWS OpsWorks, lihat [Sumber aplikasi](#) dan [Menggunakan CodePipeline dengan AWS OpsWorks](#) di Panduan AWS OpsWorks Pengguna.

**Topik**

- [Membuat pipeline yang menggunakan CodeBuild \(CodePipeline konsol\)](#)
- [Buat pipeline yang menggunakan CodeBuild \(AWS CLI\)](#)
- [Menambahkan aksi CodeBuild build ke pipeline \(CodePipeline konsol\)](#)
- [Menambahkan tindakan CodeBuild pengujian ke pipeline \(CodePipeline konsol\)](#)

**Membuat pipeline yang menggunakan CodeBuild (CodePipeline konsol)**

Gunakan prosedur berikut untuk membuat alur yang digunakan CodeBuild untuk membangun dan menerapkan kode sumber Anda.

Untuk membuat pipeline yang hanya menguji kode sumber Anda:

- Gunakan prosedur berikut untuk membuat pipeline, lalu hapus tahap Build dan Beta dari pipeline. Kemudian gunakan [Menambahkan tindakan CodeBuild pengujian ke pipeline \(CodePipeline konsol\)](#) prosedur dalam topik ini untuk menambahkan ke pipeline tindakan pengujian yang menggunakan CodeBuild.
- Gunakan salah satu prosedur lain dalam topik ini untuk membuat pipeline, dan kemudian gunakan [Menambahkan tindakan CodeBuild pengujian ke pipeline \(CodePipeline konsol\)](#) prosedur dalam topik ini untuk menambahkan ke pipeline tindakan pengujian yang menggunakan CodeBuild.

Untuk menggunakan wizard create pipeline CodePipeline untuk membuat pipeline yang menggunakan CodeBuild

1. Masuk ke AWS Management Console dengan menggunakan:

- Akun AWS root Anda. Ini tidak disarankan. Untuk informasi selengkapnya, lihat [Pengguna root akun](#) di Panduan Pengguna.
- Pengguna administrator di AWS akun Anda. Untuk informasi selengkapnya, lihat [Membuat pengguna dan grup Akun AWS root pertama Anda](#) di Panduan Pengguna.
- Pengguna di AWS akun Anda dengan izin untuk menggunakan serangkaian tindakan minimum berikut:

```
codepipeline:*
iam:ListRoles
iam:PassRole
s3:CreateBucket
s3:GetBucketPolicy
s3:GetObject
s3:ListAllMyBuckets
s3:ListBucket
s3:PutBucketPolicy
codecommit:ListBranches
codecommit:ListRepositories
codedeploy:GetApplication
codedeploy:GetDeploymentGroup
codedeploy:ListApplications
codedeploy:ListDeploymentGroups
elasticbeanstalk:DescribeApplications
elasticbeanstalk:DescribeEnvironments
lambda:GetFunctionConfiguration
lambda:ListFunctions
opsworks:DescribeStacks
opsworks:DescribeApps
opsworks:DescribeLayers
```

2. Buka AWS CodePipeline konsol di <https://console.aws.amazon.com/codesuite/codepipeline/home>.
3. Di pemilih AWS Wilayah, pilih AWS Wilayah tempat AWS sumber daya proyek build Anda berada. Ini harus menjadi AWS Wilayah di CodeBuild mana didukung. Untuk informasi lain, lihat [AWS CodeBuild](#) di Referensi Umum Amazon Web Services.
4. Buat pipeline. Jika halaman CodePipeline informasi ditampilkan, pilih Buat pipeline. Jika halaman Pipelines ditampilkan, pilih Buat pipeline.



5. Pada Langkah 1: Pilih halaman alur, untuk nama alur, masukkan nama untuk alur (misalnya, **CodeBuildDemoPipeline**). Jika Anda memilih nama yang berbeda, pastikan untuk menggunakannya di seluruh prosedur ini.

6. Untuk Nama peran, lakukan salah satu hal berikut:

Pilih Peran layanan baru, dan di Nama Peran, masukkan nama untuk peran layanan baru Anda.

Pilih Peran layanan yang ada, lalu pilih peranCodePipeline layanan yang Anda buat atau identifikasi sebagai bagian dari prasyarat topik ini.

7. Untuk Penyimpanan artifact, lakukan salah satu hal berikut:

- Pilih Lokasi default untuk menggunakan penyimpanan artifact default, seperti bucket artifact S3 yang ditetapkan sebagai default, untuk alur Anda diAWS Wilayah yang Anda pilih untuk alur Anda.
- Pilih Lokasi khusus jika Anda sudah memiliki toko artefak yang sudah ada yang telah Anda buat, seperti bucket artefak S3, diAWS Wilayah yang sama dengan pipeline Anda.

#### Note

Ini bukan bucket sumber untuk kode sumber Anda. Ini adalah penyimpanan artifact untuk alur Anda. Penyimpanan artifact terpisah, seperti bucket S3, diperlukan untuk setiap alur, diAWS Wilayah yang sama dengan alur.

8. Pilih Selanjutnya.

9. Pada Langkah 2: Tambahkan halaman tahap sumber, untuk penyedia sumber, lakukan salah satu dari berikut ini:

- Jika kode sumber Anda disimpan dalam bucket S3, pilih Amazon S3. Untuk Bucket, pilih bucket S3 yang berisi kode sumber Anda. Untuk kunci objek S3, masukkan nama file berisi kode sumber (misalnya, *file-name.zip*). Pilih Selanjutnya.
- Jika kode sumber Anda disimpan dalamAWS CodeCommit repositori, pilih CodeCommit. Untuk Nama repositori, pilih nama repositori yang berisi kode sumber. Untuk Nama cabang, pilih nama cabang yang berisi versi kode sumber yang ingin Anda bangun. Pilih Selanjutnya.
- Jika kode sumber Anda disimpan dalamGitHub repositori, pilih GitHub. Pilih Connect keGitHub, dan ikuti petunjuk untuk mengautentikasiGitHub. Untuk Repositori, pilih nama

repositori yang berisi kode sumber. Untuk Cabang, pilih nama cabang yang berisi versi kode sumber yang ingin Anda bangun.

Pilih Selanjutnya.

10. Pada Langkah 3: Tambahkan halaman tahap build, untuk penyedia Build, pilih CodeBuild.
11. Jika Anda sudah memiliki proyek pembangunan yang ingin Anda gunakan, untuk nama Project, pilih nama proyek build dan lewati ke langkah berikutnya dalam prosedur ini.

Jika Anda perlu membuat proyekCodeBuild build baru, ikuti instruksi[Buat proyek build \(konsol\)](#) dan kembali ke prosedur ini.

Jika Anda memilih proyek pembangunan yang ada, itu harus memiliki pengaturan artefak keluaran build yang sudah ditentukan (meskipunCodePipeline menyimpannya). Untuk informasi selengkapnya, lihat [Mengubah setelan proyek build \(konsol\)](#).

#### Important


Jika Anda mengaktifkan webhook untukCodeBuild proyek, dan proyek digunakan sebagai langkah pembangunanCodePipeline, maka dua build identik dibuat untuk setiap komit. Satu build dipicu melalui webhook, dan satu melaluiCodePipeline. Karena penagihan berdasarkan per-build, Anda akan ditagih untuk kedua build. Oleh karena itu, jika Anda menggunakanCodePipeline, kami sarankan Anda menonaktifkan webhook diCodeBuild. DiAWS CodeBuild konsol, kosongkan kotak Webhook. Untuk informasi selengkapnya, lihat [Mengubah setelan proyek build \(konsol\)](#).

12. Pada Langkah 4: Tambahkan halaman tahap, lakukan salah satu hal berikut:
  - Jika Anda tidak ingin menyebarkan artefak keluaran build, pilih Lewati, dan konfirmasi pilihan ini saat diminta.
  - Jika Anda ingin menerapkan artefak keluaran build, untuk Penyedia Deploy, pilih penyedia penyebaran, lalu tentukan pengaturan saat diminta.

Pilih Selanjutnya.

13. Pada halaman Tinjau, tinjau pilihan Anda, lalu pilih Buat pipeline.

14. Setelah pipeline berjalan dengan sukses, Anda bisa mendapatkan artefak keluaran build. Dengan pipeline yang ditampilkan di CodePipeline konsol, dalam aksi Build, pilih tooltip. Buat catatan nilai untuk artefak Output (misalnya, MyAppBuild).

 Note

Anda juga bisa mendapatkan artefak keluaran build dengan memilih tautan Build artefak pada halaman detail build di CodeBuild konsol. Untuk sampai ke halaman ini, lewati langkah-langkah lainnya dalam prosedur ini, dan lihat [Lihat detail build \(konsol\)](#).

15. Buka konsol Amazon S3 di <https://console.aws.amazon.com/s3/>.
16. Di daftar bucket, buka bucket yang digunakan oleh alur. Nama bucket harus mengikuti formatnya `codepipeline-region-ID-random-number`. Anda dapat menggunakan `CodePipeline get-pipeline` perintah AWS CLI untuk menjalankan untuk mendapatkan nama bucket, di `my-pipeline-name` mana nama tampilan pipeline Anda:

```
aws codepipeline get-pipeline --name my-pipeline-name
```

Dalam output, `pipeline` objek berisi `artifactStore` objek, yang berisi `location` nilai dengan nama ember.

17. Buka folder yang cocok dengan nama pipeline Anda (tergantung pada panjang nama pipeline, nama folder mungkin dipotong), dan kemudian buka folder yang cocok dengan nilai untuk artefak Output yang Anda catat sebelumnya.
18. Ekstrak isi file tersebut. Jika ada beberapa file di folder itu, ekstrak isi file dengan stempel waktu terakhir Modifikasi terbaru. (Anda mungkin perlu memberikan file `.zip` ekstensi sehingga Anda dapat bekerja dengannya di utilitas ZIP sistem Anda.) Artefak keluaran build ada di isi file yang diekstrak.
19. Jika Anda menginstruksikan CodePipeline untuk menyebarkan artefak keluaran build, gunakan instruksi penyedia penyebaran untuk membuka artefak keluaran build pada target penerapan.

## Buat pipeline yang menggunakan CodeBuild (AWS CLI)

Gunakan prosedur berikut untuk membuat pipeline yang digunakan CodeBuild untuk pembangunan kode sumber Anda.

Untuk menggunakan AWS CLI untuk membuat pipeline yang menerapkan kode sumber bawaan Anda atau yang hanya menguji kode sumber Anda, Anda dapat menyesuaikan instruksi di [Edit pipeline \(AWS CLI\)](#) dan [referensi struktur CodePipeline pipeline](#) di Panduan AWS CodePipeline Pengguna.

1. Membuat proyek pembangunan di CodeBuild. Untuk informasi selengkapnya, lihat [Membuat proyek pembangunan](#).

**⚠ Important**

Proyek build harus menentukan pengaturan artefak keluaran build (meskipun CodePipeline menyimpannya). Untuk informasi lebih lanjut, lihat deskripsi artifacts di [Buat proyek build \(AWS CLI\)](#).

2. Pastikan Anda telah mengkonfigurasi AWS CLI dengan kunci AWS akses dan kunci akses AWS rahasia yang sesuai dengan salah satu entitas IAM yang dijelaskan dalam topik ini. Untuk informasi selengkapnya, lihat [Mengatur AWS Command Line Interface di Panduan AWS Command Line Interface Pengguna](#).
3. Buat file berformat JSON yang mewakili struktur pipeline. Beri nama file `create-pipeline.json` atau serupa. Misalnya, struktur berformat JSON ini membuat pipeline dengan aksi sumber yang mereferensikan bucket input S3 dan aksi build yang menggunakan CodeBuild:

```
{
 "pipeline": {
 "roleArn": "arn:aws:iam::<account-id>:role/<AWS-CodePipeline-service-role-name>",
 "stages": [
 {
 "name": "Source",
 "actions": [
 {
 "inputArtifacts": [],
 "name": "Source",
 "actionTypeId": {
 "category": "Source",
 "owner": "AWS",
 "version": "1",
 "provider": "S3"
 },
 "outputArtifacts": [
 {
```

```
 "name": "MyApp"
 }
],
 "configuration": {
 "S3Bucket": "<bucket-name>",
 "S3ObjectKey": "<source-code-file-name.zip>"
 },
 "runOrder": 1
 }
]
},
{
 "name": "Build",
 "actions": [
 {
 "inputArtifacts": [
 {
 "name": "MyApp"
 }
],
 "name": "Build",
 "actionTypeId": {
 "category": "Build",
 "owner": "AWS",
 "version": "1",
 "provider": "CodeBuild"
 },
 "outputArtifacts": [
 {
 "name": "default"
 }
],
 "configuration": {
 "ProjectName": "<build-project-name>"
 },
 "runOrder": 1
 }
]
}
],
"artifactStore": {
 "type": "S3",
 "location": "<CodePipeline-internal-bucket-name>"
},
```

```
"name": "<my-pipeline-name>",
"version": 1
}
}
```

Dalam data berformat JSON ini:

- Nilai `roleArn` harus sesuai dengan ARN dari peran `CodePipeline` layanan yang Anda buat atau identifikasi sebagai bagian dari prasyarat.
- Nilai-nilai `S3Bucket` dan `S3ObjectKey` di `configuration` asumsikan kode sumber disimpan dalam bucket S3. Untuk pengaturan jenis repositori kode sumber lainnya, lihat [referensi struktur CodePipeline pipeline](#) di Panduan AWS CodePipeline Pengguna.
- Nilai `ProjectName` adalah nama proyek CodeBuild pembangunan yang Anda buat sebelumnya dalam prosedur ini.
- Nilai bucket S3 yang digunakan oleh pipeline ini. `location` Untuk informasi selengkapnya, lihat [Membuat kebijakan untuk Bucket S3 yang akan digunakan sebagai penyimpanan artefak CodePipeline](#) di Panduan AWS CodePipeline Pengguna.
- Nilai `name` adalah nama pipa ini. Semua nama pipa harus unik untuk akun Anda.

Meskipun data ini hanya menjelaskan tindakan sumber dan aksi build, Anda dapat menambahkan tindakan untuk aktivitas yang terkait dengan pengujian, menerapkan artefak keluaran build, memanggil AWS Lambda fungsi, dan banyak lagi. Untuk informasi selengkapnya, lihat [referensi struktur AWS CodePipeline pipeline](#) di Panduan AWS CodePipeline Pengguna.

4. Beralih ke folder yang berisi file JSON, dan kemudian jalankan `CodePipeline create-pipeline` perintah, menentukan nama file:

```
aws codepipeline create-pipeline --cli-input-json file://create-pipeline.json
```

#### Note

Anda harus membuat pipeline di AWS Wilayah yang CodeBuild didukung. Untuk informasi lain, lihat [AWS CodeBuild](#) di Referensi Umum Amazon Web Services.

Data berformat JSON muncul dalam output, dan `CodePipeline` membuat pipeline.

5. Untuk mendapatkan informasi tentang status pipeline, jalankan `CodePipeline` [get-pipeline-state](#) perintah, tentukan nama pipeline:

```
aws codepipeline get-pipeline-state --name <my-pipeline-name>
```

Dalam output, cari informasi yang menegaskan pembangunan berhasil. Ellipses ( . . . ) digunakan untuk menampilkan data yang telah dihilangkan untuk singkatnya.

```
{
 ...
 "stageStates": [
 ...
 {
 "actionStates": [
 {
 "actionName": "CodeBuild",
 "latestExecution": {
 "status": "SUCCEEDED",
 ...
 },
 ...
 }
]
 }
]
}
```

Jika Anda menjalankan perintah ini terlalu dini, Anda mungkin tidak melihat informasi apa pun tentang aksi build. Anda mungkin perlu menjalankan perintah ini beberapa kali hingga pipeline selesai menjalankan aksi build.

6. Setelah membangun sukses, ikuti petunjuk ini untuk mendapatkan artefak keluaran build. Buka konsol Amazon S3 di <https://console.aws.amazon.com/s3/>.

#### Note

Anda juga bisa mendapatkan artefak keluaran build dengan memilih tautan Build artefak pada halaman detail build terkait di CodeBuild konsol. Untuk sampai ke halaman ini, lewati langkah-langkah lainnya dalam prosedur ini, dan lihat [Lihat detail build \(konsol\)](#).

7. Dalam daftar bucket, buka bucket yang digunakan oleh pipa. Nama bucket harus mengikuti format `codepipeline-<region-ID>-<random-number>`. Anda bisa mendapatkan nama bucket dari `create-pipeline.json` file atau Anda dapat menjalankan `CodePipeline get-pipeline` perintah untuk mendapatkan nama bucket.

```
aws codepipeline get-pipeline --name <pipeline-name>
```

Dalam output, `pipeline` objek berisi `artifactStore` objek, yang berisi `location` nilai dengan nama ember.

8. Buka folder yang cocok dengan nama pipeline Anda (misalnya, *<pipeline-name>*).
9. Di folder itu, buka folder bernama `default`.
10. Ekstrak isi file tersebut. Jika ada beberapa file di folder itu, ekstrak isi file dengan stempel waktu terakhir Modifikasi terbaru. (Anda mungkin perlu memberikan file `.zip` ekstensi sehingga Anda dapat bekerja dengannya di utilitas ZIP sistem Anda.) Artefak keluaran build ada di isi file yang diekstrak.

## Menambahkan aksi CodeBuild build ke pipeline (CodePipeline konsol)

1. Masuk ke AWS Management Console dengan menggunakan:
  - Akun AWS root Anda. Ini tidak disarankan. Untuk informasi selengkapnya, lihat [Pengguna root akun](#) di Panduan Pengguna.
  - Pengguna administrator di AWS akun Anda. Untuk informasi selengkapnya, lihat [Membuat pengguna dan grup Akun AWS root pertama Anda](#) di Panduan Pengguna.
  - Pengguna di AWS akun Anda dengan izin untuk melakukan serangkaian tindakan minimum berikut:

```
codepipeline:*
iam:ListRoles
iam:PassRole
s3:CreateBucket
s3:GetBucketPolicy
s3:GetObject
s3:ListAllMyBuckets
s3:ListBucket
s3:PutBucketPolicy
codecommit:ListBranches
```



```
codecommit:ListRepositories
codedeploy:GetApplication
codedeploy:GetDeploymentGroup
codedeploy:ListApplications
codedeploy:ListDeploymentGroups
elasticbeanstalk:DescribeApplications
elasticbeanstalk:DescribeEnvironments
lambda:GetFunctionConfiguration
lambda:ListFunctions
opsworks:DescribeStacks
opsworks:DescribeApps
opsworks:DescribeLayers
```

2. Buka CodePipeline konsol di <https://console.aws.amazon.com/codesuite/codepipeline/home>.
3. Di pilih AWS wilayah, pilih AWS Wilayah tempat pipeline Anda berada. Ini harus menjadi Wilayah di CodeBuild mana didukung. Untuk informasi lain, lihat [CodeBuild](#) di Referensi Umum Amazon Web Services.
4. Pada halaman Pipelines, pilih nama pipa.
5. Pada halaman detail pipeline, dalam tindakan Sumber, pilih tooltip. Buat catatan nilai untuk artefak Output (misalnya, MyApp).

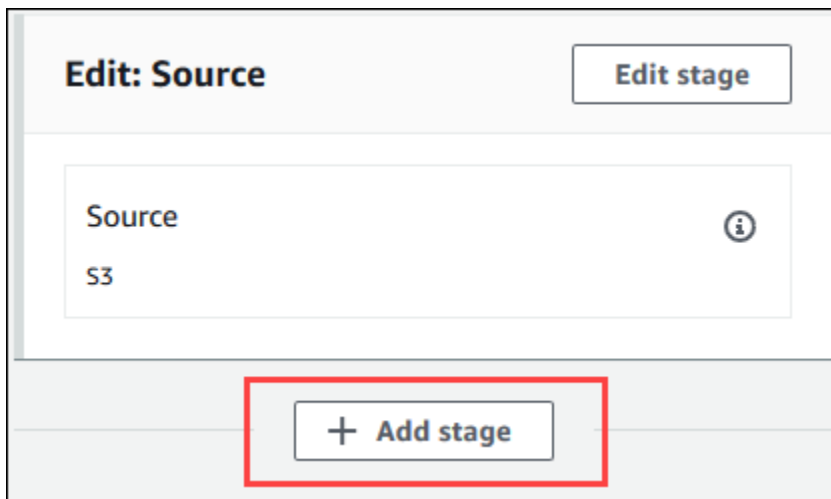
#### Note

Prosedur ini menunjukkan kepada Anda cara menambahkan aksi build dalam tahap build antara tahap Sumber dan Beta. Jika Anda ingin menambahkan aksi build di tempat lain, pilih tooltip pada tindakan tepat sebelum tempat Anda ingin menambahkan aksi build, dan catat nilai untuk artefak Output.

6. Pilih Edit.
7. Antara tahap Sumber dan Beta, pilih Tambahkan tahap.

#### Note

Prosedur ini menunjukkan cara menambahkan tahap pembangunan antara tahap Sumber dan Beta ke pipeline Anda. Untuk menambahkan aksi build ke tahap yang ada, pilih Edit stage di stage, lalu lewati ke langkah 8 dari prosedur ini. Untuk menambahkan tahap build di tempat lain, pilih Add stage di tempat yang diinginkan.



8. Untuk nama Stage, masukkan nama tahap build (misalnya, **Build**). Jika Anda memilih nama yang berbeda, gunakan di seluruh prosedur ini.
9. Di dalam tahap yang dipilih, pilih Tambahkan tindakan.

**Note**

Prosedur ini menunjukkan cara menambahkan build di dalam tahap build. Untuk menambahkan aksi build di tempat lain, pilih Tambahkan tindakan di tempat yang diinginkan. Anda mungkin harus terlebih dahulu memilih Edit stage pada tahap yang ada di mana Anda ingin menambahkan aksi build.

10. Di Edit tindakan, untuk nama Tindakan, masukkan nama untuk tindakan (misalnya, **CodeBuild**). Jika Anda memilih nama yang berbeda, gunakan di seluruh prosedur ini.
11. Untuk penyedia Action, pilih CodeBuild.
12. Jika Anda sudah memiliki proyek pembangunan yang ingin Anda gunakan, untuk nama Project, pilih nama proyek build dan lewati ke langkah berikutnya dalam prosedur ini.

Jika Anda perlu membuat proyek CodeBuild build baru, ikuti instruksi [Buat proyek build \(konsol\)](#) dan kembali ke prosedur ini.

Jika Anda memilih proyek pembangunan yang ada, itu harus memiliki pengaturan artefak keluaran build yang sudah ditentukan (meskipun CodePipeline menyimpannya). Untuk informasi lebih lanjut, lihat deskripsi Artefak di [Buat proyek build \(konsol\)](#) atau [Mengubah setelan proyek build \(konsol\)](#).

**⚠ Important**

Jika Anda mengaktifkan webhook untuk CodeBuild proyek, dan proyek digunakan sebagai langkah pembangunan CodePipeline, maka dua build identik dibuat untuk setiap komit. Satu build dipicu melalui webhook dan satu melalui CodePipeline. Karena penagihan berdasarkan per-build, Anda akan ditagih untuk kedua build. Oleh karena itu, jika Anda menggunakan CodePipeline, kami sarankan Anda menonaktifkan webhook di CodeBuild. Di CodeBuild konsol, kosongkan kotak Webhook. Untuk informasi selengkapnya, lihat [Mengubah setelan proyek build \(konsol\)](#)

13. Untuk artefak Input, pilih artefak keluaran yang Anda catat sebelumnya dalam prosedur ini.
14. Untuk artefak Output, masukkan nama untuk artefak keluaran (misalnya, **MyAppBuild**).
15. Pilih Tambahkan tindakan.
16. Pilih Simpan, lalu pilih Simpan untuk menyimpan perubahan Anda ke pipeline.
17. Pilih Rilis perubahan.
18. Setelah pipeline berjalan dengan sukses, Anda bisa mendapatkan artefak keluaran build. Dengan pipeline yang ditampilkan di CodePipeline konsol, dalam aksi Build, pilih tooltip. Buat catatan nilai untuk artefak Output (misalnya, MyAppBuild).

**ℹ Note**

Anda juga bisa mendapatkan artefak keluaran build dengan memilih tautan Build artefak pada halaman detail build di CodeBuild konsol. Untuk sampai ke halaman ini, lihat [Lihat detail build \(konsol\)](#), dan kemudian lewati ke langkah 31 dari prosedur ini.

19. Buka konsol Amazon S3 di <https://console.aws.amazon.com/s3/>.
20. Dalam daftar ember, buka ember yang digunakan oleh pipa. Nama bucket harus mengikuti formatnya `codepipeline-region-ID-random-number`. Anda dapat menggunakan AWS CLI untuk menjalankan `CodePipeline get-pipeline` perintah untuk mendapatkan nama bucket:

```
aws codepipeline get-pipeline --name my-pipeline-name
```

Dalam output, `pipeline` objek berisi `artifactStore` objek, yang berisi `location` nilai dengan nama ember.

21. Buka folder yang cocok dengan nama pipeline Anda (tergantung pada panjang nama pipeline, nama folder mungkin dipotong), dan kemudian buka folder yang cocok dengan nilai artefak Output yang Anda catat sebelumnya dalam prosedur ini.
22. Ekstrak isi file tersebut. Jika ada beberapa file di folder itu, ekstrak isi file dengan stempel waktu terakhir Modifikasi terbaru. (Anda mungkin perlu memberikan file .zip ekstensi sehingga Anda dapat bekerja dengannya di utilitas ZIP sistem Anda.) Artefak keluaran build ada di isi file yang diekstrak.
23. Jika Anda menginstruksikan CodePipeline untuk menyebarkan artefak keluaran build, gunakan instruksi penyedia penyebaran untuk membuka artefak keluaran build pada target penerapan.

## Menambahkan tindakan CodeBuild pengujian ke pipeline (CodePipelinekonsol)

1. Tindakan masuk AWS Management Console menggunakan:
  - Akun AWS root Anda. Ini tidak disarankan. Untuk informasi selengkapnya, lihat [Pengguna root akun](#) di Panduan Pengguna.
  - Pengguna menambahkan administrator di AWS akun Anda. Untuk informasi selengkapnya, lihat [Membuat pengguna dan grup Akun AWS root pertama Anda](#) di Panduan Pengguna.
  - Pengguna di AWS akun Anda dengan izin untuk melakukan serangkaian tindakan minimum berikut:

```
codepipeline:*
iam:ListRoles
iam:PassRole
s3:CreateBucket
s3:GetBucketPolicy
s3:GetObject
s3:ListAllMyBuckets
s3:ListBucket
s3:PutBucketPolicy
codecommit:ListBranches
codecommit:ListRepositories
codedeploy:GetApplication
codedeploy:GetDeploymentGroup
codedeploy:ListApplications
codedeploy:ListDeploymentGroups
elasticbeanstalk:DescribeApplications
```

```
elasticbeanstalk:DescribeEnvironments
lambda:GetFunctionConfiguration
lambda:ListFunctions
opsworks:DescribeStacks
opsworks:DescribeApps
opsworks:DescribeLayers
```

2. Buka CodePipeline konsol di <https://console.aws.amazon.com/codesuite/codepipeline/home>.
3. Di pemilih AWS wilayah, pilih AWS Wilayah tempat pipeline Anda berada. Ini harus menjadi AWS Wilayah di CodeBuild mana didukung. Untuk informasi lain, lihat [AWS CodeBuild](#) di Referensi Umum Amazon Web Services.
4. Pada halaman Pipelines, pilih nama pipa.
5. Pada halaman detail pipeline, dalam tindakan Sumber, pilih tooltip. Buat catatan nilai untuk artefak Output (misalnya, MyApp).

#### Note

Prosedur ini menunjukkan kepada Anda cara menambahkan tindakan pengujian di dalam tahap pengujian antara tahap Sumber dan Beta. Jika Anda ingin menambahkan tindakan pengujian di tempat lain, letakkan penunjuk mouse Anda pada tindakan sebelumnya, dan catat nilai untuk artefak Output.

6. Pilih Edit.
7. Segera setelah tahap Sumber, pilih Tambah tahap.

#### Note

Prosedur ini menunjukkan kepada Anda cara menambahkan tahap pengujian segera setelah tahap Sumber ke pipeline Anda. Untuk menambahkan tindakan pengujian ke tahap yang ada, pilih Edit stage di stage, dan kemudian lewati ke langkah 8 dari prosedur ini. Untuk menambahkan tahap pengujian di tempat lain, pilih Tambahkan tahap di tempat yang diinginkan.



**⚠ Important**

Jika Anda mengaktifkan webhook untuk CodeBuild proyek, dan proyek digunakan sebagai langkah pembangunan CodePipeline, maka dua build identik dibuat untuk setiap komit. Satu build dipicu melalui webhook dan satu melalui CodePipeline. Karena penagihan berdasarkan per-build, Anda akan ditagih untuk kedua build. Oleh karena itu, jika Anda menggunakan CodePipeline, kami sarankan Anda menonaktifkan webhook di CodeBuild. Di CodeBuild konsol, kosongkan kotak Webhook. Untuk informasi selengkapnya, lihat [Mengubah setelan proyek build \(konsol\)](#)

13. Untuk artefak Input, pilih nilai untuk artefak Output yang Anda catat sebelumnya dalam prosedur ini.
14. (Opsional) Jika Anda ingin tindakan pengujian Anda menghasilkan artefak keluaran, dan Anda mengatur buildspec Anda sesuai, maka untuk artefak Output, masukkan nilai yang ingin Anda tetapkan ke artefak keluaran.
15. Pilih Save (Simpan).
16. Pilih Rilis perubahan.
17. Setelah pipeline berjalan dengan sukses, Anda bisa mendapatkan hasil tes. Pada tahap Test pipeline, pilih CodeBuild hyperlink untuk membuka halaman proyek build terkait di CodeBuild konsol.
18. Pada halaman proyek build, dalam sejarah Build, pilih hyperlink Build run.
19. Pada halaman build run, di Build log, pilih Lihat seluruh hyperlink log untuk membuka log build di CloudWatch konsol Amazon.
20. Gulir log build untuk melihat hasil pengujian.

## Gunakan AWS CodeBuild dengan Jenkins

Anda dapat menggunakan plugin Jenkins untuk AWS CodeBuild mengintegrasikan CodeBuild dengan pekerjaan membangun Jenkins Anda. Alih-alih mengirim pekerjaan build Anda ke node build Jenkins, Anda menggunakan plugin untuk mengirim pekerjaan build Anda CodeBuild. Hal ini meniadakan kebutuhan Anda untuk menyediakan, mengonfigurasi, dan mengelola node pembangunan Jenkins.

## Menyiapkan Jenkins

Untuk informasi tentang pengaturan Jenkins dengan AWS CodeBuild plugin, dan untuk mengunduh kode sumber plugin, lihat <https://github.com/aws-labs/aws-codebuild-jenkins-plugin>.

## Menginstal plugin

Jika Anda sudah menyiapkan server Jenkins dan hanya ingin menginstal AWS CodeBuild plugin, pada instance Jenkins Anda, di Plugin Manager, cari **CodeBuild Plugin for Jenkins**.

## Menggunakan plugin

Untuk digunakan AWS CodeBuild dengan sumber dari luar VPC

1. Buat proyek di CodeBuild konsol. Untuk informasi selengkapnya, lihat [Buat proyek build \(konsol\)](#).
  - Pilih AWS Wilayah tempat Anda ingin menjalankan build.
  - (Opsional) Tetapkan konfigurasi Amazon VPC untuk memungkinkan kontainer CodeBuild build mengakses sumber daya di VPC Anda.
  - Tuliskan nama proyek Anda. Anda membutuhkannya pada langkah 3.
  - (Opsional) Jika repositori sumber Anda tidak didukung secara native CodeBuild, Anda dapat menetapkan Amazon S3 sebagai jenis sumber masukan untuk proyek Anda.
2. Di IAM Console, buat pengguna yang akan digunakan oleh plugin Jenkins.
  - Saat Anda membuat kredensi untuk pengguna, pilih Akses Terprogram.
  - Buat kebijakan yang serupa dengan yang berikut, lalu lampirkan kebijakan tersebut ke pengguna Anda.

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Resource": ["arn:aws:logs:{{region}}:{{awsAccountId}}:log-group:/aws/codebuild/{{projectName}}:*"],
 "Action": ["logs:GetLogEvents"]
 },
 {
 "Effect": "Allow",
 "Resource": ["arn:aws:s3:::{{inputBucket}}"],

```



```

 "Action": ["s3:GetBucketVersioning"]
 },
 {
 "Effect": "Allow",
 "Resource": ["arn:aws:s3:::{{inputBucket}}/{{inputObject}}"],
 "Action": ["s3:PutObject"]
 },
 {
 "Effect": "Allow",
 "Resource": ["arn:aws:s3:::{{outputBucket}}/*"],
 "Action": ["s3:GetObject"]
 },
 {
 "Effect": "Allow",
 "Resource": ["arn:aws:codebuild:{{region}}:{{awsAccountId}}:project/
{{projectName}}"],
 "Action": ["codebuild:StartBuild",
 "codebuild:BatchGetBuilds",
 "codebuild:BatchGetProjects"]
 }
]
}

```

### 3. Buat proyek gaya bebas di Jenkins.

- Pada halaman Konfigurasi, pilih Tambahkan langkah build, lalu pilih Jalankan build on CodeBuild.
- Konfigurasikan langkah build Anda.
  - Berikan nilai untuk Wilayah, Kredensi, dan Nama Proyek.
  - Pilih Gunakan sumber Proyek.
  - Simpan konfigurasi dan jalankan build dari Jenkins.

### 4. Untuk Source Code Management, pilih bagaimana Anda ingin mengambil sumber Anda. Anda mungkin perlu menginstal GitHub plugin (atau plugin Jenkins untuk penyedia repositori sumber Anda) di server Jenkins Anda.

- Pada halaman Konfigurasi, pilih Tambahkan langkah build, lalu pilih Jalankan build on AWS CodeBuild.
- Konfigurasikan langkah build Anda.
  - Berikan nilai untuk Wilayah, Kredensi, dan Nama Proyek.
  - Pilih Gunakan sumber Jenkins.

- Simpan konfigurasi dan jalankan build dari Jenkins.

Untuk menggunakan AWS CodeBuild plugin dengan plugin pipeline Jenkins

- Pada halaman proyek pipeline Jenkins Anda, gunakan generator snippet untuk menghasilkan skrip pipeline yang menambahkan CodeBuild sebagai langkah dalam pipeline Anda. Hal ini meniadakan skrip yang serupa dengan ini:

```
awsCodeBuild projectName: 'project', credentialsType: 'keys', region: 'us-west-2',
sourceControlType: 'jenkins'
```

## Gunakan AWS CodeBuild dengan Codecov

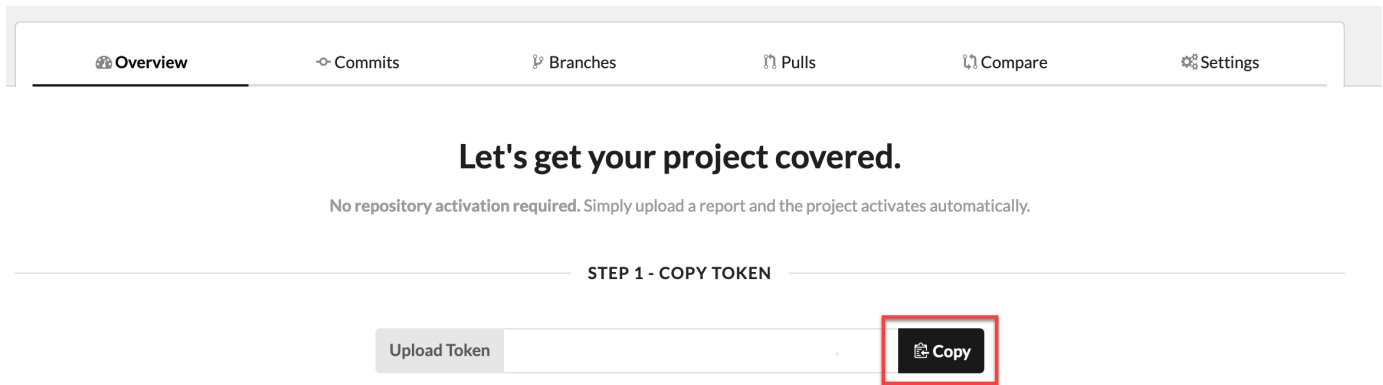
Codecov adalah alat yang mengukur cakupan uji kode Anda. Codecov mengidentifikasi metode dan pernyataan dalam kode Anda tidak diuji. Gunakan hasil untuk menentukan di mana menulis tes untuk meningkatkan kualitas kode Anda. Codecov tersedia untuk tiga repositori sumber yang didukung oleh CodeBuild: GitHub, GitHub Enterprise Server, dan Bitbucket. Jika proyek pembangunan Anda menggunakan GitHub Enterprise Server, Anda harus menggunakan Codecov Enterprise.

Ketika Anda menjalankan membangun proyek CodeBuild yang terintegrasi dengan Codecov, Codecov melaporkan bahwa menganalisis kode dalam repositori Anda diunggah ke Codecov. Log build menyertakan tautan ke laporan. Contoh ini menunjukkan kepada Anda bagaimana mengintegrasikan Python dan proyek pembangunan Java dengan Codecov. Untuk daftar bahasa yang didukung oleh Codecov, lihat [Bahasa yang didukung Codecov](#) di situs Codecov.

## Mengintegrasikan Codecov ke proyek pembangunan

Untuk mengintegrasikan Codecov dengan proyek build Anda

1. Ke <https://codecov.io/signup> dan mendaftarkan untuk repositori sumber GitHub atau Bitbucket. Jika Anda menggunakan GitHub Enterprise, lihat [Perusahaan Codecov](#) di situs Codecov.
2. Di Codecov, tambahkan repositori yang Anda inginkan cakupan.
3. Saat informasi token ditampilkan, pilih Salin.



4. Tambahkan token yang disalin sebagai variabel lingkungan bernama `CODECOV_TOKEN` untuk membangun proyek Anda. Untuk informasi selengkapnya, lihat [Mengubah setelan proyek build \(konsol\)](#).
5. Membuat file teks bernama `my_script.sh` di repositori Anda. Masukkan yang berikut ke dalam file:

```
#!/bin/bash
bash <(curl -s https://codecov.io/bash) -t $CODECOV_TOKEN
```

6. Pilih Python atau Java tab, yang sesuai untuk proyek pembangunan Anda menggunakan, dan ikuti langkah-langkah ini.

#### Java

1. Tambahkan plugin JaCoCo berikut: `pom.xml` di repositori Anda.

```
<build>
 <plugins>
 <plugin>
 <groupId>org.jacoco</groupId>
 <artifactId>jacoco-maven-plugin</artifactId>
 <version>0.8.2</version>
 <executions>
 <execution>
 <goals>
 <goal>prepare-agent</goal>
 </goals>
 </execution>
 <execution>
 <id>report</id>
 <phase>test</phase>
 </execution>
 </executions>
 </plugin>
 </plugins>
</build>
```

```

 <goals>
 <goal>report</goal>
 </goals>
 </execution>
 </executions>
 </plugin>
</plugins>
</build>

```

2. Masukkan perintah berikut di file buildspec Anda. Untuk informasi selengkapnya, lihat [Sintaks Buildspec](#).

```

build:
 - mvn test -f pom.xml -fn
postbuild:
 - echo 'Connect to CodeCov'
 - bash my_script.sh

```

## Python

Masukkan perintah berikut di file buildspec Anda. Untuk informasi selengkapnya, lihat [Sintaks Buildspec](#).

```

build:
 - pip install coverage
 - coverage run -m unittest discover
postbuild:
 - echo 'Connect to CodeCov'
 - bash my_script.sh

```

7. Jalankan membangun proyek build Anda. Tautan ke laporan Codecov yang dihasilkan untuk proyek Anda muncul di log build Anda. Gunakan tautan untuk melihat laporan Codecov. Untuk informasi selengkapnya, lihat [Jalankan build diAWS CodeBuild](#) dan [Mencatat panggilan API AWS CodeBuild dengan AWS CloudTrail](#). Informasi Codecov di log pembangunan terlihat seperti berikut ini:

```
[Container] 2020/03/09 16:31:04 Running command bash my_script.sh
```

```

 / _____ \
| | | | | | | | | |

```

```

| | / _ \ / _ ` | / _ \ \ / _ \ \ / /
| | _ | () | (| | _ / (| () \ \ /
\ \ \ \ / \ \ \ \ \ \ \ \ \ \ \ \ \ \ \

```

Bash-20200303-bc4d7e6

·[0;90m==>·[0m AWS CodeBuild detected.

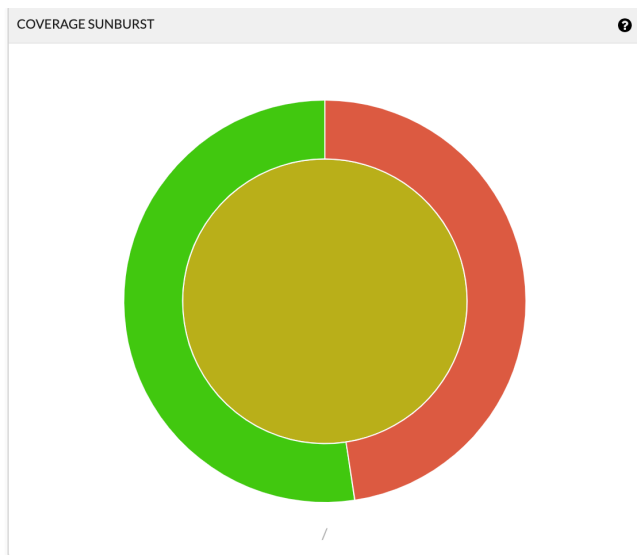
... The full list of Codecov log entries has been omitted for brevity ...

·

·[0;32m->·[0m View reports at ·[0;36m[https://codecov.io/github/user/test\\_py/commit/commit-id](https://codecov.io/github/user/test_py/commit/commit-id)·[0m

[Container] 2020/03/09 16:31:07 Phase complete: POST\_BUILD State: SUCCEEDED

Laporannya terlihat seperti berikut ini:



Files	Files	Green	Yellow	Red	Coverage
<a href="#">code.py</a>	10	7	0	3	70.00%
<a href="#">tests.py</a>	11	11	0	0	100.00%
<b>Project Totals (2 files)</b>	<b>21</b>	<b>18</b>	<b>0</b>	<b>3</b>	<b>85.71%</b>

## Gunakan AWS CodeBuild aplikasi tanpa server

Parameter AWS Serverless Application Model (AWS SAM) adalah kerangka kerja sumber terbuka untuk membangun aplikasi nirserver. Untuk informasi selengkapnya, lihat [AWS model aplikasi tanpa server](#) repositori pada GitHub.

Anda dapat menggunakan AWS CodeBuild untuk mengemas dan menyebarkan aplikasi tanpa server yang mengikuti AWS SAM standar. Untuk langkah penyebaran, CodeBuild dapat menggunakan AWS CloudFormation. Untuk mengotomatisasi bangunan dan penyebaran aplikasi tanpa server dengan CodeBuild dan AWS CloudFormation, Anda dapat menggunakan AWS CodePipeline.

Untuk informasi selengkapnya, lihat [Men-deploy Aplikasi tanpa server](#) di AWS Serverless Application Model Panduan Pengembang.

## Sumber daya terkait

- Untuk informasi tentang memulai AWS CodeBuild, lihat [Memulai dengan AWS CodeBuild menggunakan konsol](#).
- Untuk informasi tentang pemecahan masalah di CodeBuild, lihat [Pemecahan masalah AWS CodeBuild](#).
- Untuk informasi tentang kuota di CodeBuild, lihat [Kuota untuk AWS CodeBuild](#).

# Pemecahan masalah AWS CodeBuild

Gunakan informasi dalam topik ini untuk membantu Anda mengidentifikasi, mendiagnosis, dan mengatasi masalah. Untuk mempelajari cara mencatat dan memantau CodeBuild build untuk memecahkan masalah, lihat. [Pencatatan dan pemantauan](#)

## Topik

- [Apache Maven membangun artefak referensi dari repositori yang salah](#)
- [Perintah build dijalankan sebagai root secara default](#)
- [Build mungkin gagal ketika nama file memiliki non-A.S. Karakter bahasa Inggris](#)
- [Build mungkin gagal saat mendapatkan parameter dari Amazon EC2 Parameter Store](#)
- [Tidak dapat mengakses filter cabang di CodeBuild konsol](#)
- [Tidak dapat melihat keberhasilan atau kegagalan bangunan](#)
- [Status build tidak dilaporkan ke penyedia sumber](#)
- [Tidak dapat menemukan dan memilih gambar dasar platform Windows Server Core 2019](#)
- [Perintah sebelumnya dalam file buildspec tidak dikenali oleh perintah selanjutnya](#)
- [Kesalahan: "Akses ditolak" saat mencoba mengunduh cache](#)
- [Kesalahan: "BUILD\\_CONTAINER\\_UNABLE\\_TO\\_PULL\\_IMAGE" saat menggunakan gambar build khusus](#)
- [Kesalahan: "Build container ditemukan mati sebelum menyelesaikan build. build container mati karena kehabisan memori, atau image Docker tidak didukung. ErrorCode: 500"](#)
- [Kesalahan: "Tidak dapat terhubung ke daemon Docker" saat menjalankan build](#)
- [Kesalahan: "CodeBuild tidak diizinkan untuk melakukan: sts:AssumeRole" saat membuat atau memperbarui proyek build](#)
- [Kesalahan: "Kesalahan memanggil GetBucketAcl: Pemilik bucket telah berubah atau peran layanan tidak lagi memiliki izin untuk memanggil s3:GetBucketAcl"](#)
- [Kesalahan: "Gagal mengunggah artefak: Arn tidak valid" saat menjalankan build](#)
- [Kesalahan: "Klon Git gagal: Tidak dapat mengakses 'your-repository-URL': Masalah sertifikat SSL: Sertifikat yang ditandatangani sendiri"](#)
- [Kesalahan: "Bucket yang Anda coba akses harus ditangani menggunakan titik akhir yang ditentukan" saat menjalankan build](#)

- [Error: “Gambar build ini memerlukan pemilihan setidaknya satu versi runtime.”](#)
- [Kesalahan: “QUEUED: INSUFFICIENT\\_SUBNET” saat build dalam antrian build gagal](#)
- [Kesalahan: “Tidak dapat mengunduh cache: RequestError: Permintaan kirim gagal disebabkan oleh: x509: Gagal memuat akar sistem dan tidak ada akar yang disediakan”](#)
- [Kesalahan: “Tidak dapat mengunduh sertifikat dari S3. AccessDenied”](#)
- [Kesalahan: “Tidak dapat menemukan kredensi”](#)
- [RequestError kesalahan batas waktu saat berjalan CodeBuild di server proxy](#)
- [Cangkang bourne \(sh\) harus ada dalam gambar build](#)
- [Peringatan: “Melewatkan pemasangan runtime. pemilihan versi runtime tidak didukung oleh image build ini” saat menjalankan build](#)
- [Kesalahan: “Tidak dapat memverifikasi JobWorker identitas” saat membuka CodeBuild konsol](#)
- [Build gagal untuk memulai](#)
- [Mengakses GitHub metadata dalam build yang di-cache secara lokal](#)
- [AccessDenied: Pemilik bucket untuk grup laporan tidak cocok dengan pemilik bucket S3...](#)

## Apache Maven membangun artefak referensi dari repositori yang salah

Masalah: [Saat Anda menggunakan Maven dengan lingkungan build Java yang AWS CodeBuild disediakan, Maven menarik dependensi build dan plugin dari repositori Maven pusat yang aman di <https://repo1.maven.org/maven2>](#). Ini terjadi meskipun pom.xml file proyek build Anda secara eksplisit mendeklarasikan lokasi lain untuk digunakan.

Kemungkinan penyebabnya: lingkungan build Java yang CodeBuild disediakan menyertakan file bernama settings.xml yang sudah diinstal sebelumnya di direktori lingkungan build. /root/.m2 settings.xml File ini berisi deklarasi berikut, yang menginstruksikan Maven untuk selalu menarik dependensi build dan plugin dari repositori Maven pusat yang aman di <https://repo1.maven.org/maven2>.

```
<settings>
 <activeProfiles>
 <activeProfile>securecentral</activeProfile>
 </activeProfiles>
```



```
<profiles>
 <profile>
 <id>securecentral</id>
 <repositories>
 <repository>
 <id>central</id>
 <url>https://repo1.maven.org/maven2</url>
 <releases>
 <enabled>true</enabled>
 </releases>
 </repository>
 </repositories>
 <pluginRepositories>
 <pluginRepository>
 <id>central</id>
 <url>https://repo1.maven.org/maven2</url>
 <releases>
 <enabled>true</enabled>
 </releases>
 </pluginRepository>
 </pluginRepositories>
 </profile>
</profiles>
</settings>
```

Solusi yang disarankan: Lakukan hal berikut:

1. Tambahkan `settings.xml` file ke kode sumber Anda.
2. Dalam `settings.xml` file ini, gunakan `settings.xml` format sebelumnya sebagai panduan untuk mendeklarasikan repositori yang Anda inginkan Maven untuk menarik dependensi build dan plugin.
3. Pada `install` fase proyek build Anda, instruksikan CodeBuild untuk menyalin `settings.xml` file Anda ke `/root/.m2` direktori lingkungan build. Misalnya, pertimbangkan cuplikan berikut dari `buildspec.yml` file yang menunjukkan perilaku ini.

```
version 0.2

phases:
 install:
 commands:
 - cp ./settings.xml /root/.m2/settings.xml
```

## Perintah build dijalankan sebagai root secara default

**Masalah:** AWS CodeBuild menjalankan perintah build Anda sebagai pengguna root. Ini terjadi bahkan jika Dockerfile image build terkait Anda menyetel USER instruksi ke pengguna yang berbeda.

**Penyebab:** Secara default, CodeBuild menjalankan semua perintah build sebagai pengguna root.

**Solusi yang disarankan:** Tidak ada.

## Build mungkin gagal ketika nama file memiliki non-A.S. Karakter bahasa Inggris

**Masalah:** Saat Anda menjalankan build yang menggunakan file dengan nama file yang berisi non-U.S. Karakter bahasa Inggris (misalnya, karakter Mandarin), build gagal.

**Kemungkinan penyebabnya:** Lingkungan build yang disediakan oleh AWS CodeBuild mengatur lokal defaultnya. POSIX pengaturannya kurang kompatibel dengan CodeBuild dan nama file yang berisi non-A.S. Karakter bahasa Inggris dan dapat menyebabkan build terkait gagal.

**Solusi yang disarankan:** Tambahkan perintah berikut ke `pre_build` bagian file buildspec Anda. Perintah ini membuat lingkungan build menggunakan U.S. English UTF-8 untuk pengaturan lokalisasi, yang lebih kompatibel dengan CodeBuild dan nama file yang berisi non-AS. Karakter bahasa Inggris.

Untuk lingkungan build berdasarkan Ubuntu:

```
pre_build:
 commands:
 - export LC_ALL="en_US.UTF-8"
 - locale-gen en_US en_US.UTF-8
 - dpkg-reconfigure locales
```

Untuk lingkungan build berbasis Amazon Linux:

```
pre_build:
 commands:
 - export LC_ALL="en_US.utf8"
```

# Build mungkin gagal saat mendapatkan parameter dari Amazon EC2 Parameter Store

Masalah: Saat build mencoba mendapatkan nilai dari satu atau beberapa parameter yang disimpan di Amazon EC2 Parameter Store, build gagal dalam `DOWNLOAD_SOURCE` fase dengan kesalahan.

`Parameter does not exist`

Kemungkinan penyebabnya: Peran layanan yang diandalkan proyek build tidak memiliki izin untuk memanggil `ssm:GetParameters` tindakan atau proyek build menggunakan peran layanan yang dihasilkan oleh AWS CodeBuild dan memungkinkan pemanggilan `ssm:GetParameters` tindakan, tetapi parameter memiliki nama yang tidak dimulai/`CodeBuild/`.

Solusi yang direkomendasikan:

- Jika peran layanan tidak dihasilkan oleh CodeBuild, perbarui definisinya untuk memungkinkan CodeBuild untuk memanggil `ssm:GetParameters` tindakan. Misalnya, pernyataan kebijakan berikut memungkinkan pemanggilan `ssm:GetParameters` tindakan untuk mendapatkan parameter dengan nama yang dimulai dengan `/CodeBuild/`:

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Action": "ssm:GetParameters",
 "Effect": "Allow",
 "Resource": "arn:aws:ssm:REGION_ID:ACCOUNT_ID:parameter/CodeBuild/*"
 }
]
}
```

- Jika peran layanan dihasilkan oleh CodeBuild, perbarui definisinya untuk memungkinkan CodeBuild untuk mengakses parameter di Amazon EC2 Parameter Store dengan nama selain yang dimulai dengan `/CodeBuild/`. Misalnya, pernyataan kebijakan berikut memungkinkan pemanggilan `ssm:GetParameters` tindakan untuk mendapatkan parameter dengan nama yang ditentukan:

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
```

```
"Action": "ssm:GetParameters",
"Effect": "Allow",
"Resource": "arn:aws:ssm:REGION_ID:ACCOUNT_ID:parameter/PARAMETER_NAME"
}
]
}
```

## Tidak dapat mengakses filter cabang di CodeBuild konsol

Masalah: Opsi filter cabang tidak tersedia di konsol saat Anda membuat atau memperbarui AWS CodeBuild proyek.

Kemungkinan penyebabnya: Opsi filter cabang tidak digunakan lagi. Ini telah digantikan oleh grup filter webhook, yang memberikan kontrol lebih besar atas peristiwa webhook yang memicu build baru. CodeBuild

Solusi yang disarankan: Untuk memigrasikan filter cabang yang Anda buat sebelum pengenalan filter webhook, buat grup filter webhook dengan HEAD\_REF filter dengan ekspresi reguler. `^refs/heads/branchName$` Misalnya, jika ekspresi reguler filter cabang Anda adalah `^branchName$`, maka ekspresi reguler yang diperbarui yang Anda masukkan ke dalam HEAD\_REF filter adalah `^refs/heads/branchName$`. Lihat informasi yang lebih lengkap di [Acara webhook Bitbucket](#) dan [Filter acara GitHub webhook \(konsol\)](#).

## Tidak dapat melihat keberhasilan atau kegagalan bangunan

Masalah: Anda tidak dapat melihat keberhasilan atau kegagalan build yang dicoba lagi.

Kemungkinan penyebabnya: Opsi untuk melaporkan status build Anda tidak diaktifkan.

Solusi yang disarankan: Aktifkan status pembuatan Laporan saat Anda membuat atau memperbarui CodeBuild proyek. Opsi ini memberi tahu CodeBuild untuk melaporkan kembali status saat Anda memicu build. Untuk informasi selengkapnya, lihat [reportBuildStatus](#) di dalam Referensi API AWS CodeBuild .

## Status build tidak dilaporkan ke penyedia sumber

Masalah: Setelah mengizinkan pelaporan status build ke penyedia sumber, seperti GitHub atau Bitbucket, status build tidak diperbarui.

Kemungkinan penyebabnya: Pengguna yang terkait dengan penyedia sumber tidak memiliki akses tulis ke repo.

Solusi yang disarankan: Untuk dapat melaporkan status build ke penyedia sumber, pengguna yang terkait dengan penyedia sumber harus memiliki akses tulis ke repo. Jika pengguna tidak memiliki akses tulis, status build tidak dapat diperbarui. Untuk informasi selengkapnya, lihat [Akses penyedia sumber](#).

## Tidak dapat menemukan dan memilih gambar dasar platform Windows Server Core 2019

Masalah: Anda tidak dapat menemukan atau memilih gambar dasar platform Windows Server Core 2019.

Kemungkinan penyebabnya: Anda menggunakan AWS Wilayah yang tidak mendukung gambar ini.

Solusi yang disarankan: Gunakan salah satu AWS Wilayah berikut di mana gambar dasar platform Windows Server Core 2019 didukung:

- AS Timur (N. Virginia)
- AS Timur (Ohio)
- AS Barat (Oregon)
- Eropa (Irlandia)

## Perintah sebelumnya dalam file buildspec tidak dikenali oleh perintah selanjutnya

Masalah: Hasil dari satu atau beberapa perintah dalam file buildspec Anda tidak dikenali oleh perintah selanjutnya dalam file buildspec yang sama. Misalnya, perintah mungkin menyetel variabel lingkungan lokal, tetapi perintah yang dijalankan nanti mungkin gagal mendapatkan nilai variabel lingkungan lokal tersebut.

Kemungkinan penyebabnya: Dalam file buildspec versi 0.1, AWS CodeBuild jalankan setiap perintah dalam instance terpisah dari shell default di lingkungan build. Ini berarti bahwa setiap perintah berjalan secara terpisah dari semua perintah lainnya. Secara default, Anda tidak dapat menjalankan satu perintah yang bergantung pada status perintah sebelumnya.

Solusi yang disarankan: Kami menyarankan Anda menggunakan spesifikasi build versi 0.2, yang memecahkan masalah ini. Jika Anda harus menggunakan buildspec versi 0.1, kami sarankan Anda menggunakan operator rantai perintah shell (misalnya, `&&` di Linux) untuk menggabungkan beberapa perintah menjadi satu perintah. Atau sertakan skrip shell dalam kode sumber Anda yang berisi beberapa perintah, lalu panggil skrip shell itu dari satu perintah di file buildspec. Lihat informasi yang lebih lengkap di [Kerang dan perintah dalam membangun lingkungan](#) dan [Variabel lingkungan di lingkungan build](#).

## Kesalahan: “Akses ditolak” saat mencoba mengunduh cache

Masalah: Saat mencoba mengunduh cache pada proyek build yang memiliki cache diaktifkan, Anda menerima `Access denied` kesalahan.

Kemungkinan penyebabnya:

- Anda baru saja mengonfigurasi caching sebagai bagian dari proyek build Anda.
- Cache baru-baru ini dibatalkan melalui API. `InvalidateProjectCache`
- Peran layanan yang digunakan oleh CodeBuild tidak memiliki `s3:GetObject` dan `s3:PutObject` izin ke bucket S3 yang menahan cache.

Solusi yang disarankan: Untuk penggunaan pertama kali, normal untuk melihat ini segera setelah memperbarui konfigurasi cache. Jika kesalahan ini berlanjut, maka Anda harus memeriksa untuk melihat apakah peran layanan Anda memiliki `s3:GetObject` dan `s3:PutObject` izin ke bucket S3 yang menahan cache. Untuk informasi selengkapnya, lihat [Menentukan izin S3](#) di Panduan Pengembang Amazon S3.

## Kesalahan: “BUILD\_CONTAINER\_UNABLE\_TO\_PULL\_IMAGE” saat menggunakan gambar build khusus

Masalah: Saat Anda mencoba menjalankan build yang menggunakan image build kustom, build gagal dengan kesalahan tersebut `BUILD_CONTAINER_UNABLE_TO_PULL_IMAGE`.

Kemungkinan penyebabnya: Ukuran keseluruhan gambar build yang tidak terkompresi lebih besar daripada ruang disk tipe komputasi lingkungan build yang tersedia. Untuk memeriksa ukuran image build Anda, gunakan Docker untuk menjalankan `docker images REPOSITORY:TAG` perintah. Untuk daftar ruang disk yang tersedia menurut jenis komputasi, lihat [Bangun mode dan tipe komputasi lingkungan](#).

Solusi yang disarankan: Gunakan jenis komputasi yang lebih besar dengan lebih banyak ruang disk yang tersedia, atau kurangi ukuran image build kustom Anda.

Kemungkinan penyebabnya: AWS CodeBuild tidak memiliki izin untuk menarik image build dari Amazon Elastic Container Registry (Amazon ECR).

Solusi yang disarankan: Perbarui izin di repositori Anda di Amazon ECR sehingga CodeBuild dapat menarik image build kustom Anda ke lingkungan build. Lihat informasi yang lebih lengkap di [Sampel Amazon ECR](#).

Kemungkinan penyebabnya: Gambar Amazon ECR yang Anda minta tidak tersedia di AWS Wilayah yang digunakan AWS akun Anda.

Solusi yang disarankan: Gunakan gambar Amazon ECR yang berada di AWS Wilayah yang sama dengan yang digunakan AWS akun Anda.

Kemungkinan penyebabnya: Anda menggunakan registri pribadi di VPC yang tidak memiliki akses internet publik. CodeBuild tidak dapat menarik gambar dari alamat IP pribadi di VPC. Untuk informasi selengkapnya, lihat [Registri pribadi dengan AWS Secrets Manager sampel untuk CodeBuild](#).

Solusi yang disarankan: Jika Anda menggunakan registri pribadi di VPC, pastikan VPC memiliki akses internet publik.

Kemungkinan penyebabnya: Jika pesan kesalahan berisi "toomanyrequests", dan gambar diperoleh dari Docker Hub, kesalahan ini berarti batas tarik Docker Hub telah tercapai.

Solusi yang disarankan: Gunakan registri pribadi Docker Hub, atau dapatkan gambar Anda dari Amazon ECR. Untuk informasi selengkapnya tentang menggunakan registri pribadi, lihat [Registri pribadi dengan AWS Secrets Manager sampel untuk CodeBuild](#). Untuk informasi selengkapnya tentang menggunakan Amazon ECR, lihat [Sampel Amazon ECR untuk CodeBuild](#).

## Kesalahan: “Build container ditemukan mati sebelum menyelesaikan build. build container mati karena kehabisan memori, atau image Docker tidak didukung. ErrorCode: 500”

Masalah: Saat Anda mencoba menggunakan wadah Microsoft Windows atau Linux AWS CodeBuild, kesalahan ini terjadi selama fase PENYEDIAAN.

Kemungkinan penyebabnya:

- Versi OS kontainer tidak didukung oleh CodeBuild.
- HTTP\_PROXY,HTTPS\_PROXY, atau keduanya ditentukan dalam wadah.

Solusi yang direkomendasikan:

- Untuk Microsoft Windows, gunakan wadah Windows dengan OS kontainer yaitu versi microsoft/windowsservercore:10.0.x (misalnya, microsoft/windowsservercore:10.0.14393.2125).
- Untuk Linux, hapus HTTPS\_PROXY pengaturan HTTP\_PROXY dan pada image Docker Anda, atau tentukan konfigurasi VPC di project build Anda.

## Kesalahan: “Tidak dapat terhubung ke daemon Docker” saat menjalankan build

Masalah: Build Anda gagal dan Anda menerima kesalahan yang mirip dengan Cannot connect to the Docker daemon at unix:///var/run/docker.sock. Is the docker daemon running? di log build.

Kemungkinan penyebabnya: Anda tidak menjalankan build dalam mode istimewa.

Solusi yang disarankan: Untuk memperbaiki kesalahan ini, Anda harus mengaktifkan mode istimewa dan memperbarui buildspec Anda menggunakan instruksi berikut.

Untuk menjalankan build Anda dalam mode istimewa, ikuti langkah-langkah berikut:

1. Buka CodeBuild konsol di <https://console.aws.amazon.com/codebuild/>.
2. Di panel navigasi, pilih Build projects, lalu pilih build project Anda.
3. Dari Edit, pilih Lingkungan.



4. Pilih Konfigurasi tambahan.
5. Dari Privileged, pilih Aktifkan bendera ini jika Anda ingin membuat gambar Docker atau ingin build Anda mendapatkan hak istimewa yang lebih tinggi. .
6. Pilih Perbarui lingkungan.
7. Pilih Mulai membangun untuk mencoba lagi build Anda.

Anda juga harus memulai daemon Docker di dalam wadah Anda. `install` fase buildspec Anda mungkin terlihat mirip dengan ini.

```
phases:
 install:
 commands:
 - nohup /usr/local/bin/dockerd --host=unix:///var/run/docker.sock --
host=tcp://127.0.0.1:2375 --storage-driver=overlay2 &
 - timeout 15 sh -c "until docker info; do echo .; sleep 1; done"
```

Untuk informasi selengkapnya tentang driver penyimpanan OverlayFS yang direferensikan dalam file buildspec, [lihat Menggunakan driver penyimpanan OverlayFS di situs web Docker](#).

#### Note

Jika sistem operasi dasar adalah Alpine Linux, buildspec.yml tambahkan `-t` argumen `timeout`:

```
- timeout -t 15 sh -c "until docker info; do echo .; sleep 1; done"
```

Untuk mempelajari lebih lanjut tentang cara membuat dan menjalankan image Docker dengan menggunakan AWS CodeBuild, lihat [Docker dalam sampel gambar khusus untuk CodeBuild](#).

## Kesalahan: "CodeBuild tidak diizinkan untuk melakukan: sts:AssumeRole" saat membuat atau memperbarui proyek build

Masalah: Saat Anda mencoba membuat atau memperbarui proyek build, Anda menerima kesalahan `Code:InvalidInputException`, `Message:CodeBuild is not authorized to perform: sts:AssumeRole on arn:aws:iam::account-ID:role/service-role-name`.

Kemungkinan penyebabnya:

- The AWS Security Token Service (AWS STS) telah dinonaktifkan untuk AWS wilayah tempat Anda mencoba membuat atau memperbarui proyek build.
- Peran AWS CodeBuild layanan yang terkait dengan proyek build tidak ada atau tidak memiliki izin yang cukup untuk dipercaya CodeBuild.

Solusi yang direkomendasikan:

- Pastikan AWS STS diaktifkan untuk AWS wilayah tempat Anda mencoba membuat atau memperbarui proyek build. Untuk informasi selengkapnya, lihat [Mengaktifkan dan menonaktifkan AWS STS di AWS Wilayah di Panduan Pengguna IAM](#).
- Pastikan peran CodeBuild layanan target ada di AWS akun Anda. Jika Anda tidak menggunakan konsol, pastikan Anda tidak salah mengeja Nama Sumber Daya Amazon (ARN) peran layanan saat Anda membuat atau memperbarui proyek build.
- Pastikan peran CodeBuild layanan target memiliki izin yang cukup untuk dipercaya CodeBuild. Untuk informasi selengkapnya, lihat pernyataan kebijakan hubungan kepercayaan di [Membuat peran CodeBuild layanan](#).

**Kesalahan: “Kesalahan memanggil GetBucketAcl: Pemilik bucket telah berubah atau peran layanan tidak lagi memiliki izin untuk memanggil s3:GetBucketAcl”**

Masalah: Saat menjalankan build, Anda menerima error tentang perubahan kepemilikan bucket dan GetBucketAcl izin S3.

Kemungkinan penyebabnya: Anda menambahkan s3:GetBucketAcl dan s3:GetBucketLocation izin ke peran IAM Anda. Izin ini mengamankan bucket S3 proyek Anda dan memastikan bahwa hanya Anda yang dapat mengaksesnya. Setelah Anda menambahkan izin ini, pemilik bucket S3 berubah.

Solusi yang disarankan: Pastikan Anda adalah pemilik bucket S3, lalu tambahkan izin ke peran IAM Anda lagi. Untuk informasi selengkapnya, lihat [Akses aman ke bucket S3](#).

## Kesalahan: “Gagal mengunggah artefak: Arn tidak valid” saat menjalankan build

Masalah: Saat Anda menjalankan build, fase `UPLOAD_ARTIFACTS` build gagal dengan kesalahan `Failed to upload artifacts: Invalid arn`.

Kemungkinan penyebabnya: Bucket keluaran S3 Anda (bucket tempat AWS CodeBuild menyimpan outputnya dari build) berada di AWS Wilayah yang berbeda dari proyek CodeBuild build.

Solusi yang disarankan: Perbarui pengaturan proyek build untuk menunjuk ke bucket keluaran yang berada di AWS Wilayah yang sama dengan proyek build.

## Kesalahan: “Klon Git gagal: Tidak dapat mengakses '**your-repository-URL**': Masalah sertifikat SSL: Sertifikat yang ditandatangani sendiri”

Masalah: Saat Anda mencoba menjalankan proyek build, build gagal dengan kesalahan ini.

Kemungkinan penyebabnya: Repositori sumber Anda memiliki sertifikat yang ditandatangani sendiri, tetapi Anda belum memilih untuk menginstal sertifikat dari bucket S3 sebagai bagian dari proyek build Anda.

Solusi yang direkomendasikan:

- Edit proyek Anda. Untuk Sertifikat, pilih Instal sertifikat dari S3. Untuk Bucket of certificate, pilih bucket S3 tempat sertifikat SSL Anda disimpan. Untuk kunci Objek sertifikat, masukkan nama kunci objek S3 Anda.
- Edit proyek Anda. Pilih SSL Tidak Aman untuk mengabaikan peringatan SSL saat menghubungkan ke repositori proyek GitHub Enterprise Server Anda.

### Note

Kami menyarankan Anda menggunakan SSL Insecure hanya untuk pengujian. Ini tidak boleh digunakan di lingkungan produksi.

## Kesalahan: “Bucket yang Anda coba akses harus ditangani menggunakan titik akhir yang ditentukan” saat menjalankan build

Masalah: Saat Anda menjalankan build, fase `DOWNLOAD_SOURCE` build gagal dengan kesalahan `The bucket you are attempting to access must be addressed using the specified endpoint. Please send all future requests to this endpoint.`

Kemungkinan penyebabnya: Kode sumber bawaan Anda disimpan dalam bucket S3, dan bucket tersebut berada di AWS Wilayah yang berbeda dari proyek AWS CodeBuild build.

Solusi yang disarankan: Perbarui pengaturan proyek build untuk menunjuk ke bucket yang berisi kode sumber bawaan Anda. Pastikan bucket berada di AWS Region yang sama dengan proyek build.

## Error: “Gambar build ini memerlukan pemilihan setidaknya satu versi runtime.”

Masalah: Saat Anda menjalankan build, fase `DOWNLOAD_SOURCE` build gagal dengan kesalahan `YAML_FILE_ERROR: This build image requires selecting at least one runtime version.`

Kemungkinan penyebabnya: Build Anda menggunakan image standar Amazon Linux 2 (AL2) versi 1.0 atau yang lebih baru, atau image standar Ubuntu versi 2.0 atau yang lebih baru, dan runtime tidak ditentukan dalam file `buildspec`.

Solusi yang disarankan: Jika Anda menggunakan image `aws/codebuild/standard:2.0` CodeBuild terkelola, Anda harus menentukan versi runtime di `runtime-versions` bagian file `buildspec`. Misalnya, Anda mungkin menggunakan file `buildspec` berikut untuk proyek yang menggunakan PHP:

```
version: 0.2

phases:
 install:
 runtime-versions:
 php: 7.3
 build:
 commands:
 - php --version
```

```
artifacts:
 files:
 - README.md
```

### Note

Jika Anda menentukan `runtime-versions` bagian dan menggunakan gambar selain Ubuntu Standard Image 2.0 atau yang lebih baru, atau Amazon Linux 2 (AL2) image standar 1.0 atau yang lebih baru, build akan mengeluarkan peringatan, "Skipping install of runtimes. Runtime version selection is not supported by this build image."

Untuk informasi selengkapnya, lihat [Specify runtime versions in the buildspec file](#).

## Kesalahan: "QUEUED: INSUFFICIENT\_SUBNET" saat build dalam antrian build gagal

Masalah: Build dalam antrian build gagal dengan kesalahan yang mirip QUEUED: INSUFFICIENT\_SUBNET dengan.

Kemungkinan penyebabnya: Blok CIDR IPv4 yang ditentukan untuk VPC Anda menggunakan alamat IP yang dipesan. Empat alamat IP pertama dan alamat IP terakhir di setiap blok CIDR subnet tidak tersedia untuk Anda gunakan dan tidak dapat ditetapkan ke sebuah instance. Sebagai contoh, dalam sebuah subnet dengan blok CIDR `10.0.0.0/24`, lima alamat IP berikut dicadangkan:

- `10.0.0.0`: Alamat jaringan.
- `10.0.0.1`: Dicadangkan oleh AWS untuk router VPC.
- `10.0.0.2`: Dicadangkan oleh AWS. Alamat IP server DNS selalu menjadi basis rentang jaringan VPC ditambah dua; Namun, kami juga mencadangkan basis setiap rentang subnet ditambah dua. Untuk VPC dengan beberapa blok CIDR, alamat IP server DNS terletak di CIDR utama. Lihat informasi yang lebih lengkap di [Server DNS Amazon](#) dalam Panduan Pengguna Amazon VPC.
- `10.0.0.3`: Dicadangkan oleh AWS untuk penggunaan masa depan.
- `10.0.0.255`: Alamat siaran jaringan. Kami tidak mendukung siaran dalam VPC. Alamat ini dipesan.

Solusi yang disarankan: Periksa apakah VPC Anda menggunakan alamat IP yang dipesan. Ganti alamat IP yang dicadangkan dengan alamat yang tidak dicadangkan. Untuk informasi selengkapnya, lihat [Ukuran VPC dan subnet](#) dalam Panduan Pengguna Amazon VPC.

## Kesalahan: “Tidak dapat mengunduh cache: RequestError: Permintaan kirim gagal disebabkan oleh: x509: Gagal memuat akar sistem dan tidak ada akar yang disediakan”

Masalah: Saat Anda mencoba menjalankan proyek build, build gagal dengan kesalahan ini.

Kemungkinan penyebabnya: Anda mengonfigurasi caching sebagai bagian dari proyek build dan menggunakan image Docker lama yang menyertakan sertifikat root yang kedaluwarsa.

Solusi yang disarankan: Perbarui gambar Docker yang sedang digunakan dalam proyek Anda AWS CodeBuild . Untuk informasi selengkapnya, lihat [Gambar Docker disediakan oleh CodeBuild](#).

## Kesalahan: “Tidak dapat mengunduh sertifikat dari S3. AccessDenied”

Masalah: Saat Anda mencoba menjalankan proyek build, build gagal dengan kesalahan ini.

Kemungkinan penyebabnya:

- Anda telah memilih bucket S3 yang salah untuk sertifikat Anda.
- Anda telah memasukkan kunci objek yang salah untuk sertifikat Anda.

Solusi yang direkomendasikan:

- Edit proyek Anda. Untuk Bucket of certificate, pilih bucket S3 tempat sertifikat SSL Anda disimpan.
- Edit proyek Anda. Untuk kunci Objek sertifikat, masukkan nama kunci objek S3 Anda.

## Kesalahan: “Tidak dapat menemukan kredensi”

Masalah: Saat Anda mencoba menjalankan AWS CLI, menggunakan AWS SDK, atau memanggil komponen serupa lainnya sebagai bagian dari build, Anda mendapatkan error build yang terkait

langsung dengan AWS CLI, AWS SDK, atau komponen. Misalnya, Anda mungkin mendapatkan kesalahan build seperti `Unable to locate credentials`.

Kemungkinan penyebabnya:

- Versi AWS CLI, AWS SDK, atau komponen di lingkungan build tidak kompatibel dengannya. AWS CodeBuild
- Anda menjalankan wadah Docker dalam lingkungan build yang menggunakan Docker, dan penampung tidak memiliki akses ke AWS kredensial secara default.

Solusi yang direkomendasikan:

- Pastikan lingkungan build Anda memiliki versi, AWS SDK AWS CLI, atau komponen berikut atau yang lebih tinggi.
  - AWS CLI: 1.10.47
  - AWS SDK for C++: 0.2.19
  - AWS SDK for Go: 1.2.5
  - AWS SDK for Java: 1.11.16
  - AWS SDK untuk JavaScript: 2.4.7
  - AWS SDK for PHP: 3.18.28
  - AWS SDK untuk Python (Boto3): 1.4.0
  - AWS SDK for Ruby: 2.3.22
  - Botocore: 1.4.37
  - CoreCLR: 3.2.6-beta
  - Node.js: 2.4.7
- Jika Anda perlu menjalankan container Docker di lingkungan build dan container memerlukan AWS kredensial, Anda harus melewati kredensial dari lingkungan build ke container. Dalam file `buildspec` Anda, sertakan `run` perintah Docker seperti berikut ini. Contoh ini menggunakan `aws s3 ls` perintah untuk membuat daftar bucket S3 Anda yang tersedia. -eOpsi melewati variabel lingkungan yang diperlukan untuk penampung Anda untuk mengakses AWS kredensial.

```
docker run -e AWS_DEFAULT_REGION -e AWS_CONTAINER_CREDENTIALS_RELATIVE_URI your-
image-tag aws s3 ls
```

- Jika Anda membuat image Docker dan build memerlukan AWS kredensial (misalnya, untuk mengunduh file dari Amazon S3), Anda harus meneruskan kredensial dari lingkungan build ke proses build Docker sebagai berikut.

1. Di Dockerfile kode sumber Anda untuk gambar Docker, tentukan instruksi berikut. ARG

```
ARG AWS_DEFAULT_REGION
ARG AWS_CONTAINER_CREDENTIALS_RELATIVE_URI
```

2. Dalam file buildspec Anda, sertakan build perintah Docker seperti berikut ini. --build-arg Opsi menetapkan variabel lingkungan yang diperlukan untuk proses pembuatan Docker Anda untuk mengakses AWS kredensial.

```
docker build --build-arg AWS_DEFAULT_REGION=$AWS_DEFAULT_REGION --build-arg
AWS_CONTAINER_CREDENTIALS_RELATIVE_URI=$AWS_CONTAINER_CREDENTIALS_RELATIVE_URI -
t your-image-tag .
```

## RequestError kesalahan batas waktu saat berjalan CodeBuild di server proxy

Masalah: Anda menerima RequestError kesalahan yang mirip dengan salah satu dari berikut ini:

- RequestError: send request failed caused by: Post https://logs.<your-region>.amazonaws.com/: dial tcp 52.46.158.105:443: i/o timeout dari CloudWatch Log.
- Error uploading artifacts: RequestError: send request failed caused by: Put https://*your-bucket*.s3.*your-aws-region*.amazonaws.com/\*: dial tcp 52.219.96.208:443: connect: connection refused dari Amazon S3.

Kemungkinan penyebabnya:

- ssl-bump tidak dikonfigurasi dengan benar.
- Kebijakan keamanan organisasi Anda tidak memungkinkan Anda untuk menggunakannya ssl\_bump.
- File buildspec Anda tidak memiliki pengaturan proxy yang ditentukan menggunakan elemen proxy



## Solusi yang direkomendasikan:

- Pastikan `ssl-bump` dikonfigurasi dengan benar. Jika Anda menggunakan Squid untuk server proxy Anda, lihat [Mengkonfigurasi Squid sebagai server proxy eksplisit](#).
- Ikuti langkah-langkah berikut untuk menggunakan titik akhir pribadi untuk Amazon S3 CloudWatch dan Log:
  1. Di tabel perutean subnet pribadi Anda, hapus aturan yang Anda tambahkan yang mengarahkan lalu lintas yang ditujukan untuk internet ke server proxy Anda. Untuk selengkapnya, lihat [Membuat subnet di VPC Anda di](#) Panduan Pengguna Amazon VPC.
  2. Buat endpoint Amazon S3 pribadi dan titik akhir CloudWatch Log dan kaitkan dengan subnet pribadi VPC Amazon Anda. Untuk selengkapnya, lihat [layanan titik akhir VPC di Panduan Pengguna Amazon VPC](#).
  3. Konfirmasikan Aktifkan Nama DNS Pribadi di Amazon VPC Anda dipilih. Untuk informasi selengkapnya, lihat [Membuat titik akhir antarmuka](#) dalam Panduan Pengguna Amazon VPC.
- Jika Anda tidak menggunakan `ssl-bump` server proxy eksplisit, tambahkan konfigurasi proxy ke file `buildspec` Anda menggunakan elemen `proxy`. Lihat informasi yang lebih lengkap di [Jalankan CodeBuild di server proxy eksplisit](#) dan [Sintaks Buildspec](#).

```
version: 0.2
proxy:
 upload-artifacts: yes
 logs: yes
phases:
 build:
 commands:
```

## Cangkang bourne (sh) harus ada dalam gambar build

Masalah: Anda menggunakan image build yang tidak disediakan oleh AWS CodeBuild, dan build Anda gagal dengan pesan `Build container found dead before completing the build` tersebut.

Kemungkinan penyebabnya: Bourne shell (`sh`) tidak disertakan dalam image build Anda. CodeBuild `sh` perlu menjalankan perintah dan skrip build.

Solusi yang disarankan: Jika sh tidak ada dalam image build Anda, pastikan untuk menyertakannya sebelum memulai build lagi yang menggunakan image Anda. (CodeBuild sudah termasuk sh dalam gambar build-nya.)

## Peringatan: “Melewatkan pemasangan runtime. pemilihan versi runtime tidak didukung oleh image build ini” saat menjalankan build

Masalah: Saat Anda menjalankan build, log build berisi peringatan ini.

Kemungkinan penyebabnya: Build Anda tidak menggunakan image standar Amazon Linux 2 (AL2) versi 1.0 atau yang lebih baru, atau image standar Ubuntu versi 2.0 atau yang lebih baru, dan runtime ditentukan di `runtime-versions` bagian dalam file `buildspec` Anda.

Solusi yang disarankan: Pastikan file `buildspec` Anda tidak berisi bagian `runtime-versions` `runtime-versions` Bagian ini hanya diperlukan jika Anda menggunakan gambar standar Amazon Linux 2 (AL2) atau yang lebih baru atau gambar standar Ubuntu versi 2.0 atau yang lebih baru.

## Kesalahan: “Tidak dapat memverifikasi JobWorker identitas” saat membuka CodeBuild konsol

Masalah: Saat Anda membuka CodeBuild konsol, pesan kesalahan “Tidak dapat memverifikasi JobWorker identitas” ditampilkan.

Kemungkinan penyebabnya: Peran IAM yang digunakan untuk akses konsol memiliki tag dengan `jobId` kunci. Kunci tag ini dicadangkan untuk CodeBuild dan akan menyebabkan kesalahan ini jika ada.

Solusi yang disarankan: Ubah tag peran IAM khusus yang memiliki kunci `jobId` untuk memiliki kunci yang berbeda, seperti `jobIdentifier`.

## Build gagal untuk memulai

Masalah: Saat memulai build, Anda menerima pesan galat `Build failed to start`.

Kemungkinan penyebabnya: Jumlah build bersamaan telah tercapai.

Solusi yang disarankan: Tunggu hingga build lain selesai, atau tingkatkan batas build bersamaan untuk proyek, dan mulai build lagi. Untuk informasi selengkapnya, lihat [Konfigurasi proyek](#).

# Mengakses GitHub metadata dalam build yang di-cache secara lokal

**Masalah:** Dalam beberapa kasus, direktori.git dalam build cache adalah file teks dan bukan direktori.

**Kemungkinan penyebabnya:** Saat caching sumber lokal diaktifkan untuk build, CodeBuild buat gitlink untuk direktori. .git Ini berarti bahwa .git direktori sebenarnya adalah file teks yang berisi path ke direktori.

**Solusi yang disarankan:** Dalam semua kasus, gunakan perintah berikut untuk mendapatkan direktori metadata Git. Perintah ini akan berfungsi apa pun format.git:

```
git rev-parse --git-dir
```

## AccessDenied: Pemilik bucket untuk grup laporan tidak cocok dengan pemilik bucket S3...

**Masalah:** Saat mengunggah data pengujian ke bucket Amazon S3 CodeBuild , tidak dapat menulis data pengujian ke bucket.

**Kemungkinan penyebabnya:**

- Akun yang ditentukan untuk pemilik bucket grup laporan tidak cocok dengan pemilik bucket Amazon S3.
- Peran layanan tidak memiliki akses tulis ke bucket.

**Solusi yang direkomendasikan:**

- Ubah pemilik bucket grup laporan agar sesuai dengan pemilik bucket Amazon S3.
- Ubah peran layanan untuk mengizinkan akses tulis ke bucket Amazon S3.

# Kuota untuk AWS CodeBuild

Tabel berikut mencantumkan kuota saat ini diAWS CodeBuild. Kuota ini untuk setiap AWS Wilayah yang didukung untuk setiap AWS akun, kecuali ditentukan lain.

## Kuota layanan

Berikut ini adalah kuota default untuk AWS CodeBuild layanan ini.

Nama	Default	Dapat disesuai	Deskripsi
Tag terkait per proyek	Setiap Wilayah yang didukung: 50	Tidak	Jumlah maksimum tag yang dapat Anda kaitkan dengan proyek build
Proyek Build	Setiap Wilayah yang didukung: 5.000	<a href="#">Ya</a>	Jumlah maksimum proyek pembangunan
Batas waktu Build dalam menit	Setiap Wilayah yang didukung: 480	Tidak	Batas waktu build maksimum dalam hitungan menit
Permintaan bersamaan untuk informasi tentang build	Setiap Wilayah yang didukung: 100	Tidak	Jumlah maksimum build yang dapat Anda minta informasi kapan saja menggunakan AWS CLI atau AWS SDK.
Permintaan bersamaan untuk informasi tentang proyek build	Setiap Wilayah yang didukung: 100	Tidak	Jumlah maksimum proyek build yang dapat Anda minta informasi kapan saja menggunakan AWS CLI atau SDK. AWS

Nama	Default	Dapat disesu-an	Deskripsi
Menjalankan build secara bersamaan untuk lingkungan ARM Lambda/10GB	Setiap Wilayah yang didukung: 1	<a href="#">Ya</a>	Jumlah maksimum build yang berjalan secara bersamaan untuk lingkungan ARM Lambda/10GB
Menjalankan build secara bersamaan untuk lingkungan ARM Lambda/1GB	Setiap Wilayah yang didukung: 1	<a href="#">Ya</a>	Jumlah maksimum build yang berjalan secara bersamaan untuk lingkungan ARM Lambda/1GB
Menjalankan build secara bersamaan untuk lingkungan ARM Lambda/2GB	Setiap Wilayah yang didukung: 1	<a href="#">Ya</a>	Jumlah maksimum build yang berjalan secara bersamaan untuk lingkungan ARM Lambda/2GB
Menjalankan build secara bersamaan untuk lingkungan ARM Lambda/4GB	Setiap Wilayah yang didukung: 1	<a href="#">Ya</a>	Jumlah maksimum build yang berjalan secara bersamaan untuk lingkungan ARM Lambda/4GB
Menjalankan build secara bersamaan untuk lingkungan ARM Lambda/8GB	Setiap Wilayah yang didukung: 1	<a href="#">Ya</a>	Jumlah maksimum build yang berjalan secara bersamaan untuk lingkungan ARM Lambda/8GB
Menjalankan build secara bersamaan untuk Arm/lingkungan besar	Setiap Wilayah yang didukung: 1	<a href="#">Ya</a>	Jumlah maksimum build yang berjalan secara bersamaan untuk ARM/lingkungan yang besar

Nama	Default	Dapat disesu an	Deskripsi
Menjalankan build secara bersamaan untuk Arm/lingkungan kecil	Setiap Wilayah yang didukung: 1	<a href="#">Ya</a>	Jumlah maksimum build yang berjalan secara bersamaan untuk Arm/lingkungan kecil
Menjalankan build secara bersamaan untuk lingkungan Linux GPU Large	Setiap Wilayah yang didukung: 0	<a href="#">Ya</a>	Jumlah maksimum build yang berjalan secara bersamaan untuk GPU Linux/lingkungan besar
Menjalankan build secara bersamaan untuk lingkungan Linux GPU Small	Setiap Wilayah yang didukung: 0	<a href="#">Ya</a>	Jumlah maksimum build yang berjalan secara bersamaan untuk GPU Linux/lingkungan kecil
Menjalankan build secara bersamaan untuk lingkungan Linux Lambda/10GB	Setiap Wilayah yang didukung: 1	<a href="#">Ya</a>	Jumlah maksimum build yang berjalan secara bersamaan untuk lingkungan Linux Lambda/10GB
Menjalankan build secara bersamaan untuk lingkungan Linux Lambda/1GB	Setiap Wilayah yang didukung: 1	<a href="#">Ya</a>	Jumlah maksimum build yang berjalan secara bersamaan untuk lingkungan Linux Lambda/1GB
Menjalankan build secara bersamaan untuk lingkungan Linux Lambda/2GB	Setiap Wilayah yang didukung: 1	<a href="#">Ya</a>	Jumlah maksimum build yang berjalan secara bersamaan untuk lingkungan Linux Lambda/2GB

Nama	Default	Dapat disesu an	Deskripsi
Menjalankan build secara bersamaan untuk lingkungan Linux Lambda/4GB	Setiap Wilayah yang didukung: 1	<a href="#">Ya</a>	Jumlah maksimum build yang berjalan secara bersamaan untuk lingkungan Linux Lambda/4GB
Menjalankan build secara bersamaan untuk lingkungan Linux Lambda/8GB	Setiap Wilayah yang didukung: 1	<a href="#">Ya</a>	Jumlah maksimum build yang berjalan secara bersamaan untuk lingkungan Linux Lambda/8GB
Menjalankan build secara bersamaan untuk lingkungan Linux/2xLarge	Setiap Wilayah yang didukung: 0	<a href="#">Ya</a>	Jumlah maksimum build yang berjalan secara bersamaan untuk lingkungan Linux/2xLarge
Menjalankan build secara bersamaan untuk lingkungan Linux/Large	Setiap Wilayah yang didukung: 1	<a href="#">Ya</a>	Jumlah maksimum build yang berjalan secara bersamaan untuk lingkungan Linux/Large
Menjalankan build secara bersamaan untuk lingkungan Linux/Medium	Setiap Wilayah yang didukung: 1	<a href="#">Ya</a>	Jumlah maksimum build yang berjalan secara bersamaan untuk lingkungan Linux/Medium
Menjalankan build secara bersamaan untuk lingkungan Linux/Small	Setiap Wilayah yang didukung: 1	<a href="#">Ya</a>	Jumlah maksimum build yang berjalan secara bersamaan untuk lingkungan Linux/Small

Nama	Default	Dapat disesu an	Deskripsi
Menjalankan build secara bersamaan untuk lingkungan Linux/XLarge	Setiap Wilayah yang didukung: 1	<a href="#">Ya</a>	Jumlah maksimum build yang berjalan secara bersamaan untuk lingkungan Linux/XLarge
Menjalankan build secara bersamaan untuk Windows Server 2019/Lingkungan besar	Setiap Wilayah yang didukung: 1	<a href="#">Ya</a>	Jumlah maksimum build yang berjalan secara bersamaan untuk Windows Server 2019/Lingkungan besar
Menjalankan build secara bersamaan untuk lingkungan Windows Server 2019/Medium	Setiap Wilayah yang didukung: 1	<a href="#">Ya</a>	Jumlah maksimum build yang berjalan secara bersamaan untuk lingkungan Windows Server 2019/Medium
Menjalankan build secara bersamaan untuk lingkungan Windows/Large	Setiap Wilayah yang didukung: 1	<a href="#">Ya</a>	Jumlah maksimum build yang berjalan secara bersamaan untuk lingkungan Windows/Large
Menjalankan build secara bersamaan untuk lingkungan Windows/Medium	Setiap Wilayah yang didukung: 1	<a href="#">Ya</a>	Jumlah maksimum build yang berjalan secara bersamaan untuk lingkungan Windows/Medium
Periode minimum untuk batas waktu build dalam menit	Setiap Wilayah yang didukung: 5	Tidak	Batas waktu build minimum dalam hitungan menit



Nama	Default	Dapat disesu an	Deskripsi
Grup keamanan di bawah konfigurasi VPC	Setiap Wilayah yang didukung: 5	Tidak	Grup keamanan tersedia untuk konfigurasi VPC
Subnet di bawah konfigurasi VPC	Setiap Wilayah yang didukung: 16	Tidak	Subnet tersedia untuk konfigurasi VPC

### Note

Metrik internal akan menentukan kuota default untuk build yang berjalan bersamaan.

Kuota untuk jumlah maksimum build yang berjalan bersamaan bervariasi, tergantung pada jenis komputasi. Untuk beberapa platform dan jenis komputasi, defaultnya adalah 20. Untuk meminta kuota build bersamaan yang lebih tinggi, atau jika Anda mendapatkan kesalahan “Tidak dapat memiliki lebih dari X build aktif untuk akun”, gunakan tautan di atas untuk membuat permintaan. Untuk informasi lebih lanjut tentang harga, lihat [AWS CodeBuildharga](#).

## Batas lainnya

### Proyek Build

Sumber daya	Default
Karakter yang diizinkan dalam deskripsi proyek build	Setiap
Karakter yang diizinkan dalam nama proyek build	Huruf A-Z dan a-z, angka-angka 0-9, dan karakter khusus - dan _
Panjang nama proyek build	2 hingga 255 karakter, inklusif
Panjang maksimum deskripsi proyek build	255 karakter

Sumber daya	Default
Jumlah maksimum laporan yang dapat Anda tambahkan ke proyek	5
Jumlah menit yang dapat Anda tentukan dalam proyek build untuk batas waktu build semua build terkait	5 hingga 480 (8 jam)

## Bangunan

Sumber daya	Default
Waktu maksimum sejarah bangunan dipertahankan	1 tahun
Jumlah menit yang dapat Anda tentukan untuk batas waktu build dari satu build	5 hingga 480 (8 jam)

## Hitung armada

Sumber daya	Default
Jumlah armada komputasi bersamaan	10
Menjalankan instance secara bersamaan untuk armada Arm/lingkungan kecil	1
Menjalankan instance secara bersamaan untuk armada ARM/lingkungan besar	1
Menjalankan instance secara bersamaan untuk armada lingkungan Linux/Small	1
Menjalankan instance secara bersamaan untuk armada lingkungan Linux/Medium	1

Sumber daya	Default
Menjalankan instance secara bersamaan untuk armada lingkungan Linux/Large	1
Menjalankan instance secara bersamaan untuk armada lingkungan Linux/XLarge	1
Menjalankan instance secara bersamaan untuk armada lingkungan Linux/2xLarge	0
Menjalankan instance secara bersamaan untuk GPU Linux/armada lingkungan kecil	0
Menjalankan instance secara bersamaan untuk GPU Linux/armada lingkungan besar	0
Instans yang berjalan bersamaan untuk armada lingkungan Windows Server 2019/Medium	1
Instans yang berjalan bersamaan untuk Windows Server 2019/Armada lingkungan besar	1
Instans yang berjalan bersamaan untuk armada lingkungan Windows Server 2022/Medium	1
Instans yang berjalan bersamaan untuk Windows Server 2022/Armada lingkungan besar	1

## Laporan

Sumber daya	Default
Durasi maksimum laporan pengujian tersedia setelah dibuat	30 hari

Sumber daya	Default
Panjang maksimum pesan kasus uji	5.000 karakter
Panjang maksimum nama kasus uji	1.000 karakter
Jumlah maksimum grup laporan per AWS akun	5.000
Jumlah maksimum kasus uji per laporan	500

## Tanda

Batas tag berlaku untuk tag pada proyek CodeBuild pembuatan dan sumber daya grup CodeBuild laporan.

Sumber daya	Default
Nama kunci tag sumber daya	<p>Kombinasi huruf Unicode, angka, spasi, dan karakter yang diizinkan dalam UTF-8 dengan panjang antara 1 dan 127 karakter. Karakter yang diizinkan adalah + - = . _ : / @</p> <p>Nama kunci tag harus unik, dan setiap kunci hanya dapat memiliki satu nilai. Nama kunci tag tidak dapat:</p> <ul style="list-style-type: none"> <li>dimulai dengan aws :</li> <li>hanya terdiri dari spasi</li> <li>diakhiri dengan spasi</li> <li>berisi spasi atau setiap karakter berikut: ? ^ * [ \ ~ ! # \$ % &amp; * ( ) &gt; &lt;   " ' ` [ ] { } ;</li> </ul>
Nilai tag sumber daya	<p>Kombinasi huruf Unicode, angka, spasi, dan karakter yang diizinkan dalam UTF-8 dengan panjang antara 0 dan 255 karakter. Karakter yang diizinkan adalah + - = . _ : / @</p>

Sumber daya	Default
	<p>Kunci hanya dapat memiliki satu nilai, tetapi banyak kunci dapat memiliki nilai yang sama.</p> <p>Nilai kunci tag tidak dapat berisi emoji atau salah satu karakter berikut: ? ^ * [ \ ~ ! # \$ % &amp; * ( ) &gt; &lt;   " ' ` [ ] { } ;</p>

# Pemberitahuan pihak ketiga AWS CodeBuild untuk Windows

Saat Anda menggunakan CodeBuild untuk build Windows, Anda memiliki opsi untuk menggunakan beberapa paket dan modul pihak ketiga untuk mengaktifkan aplikasi bawaan Anda berjalan di sistem operasi Microsoft Windows dan untuk beroperasi dengan beberapa produk pihak ketiga. Daftar berikut berisi ketentuan hukum pihak ketiga yang berlaku yang mengatur penggunaan Anda atas paket dan modul pihak ketiga yang ditentukan.

## Topik

- [1\) gambar Docker dasar—windowsservercore](#)
- [2\) gambar Docker berbasis jendela — choco](#)
- [3\) gambar Docker berbasis jendela — git --version 2.16.2](#)
- [4\) gambar Docker dasar jendela— --version 15.0.26320.2 microsoft-build-tools](#)
- [5\) gambar Docker berbasis jendela — nuget.commandline --versi 4.5.1](#)
- [7\) gambar Docker berbasis jendela — netfx-4.6.2-devpack](#)
- [8\) gambar Docker dasar jendela — visualfsharpools, v 4.0](#)
- [9\) gambar Docker dasar jendela— -4.6 netfx-pcl-reference-assemblies](#)
- [10\) gambar Docker dasar jendela—visualcppbuildtools v 14.0.25420.1](#)
- [11\) gambar Docker dasar jendela- 3-ondemand-package.cab microsoft-windows-netfx](#)
- [12\) gambar Docker berbasis jendela — dotnet-sdk](#)

## 1) gambar Docker dasar—windowsservercore

(ketentuan lisensi tersedia di: [https://hub.docker.com/\\_/microsoft-windows-servercore](https://hub.docker.com/_/microsoft-windows-servercore))

Lisensi: Dengan meminta dan menggunakan Container OS Image untuk kontainer Windows ini, Anda mengakui, memahami, dan menyetujui Ketentuan Lisensi Tambahan berikut:

### PERSYARATAN LISENSI TAMBAHAN PERANGKAT LUNAK MICROSOFT

#### GAMBAR OS KONTAINER

Microsoft Corporation (atau berdasarkan tempat tinggal Anda, salah satu afiliasinya) (direferensikan sebagai “kami,” “kami”, atau “Microsoft”) melisensikan suplemen Gambar Container OS ini kepada

Anda (“Tambahan”). Anda dilisensikan untuk menggunakan Suplemen ini bersama dengan perangkat lunak sistem operasi host yang mendasarinya (“Perangkat Lunak Host”) semata-mata untuk membantu menjalankan fitur kontainer di Perangkat Lunak Host. Ketentuan lisensi Perangkat Lunak Host berlaku untuk penggunaan Suplemen oleh Anda. Anda tidak boleh menggunakannya jika Anda tidak memiliki lisensi untuk Perangkat Lunak Host. Anda dapat menggunakan Suplemen ini dengan setiap salinan Perangkat Lunak Host yang dilisensikan secara sah.

## PERSYARATAN LISENSI TAMBAHAN DAN/ATAU HAK PENGGUNAAN

Penggunaan Anda atas Suplemen sebagaimana ditentukan dalam paragraf sebelumnya dapat mengakibatkan pembuatan atau modifikasi gambar kontainer (“Gambar Kontainer”) yang mencakup komponen Tambahan tertentu. Untuk kejelasan, Gambar Kontainer terpisah dan berbeda dari mesin virtual atau gambar alat virtual. Sesuai dengan ketentuan lisensi ini, kami memberi Anda hak terbatas untuk mendistribusikan kembali komponen Suplemen tersebut dalam kondisi berikut:

- (i) Anda dapat menggunakan komponen Suplemen hanya seperti yang digunakan dalam, dan sebagai bagian dari Gambar Kontainer Anda,
- (ii) Anda dapat menggunakan komponen Suplemen tersebut dalam Gambar Kontainer Anda selama Anda memiliki fungsionalitas utama yang signifikan dalam Gambar Kontainer Anda yang secara material terpisah dan berbeda dari Suplemen; dan
- (iii) Anda setuju untuk menyertakan ketentuan lisensi ini (atau ketentuan serupa yang diwajibkan oleh kami atau hoster) dengan Gambar Kontainer Anda untuk melisensikan dengan benar kemungkinan penggunaan komponen Tambahan oleh pengguna akhir Anda.

Kami memiliki semua hak lain yang tidak secara tegas diberikan di sini.

Dengan menggunakan Suplemen ini, Anda menerima persyaratan ini. Jika Anda tidak menerimanya, jangan gunakan Suplemen ini.

[Sebagai bagian dari Ketentuan Lisensi Tambahan untuk Container OS Image for Windows Container ini, Anda juga tunduk pada persyaratan lisensi perangkat lunak host Windows Server yang mendasarinya, yang terletak di: <https://www.microsoft.com/en-us/useterms>.](#)

## 2) gambar Docker berbasis jendela — choco

(ketentuan lisensi tersedia di: <https://github.com/chocolatey/choco/blob/master/LICENSE>)

Hak Cipta 2011 - RealDimensions Perangkat Lunak Saat Ini, LLC

Dilisensikan di bawah Lisensi Apache, versi 2.0 (“Lisensi”); Anda tidak boleh menggunakan file-file ini kecuali sesuai dengan Lisensi. Anda dapat memperoleh salinan Lisensi di

<http://www.apache.org/licenses/LICENSE-2.0>

Kecuali diwajibkan oleh hukum yang berlaku atau sebagaimana disepakati secara tertulis, perangkat lunak yang didistribusikan di bawah Lisensi didistribusikan berdasarkan “SEBAGAIMANA ADANYA”, TANPA JAMINAN ATAU KETENTUAN APA PUN, baik tersurat maupun tersirat. Lihat Lisensi untuk bahasa tertentu yang mengatur izin dan batasan di bawah Lisensi.

### 3) gambar Docker berbasis jendela — git --version 2.16.2

(ketentuan lisensi tersedia di: <https://chocolatey.org/packages/git/2.16.2>)

Berlisensi di bawah Lisensi Publik Umum GNU, versi 2, tersedia di: <https://www.gnu.org/licenses/old-licenses/gpl-2.0.html>.

### 4) gambar Docker dasar jendela— --version 15.0.26320.2 microsoft-build-tools

(ketentuan lisensi tersedia di: <https://www.visualstudio.com/license-terms/mt171552/>)

MICROSOFT VISUAL STUDIO 2015 EXTENSIONS, VISUAL STUDIO SHELLS dan C ++  
REDISTRIBUTABLE

-----

Ketentuan lisensi ini adalah perjanjian antara Microsoft Corporation (atau berdasarkan tempat tinggal Anda, salah satu afiliasinya) dan Anda. Mereka berlaku untuk perangkat lunak yang disebutkan di atas. Persyaratan ini juga berlaku untuk setiap layanan Microsoft atau pembaruan untuk perangkat lunak, kecuali sejauh persyaratan tersebut memiliki persyaratan tambahan.

-----

JIKA ANDA MEMATUHI PERSYARATAN LISENSI INI, ANDA MEMILIKI HAK DI BAWAH INI.

1. HAK INSTALASI DAN PENGGUNAAN. Anda dapat menginstal dan menggunakan sejumlah salinan perangkat lunak.
2. KETENTUAN UNTUK KOMPONEN TERTENTU.



- a. **Utilitas.** Perangkat lunak ini mungkin berisi beberapa item pada Daftar Utilitas di <https://docs.microsoft.com/en-us/visualstudio/productinfo/2015-redistribution-vs>. Anda dapat menyalin dan menginstal item tersebut, jika disertakan dengan perangkat lunak, ke milik Anda atau mesin pihak ketiga lainnya, untuk men-debug dan menyebarkan aplikasi dan database yang Anda kembangkan dengan perangkat lunak. Harap dicatat bahwa Utilitas dirancang untuk penggunaan sementara, bahwa Microsoft mungkin tidak dapat menambal atau memperbaiki Utilitas secara terpisah dari perangkat lunak lainnya, dan bahwa beberapa Utilitas menurut sifatnya memungkinkan orang lain untuk mengakses mesin tempat mereka diinstal. Akibatnya, Anda harus menghapus semua Utilitas yang telah Anda instal setelah Anda selesai men-debug atau menyebarkan aplikasi dan database Anda. Microsoft tidak bertanggung jawab atas penggunaan pihak ketiga atau akses Utilitas yang Anda instal di mesin apa pun.
- b. **Platform Microsoft.** Perangkat lunak ini dapat mencakup komponen dari Microsoft Windows; Microsoft Windows Server; Microsoft SQL Server; Microsoft Exchange; Microsoft Office; dan Microsoft SharePoint. Komponen-komponen ini diatur oleh perjanjian terpisah dan kebijakan dukungan produk mereka sendiri, seperti yang dijelaskan dalam ketentuan lisensi yang ditemukan di direktori instalasi untuk komponen itu atau di folder “Lisensi” yang menyertai perangkat lunak.
- c. **Komponen Pihak Ketiga.** Perangkat lunak ini dapat mencakup komponen pihak ketiga dengan pemberitahuan hukum terpisah atau diatur oleh perjanjian lain, seperti yang dijelaskan dalam ThirdPartyNotices file yang menyertai perangkat lunak. Bahkan jika komponen tersebut diatur oleh perjanjian lain, penafian dan batasan dan pengecualian kerusakan di bawah ini juga berlaku. Perangkat lunak ini juga dapat mencakup komponen yang dilisensikan di bawah lisensi open source dengan kewajiban ketersediaan kode sumber. Salinan lisensi tersebut, jika berlaku, disertakan dalam ThirdPartyNotices file. Anda dapat memperoleh kode sumber ini dari kami, jika dan sebagaimana diperlukan berdasarkan lisensi open source yang relevan, dengan mengirimkan wesel atau cek sebesar \$5,00 ke: Tim Kepatuhan Kode Sumber, Microsoft Corporation, 1 Microsoft Way, Redmond, WA 98052. Harap tulis kode sumber untuk satu atau lebih komponen yang tercantum di bawah ini di baris memo pembayaran Anda:
  - Alat Jarak Jauh untuk Visual Studio 2015;
  - Profiler Mandiri untuk Visual Studio 2015;
  - IntelliTraceCollector untuk Visual Studio 2015;
  - Microsoft VC++ Dapat Didistribusikan Ulang 2015;
  - Perpustakaan MFC Multibyte untuk Visual Studio 2015;
  - Alat Microsoft Build 2015;

- Umpan Balik Klien;
- Visual Studio 2015 Shell Terpadu; atau
- Visual Studio 2015 Cangkang Terisolasi.

Kami juga dapat membuat salinan kode sumber yang tersedia di <http://thirdpartysource.microsoft.com>.

3. DATA. Perangkat lunak dapat mengumpulkan informasi tentang Anda dan penggunaan perangkat lunak oleh Anda, dan mengirimkannya ke Microsoft. Microsoft dapat menggunakan informasi ini untuk menyediakan layanan dan meningkatkan produk dan layanan kami. Anda dapat memilih keluar dari banyak skenario ini, tetapi tidak semua, seperti yang dijelaskan dalam dokumentasi produk. Ada juga beberapa fitur dalam perangkat lunak yang memungkinkan Anda mengumpulkan data dari pengguna aplikasi Anda. Jika Anda menggunakan fitur ini untuk mengaktifkan pengumpulan data dalam aplikasi Anda, Anda harus mematuhi hukum yang berlaku, termasuk memberikan pemberitahuan yang sesuai kepada pengguna aplikasi Anda. Anda dapat mempelajari lebih lanjut tentang pengumpulan dan penggunaan data dalam dokumentasi bantuan dan pernyataan privasi di <https://privacy.microsoft.com/en-us/privacystatement>. Penggunaan Anda atas perangkat lunak beroperasi sebagai persetujuan Anda terhadap praktik ini.
4. RUANG LINGKUP LISENSI. Perangkat lunak ini berlisensi, tidak dijual. Perjanjian ini hanya memberi Anda beberapa hak untuk menggunakan perangkat lunak. Microsoft memiliki semua hak lainnya. Kecuali hukum yang berlaku memberi Anda lebih banyak hak terlepas dari batasan ini, Anda dapat menggunakan perangkat lunak hanya sebagaimana diizinkan secara tegas dalam perjanjian ini. Dengan demikian, Anda harus mematuhi batasan teknis apa pun dalam perangkat lunak yang hanya memungkinkan Anda menggunakannya dengan cara tertentu. Anda mungkin tidak
  - mengatasi keterbatasan teknis apa pun dalam perangkat lunak;
  - merekayasa balik, mendekompilasi atau membongkar perangkat lunak, atau mencoba melakukannya, kecuali dan hanya sejauh yang disyaratkan oleh ketentuan lisensi pihak ketiga yang mengatur penggunaan komponen sumber terbuka tertentu yang mungkin disertakan dengan perangkat lunak;
  - menghapus, meminimalkan, memblokir, atau memodifikasi pemberitahuan Microsoft atau pemasoknya dalam perangkat lunak;
  - menggunakan perangkat lunak dengan cara apa pun yang melanggar hukum; atau
  - berbagi, menerbitkan, menyewakan atau menyewakan perangkat lunak, atau menyediakan perangkat lunak sebagai host yang berdiri sendiri sebagai solusi untuk digunakan orang lain.

5. **PEMBATASAN EKSPOR.** Anda harus mematuhi semua undang-undang dan peraturan ekspor domestik dan internasional yang berlaku untuk perangkat lunak, yang mencakup pembatasan tujuan, pengguna akhir, dan penggunaan akhir. Untuk informasi lebih lanjut tentang pembatasan ekspor, kunjungi ([aka.ms/exporting](http://aka.ms/exporting)).
6. **LAYANAN DUKUNGAN.** Karena perangkat lunak ini “sebagaimana adanya,” kami mungkin tidak menyediakan layanan dukungan untuk itu.
7. **SELURUH PERJANJIAN.** Perjanjian ini, dan ketentuan untuk suplemen, pembaruan, layanan berbasis Internet, dan layanan dukungan yang Anda gunakan, adalah keseluruhan perjanjian untuk perangkat lunak dan layanan dukungan.
8. **HUKUM YANG BERLAKU.** Jika Anda memperoleh perangkat lunak di Amerika Serikat, hukum Washington berlaku untuk interpretasi dan klaim atas pelanggaran perjanjian ini, dan hukum negara tempat Anda tinggal berlaku untuk semua klaim lainnya. Jika Anda memperoleh perangkat lunak di negara lain, hukumnya berlaku.
9. **HAK KONSUMEN; VARIASI REGIONAL.** Perjanjian ini menjelaskan hak-hak hukum tertentu. Anda mungkin memiliki hak lain, termasuk hak-hak konsumen, berdasarkan hukum negara bagian atau negara Anda. Terpisah dan terpisah dari hubungan Anda dengan Microsoft, Anda mungkin juga memiliki hak sehubungan dengan pihak tempat Anda memperoleh perangkat lunak. Perjanjian ini tidak mengubah hak-hak lainnya jika hukum negara bagian atau negara Anda tidak mengizinkannya untuk melakukannya. Misalnya, jika Anda memperoleh perangkat lunak di salah satu wilayah di bawah ini, atau hukum negara wajib berlaku, maka ketentuan berikut berlaku untuk Anda:
  - a. **Australia.** Anda memiliki jaminan hukum berdasarkan Undang-Undang Konsumen Australia dan tidak ada dalam perjanjian ini yang dimaksudkan untuk memengaruhi hak-hak tersebut.
  - b. **Kanada.** Jika Anda memperoleh perangkat lunak ini di Kanada, Anda dapat berhenti menerima pembaruan dengan mematikan fitur pembaruan otomatis, memutuskan sambungan perangkat Anda dari Internet (jika dan ketika Anda terhubung kembali ke Internet, namun, perangkat lunak akan melanjutkan memeriksa dan menginstal pembaruan), atau menghapus instalasi perangkat lunak. Dokumentasi produk, jika ada, juga dapat menentukan cara menonaktifkan pembaruan untuk perangkat atau perangkat lunak tertentu Anda.
  - c. **Jerman dan Austria.**
    - i. **Garansi.** Perangkat lunak berlisensi yang benar akan bekerja secara substansif seperti yang dijelaskan dalam materi Microsoft apa pun yang menyertai perangkat lunak. Namun, Microsoft tidak memberikan jaminan kontrak sehubungan dengan perangkat lunak berlisensi.
    - ii. **Batasan Tanggung Jawab.** Dalam hal perilaku yang disengaja, kelalaian berat, klaim berdasarkan Undang-Undang Kewajiban Produk, serta, dalam kasus kematian atau

cedera pribadi atau fisik, Microsoft bertanggung jawab sesuai dengan hukum perundang-undang. Tunduk pada klausul di atas (ii), Microsoft hanya akan bertanggung jawab atas kelalaian ringan jika Microsoft melanggar kewajiban kontrak material tersebut, yang pemenuhannya memfasilitasi pelaksanaan yang wajar dari perjanjian ini, pelanggaran yang akan membahayakan tujuan perjanjian ini dan kepatuhan yang dapat dipercaya secara konstan oleh suatu pihak (jadi- disebut “kewajiban kardinal”). Dalam kasus kelalaian ringan lainnya, Microsoft tidak akan bertanggung jawab atas sedikit kelalaian.

10. PENAFIAN GARANSI. PERANGKAT LUNAK INI DILISENSIKAN “APA ADANYA.” ANDA MENANGGUNG RISIKO MENGGUNAKANNYA. MICROSOFT TIDAK MEMBERIKAN JAMINAN, JAMINAN, ATAU KETENTUAN TERSURAT. SEJAUH DIIZINKAN BERDASARKAN UNDANG-UNDANG SETEMPAT ANDA, MICROSOFT MENGECEUALIKAN JAMINAN TERSIRAT ATAS KELAYAKAN UNTUK MERCHANTABILITY, KESESUAIAN UNTUK TUJUAN TERTENTU, DAN NON-PELANGGARAN.

11. PEMBatasan dan Pengecualian Kerusakan. ANDA DAPAT MEMULIHKAN DARI MICROSOFT DAN PEMASOKNYA HANYA KERUSAKAN LANGSUNG HINGGA US \$5,00. ANDA TIDAK DAPAT MEMULIHKAN KERUSAKAN LAIN, TERMASUK KONSEKUENSIAL, KEHILANGAN KEUNTUNGAN, KERUSAKAN KHUSUS, TIDAK LANGSUNG ATAU INSIDENTAL. Batasan ini berlaku untuk (a) apa pun yang terkait dengan perangkat lunak, layanan, konten (termasuk kode) di situs Internet pihak ketiga, atau aplikasi pihak ketiga; dan (b) klaim atas pelanggaran kontrak, pelanggaran garansi, jaminan atau kondisi, tanggung jawab ketat, kelalaian, atau kesalahan lainnya sejauh diizinkan oleh hukum yang berlaku.

Ini juga berlaku bahkan jika Microsoft tahu atau seharusnya tahu tentang kemungkinan kerusakan. Batasan atau pengecualian di atas mungkin tidak berlaku untuk Anda karena negara Anda mungkin tidak mengizinkan pengecualian atau pembatasan kerusakan insidental, konsekuensial atau lainnya.

ID EULA: ShellsRedist VS2015\_UPDATE3\_ \_ <ENU>

## 5) gambar Docker berbasis jendela — nuget.commandline --versi 4.5.1

(ketentuan lisensi tersedia di: <https://github.com/NuGet/Home/blob/dev/LICENSE.txt>)

Hak Cipta (c) .NET Foundation. Semua hak dilindungi undang-undang.

Dilisensikan di bawah Lisensi Apache, versi 2.0 (“Lisensi”); Anda tidak boleh menggunakan file-file ini kecuali sesuai dengan Lisensi. Anda dapat memperoleh salinan Lisensi di

<http://www.apache.org/licenses/LICENSE-2.0>

Kecuali diwajibkan oleh hukum yang berlaku atau sebagaimana disepakati secara tertulis, perangkat lunak yang didistribusikan di bawah Lisensi didistribusikan berdasarkan “SEBAGAIMANA ADANYA”, TANPA JAMINAN ATAU KETENTUAN APA PUN, baik tersurat maupun tersirat. Lihat Lisensi untuk bahasa tertentu yang mengatur izin dan batasan di bawah Lisensi.

## 7) gambar Docker berbasis jendela — netfx-4.6.2-devpack

### PERSYARATAN LISENSI TAMBAHAN PERANGKAT LUNAK MICROSOFT

#### .NET FRAMEWORK DAN PAKET BAHASA TERKAIT UNTUK SISTEM OPERASI MICROSOFT WINDOWS

-----

Microsoft Corporation (atau berdasarkan tempat tinggal Anda, salah satu afiliasinya) melisensikan suplemen ini kepada Anda. Jika Anda memiliki lisensi untuk menggunakan perangkat lunak sistem operasi Microsoft Windows (“perangkat lunak”), Anda dapat menggunakan suplemen ini. Anda tidak boleh menggunakannya jika Anda tidak memiliki lisensi untuk perangkat lunak. Anda dapat menggunakan suplemen ini dengan setiap salinan perangkat lunak berlisensi sah.

Ketentuan lisensi berikut menjelaskan istilah penggunaan tambahan untuk suplemen ini. Ketentuan ini dan ketentuan lisensi untuk perangkat lunak berlaku untuk penggunaan suplemen Anda. Jika ada konflik, ketentuan lisensi tambahan ini berlaku.

DENGAN MENGGUNAKAN SUPLEMEN INI, ANDA MENERIMA PERSYARATAN INI. JIKA ANDA TIDAK MENERIMANYA, JANGAN GUNAKAN SUPLEMEN INI.

-----

Jika Anda mematuhi persyaratan lisensi ini, Anda memiliki hak di bawah ini.

1. KODE YANG DAPAT DIDISTRIBUSIKAN. Suplemen ini terdiri dari Kode Distribusi. “Kode Distribusi” adalah kode yang diizinkan untuk Anda distribusikan dalam program yang Anda kembangkan jika Anda mematuhi persyaratan di bawah ini.
  - a. Hak untuk menggunakan dan mendistribusikan.
    - Anda dapat menyalin dan mendistribusikan bentuk kode objek suplemen.

- Distribusi Pihak Ketiga. Anda dapat mengizinkan distributor program Anda untuk menyalin dan mendistribusikan Kode Distribusi sebagai bagian dari program tersebut.
- b. Persyaratan Distribusi. Untuk setiap Kode Distribusi yang Anda distribusikan, Anda harus
- tambahkan fungsionalitas utama yang signifikan ke dalam program Anda;
  - untuk setiap Kode Distribusi yang memiliki ekstensi nama file.lib, hanya mendistribusikan hasil menjalankan Kode Distribusi tersebut melalui tautan dengan program Anda;
  - mendistribusikan Kode Distribusi yang disertakan dalam program penyiapan hanya sebagai bagian dari program penyiapan itu tanpa modifikasi;
  - mewajibkan distributor dan pengguna akhir eksternal untuk menyetujui persyaratan yang melindunginya setidaknya sebanyak perjanjian ini;
  - menampilkan pemberitahuan hak cipta Anda yang valid pada program Anda; dan
  - mengganti rugi, membela, dan membebaskan Microsoft dari klaim apa pun, termasuk biaya pengacara, yang terkait dengan distribusi atau penggunaan program Anda.
- c. Pembatasan Distribusi. Anda mungkin tidak
- mengubah hak cipta, merek dagang, atau pemberitahuan paten apa pun dalam Kode Distribusi;
  - menggunakan merek dagang Microsoft dalam nama program Anda atau dengan cara yang menunjukkan program Anda berasal atau didukung oleh Microsoft;
  - mendistribusikan Kode Distributable untuk dijalankan pada platform selain platform Windows;
  - menyertakan Kode Distribusi dalam program jahat, menipu atau melanggar hukum; atau
  - memodifikasi atau mendistribusikan kode sumber dari setiap Kode Distribusi sehingga bagian mana pun darinya menjadi tunduk pada Lisensi yang Dikecualikan. Lisensi yang Dikecualikan adalah lisensi yang mensyaratkan, sebagai syarat penggunaan, modifikasi atau distribusi, bahwa
    - kode diungkapkan atau didistribusikan dalam bentuk kode sumber; atau
    - yang lain memiliki hak untuk memodifikasinya.
2. LAYANAN DUKUNGAN UNTUK SUPLEMEN. Microsoft menyediakan layanan dukungan untuk perangkat lunak ini seperti yang dijelaskan di [www.support.microsoft.com/common/international.aspx](http://www.support.microsoft.com/common/international.aspx).

## 8) gambar Docker dasar jendela — visualfsharpools, v 4.0

(ketentuan lisensi tersedia di: <https://github.com/dotnet/fsharp/blob/main/License.txt>)

Hak Cipta (c) Microsoft Corporation. Semua hak dilindungi undang-undang.

Dilisensikan di bawah Lisensi Apache, versi 2.0 (“Lisensi”); Anda tidak boleh menggunakan file-file ini kecuali sesuai dengan Lisensi. Anda dapat memperoleh salinan Lisensi di

<http://www.apache.org/licenses/LICENSE-2.0>

Kecuali diwajibkan oleh hukum yang berlaku atau sebagaimana disepakati secara tertulis, perangkat lunak yang didistribusikan di bawah Lisensi didistribusikan berdasarkan “SEBAGAIMANA ADANYA”, TANPA JAMINAN ATAU KETENTUAN APA PUN, baik tersurat maupun tersirat. Lihat Lisensi untuk bahasa tertentu yang mengatur izin dan batasan di bawah Lisensi.

## 9) gambar Docker dasar jendela— -4.6 netfx-pcl-reference-assemblies

PERSYARATAN LISENSI PERANGKAT LUNAK MICROSOFT

RAKITAN REFERENSI PERPUSTAKAAN KELAS PORTABEL MICROSOFT .NET — 4.6

-----

Ketentuan lisensi ini adalah perjanjian antara Microsoft Corporation (atau berdasarkan tempat tinggal Anda, salah satu afiliasinya) dan Anda. Silakan baca mereka. Mereka berlaku untuk perangkat lunak yang disebutkan di atas. Ketentuan ini juga berlaku untuk Microsoft

- pembaruan,
- suplemen,
- Layanan berbasis internet, dan
- layanan dukungan

untuk perangkat lunak ini, kecuali persyaratan lain menyertai item tersebut. Jika demikian, persyaratan tersebut berlaku.

DENGAN MENGGUNAKAN PERANGKAT LUNAK, ANDA MENERIMA KETENTUAN INI. JIKA ANDA TIDAK MENERIMANYA, JANGAN GUNAKAN PERANGKAT LUNAK.

-----

JIKA ANDA MEMATUHI KETENTUAN LISENSI INI, ANDA MEMILIKI HAK ABADI DI BAWAH INI.

1. HAK INSTALASI DAN PENGGUNAAN. Anda dapat menginstal dan menggunakan sejumlah salinan perangkat lunak untuk merancang, mengembangkan, dan menguji program Anda.
2. PERSYARATAN LISENSI TAMBAHAN DAN/ATAU HAK PENGGUNAAN.
  - a. Kode yang Dapat Didistribusikan. Anda dapat mendistribusikan perangkat lunak dalam program alat pengembang yang Anda kembangkan, untuk memungkinkan pelanggan program Anda mengembangkan pustaka portabel untuk digunakan dengan perangkat atau sistem operasi apa pun, jika Anda mematuhi ketentuan di bawah ini.
    - i. Hak untuk menggunakan dan mendistribusikan. Perangkat lunak ini adalah “Kode Distribusi.”
      - Kode yang Dapat Didistribusikan. Anda dapat menyalin dan mendistribusikan bentuk kode objek perangkat lunak.
      - Distribusi Pihak Ketiga. Anda dapat mengizinkan distributor program Anda untuk menyalin dan mendistribusikan Kode Distribusi sebagai bagian dari program tersebut.
    - ii. Persyaratan Distribusi. Untuk setiap Kode Distribusi yang Anda distribusikan, Anda harus
      - tambahkan fungsionalitas utama yang signifikan ke dalam program Anda;
      - mewajibkan distributor dan pelanggan Anda untuk menyetujui persyaratan yang melindunginya setidaknya sebanyak perjanjian ini;
      - menampilkan pemberitahuan hak cipta Anda yang valid pada program Anda; dan
      - mengganti rugi, membela, dan membebaskan Microsoft dari klaim apa pun, termasuk biaya pengacara, yang terkait dengan distribusi atau penggunaan program Anda.
    - iii. Pembatasan Distribusi. Anda mungkin tidak
      - mengubah hak cipta, merek dagang, atau pemberitahuan paten apa pun dalam Kode Distribusi;
      - menggunakan merek dagang Microsoft dalam nama program Anda atau dengan cara yang menunjukkan program Anda berasal atau didukung oleh Microsoft;
      - menyertakan Kode Distribusi dalam program jahat, menipu atau melanggar hukum; atau
      - memodifikasi atau mendistribusikan Kode Distribusi sehingga bagian mana pun darinya menjadi tunduk pada Lisensi yang Dikecualikan. Lisensi yang Dikecualikan adalah lisensi yang mensyaratkan, sebagai syarat penggunaan, modifikasi atau distribusi, bahwa
        - kode diungkapkan atau didistribusikan dalam bentuk kode sumber; atau
        - yang lain memiliki hak untuk memodifikasinya.
  3. RUANG LINGKUP LISENSI. Perangkat lunak ini berlisensi, tidak dijual. Perjanjian ini hanya memberi Anda beberapa hak untuk menggunakan perangkat lunak. Microsoft memiliki semua hak lainnya. Kecuali hukum yang berlaku memberi Anda lebih banyak hak terlepas dari batasan ini,



Anda dapat menggunakan perangkat lunak hanya sebagaimana diizinkan secara tegas dalam perjanjian ini. Dengan demikian, Anda harus mematuhi batasan teknis apa pun dalam perangkat lunak yang hanya memungkinkan Anda menggunakannya dengan cara tertentu. Anda mungkin tidak

- mengatasi keterbatasan teknis apa pun dalam perangkat lunak;
  - merekayasa balik, mendekompilasi atau membongkar perangkat lunak, kecuali dan hanya sejauh hukum yang berlaku secara tegas mengizinkan, terlepas dari batasan ini;
  - mempublikasikan perangkat lunak untuk disalin orang lain; atau
  - menyewa, menyewakan atau meminjamkan perangkat lunak.
4. UMPAN BALIK. Anda dapat memberikan umpan balik tentang perangkat lunak. Jika Anda memberikan umpan balik tentang perangkat lunak kepada Microsoft, Anda memberikan kepada Microsoft, tanpa biaya, hak untuk menggunakan, berbagi, dan mengkomersialkan umpan balik Anda dengan cara apa pun dan untuk tujuan apa pun. Anda juga memberikan kepada pihak ketiga, tanpa biaya, hak paten apa pun yang diperlukan untuk produk, teknologi, dan layanan mereka untuk digunakan atau berinteraksi dengan bagian tertentu dari perangkat lunak atau layanan Microsoft yang menyertakan umpan balik. Anda tidak akan memberikan umpan balik yang tunduk pada lisensi yang mengharuskan Microsoft untuk melisensikan perangkat lunak atau dokumentasinya kepada pihak ketiga karena kami menyertakan umpan balik Anda di dalamnya. Hak-hak ini bertahan dari perjanjian ini.
  5. TRANSFER KE PIHAK KETIGA. Pengguna pertama perangkat lunak dapat mentransfernya, dan perjanjian ini, langsung ke pihak ketiga. Sebelum transfer, pihak tersebut harus setuju bahwa perjanjian ini berlaku untuk transfer dan penggunaan perangkat lunak. Pengguna pertama harus menghapus instalasi perangkat lunak sebelum mentransfernya secara terpisah dari perangkat. Pengguna pertama tidak boleh menyimpan salinan apa pun.
  6. PEMBATASAN EKSPOR. Perangkat lunak ini tunduk pada undang-undang dan peraturan ekspor Amerika Serikat. Anda harus mematuhi semua undang-undang dan peraturan ekspor domestik dan internasional yang berlaku untuk perangkat lunak. Undang-undang ini mencakup pembatasan tujuan, pengguna akhir, dan penggunaan akhir. Untuk informasi tambahan, lihat [www.microsoft.com/exporting](http://www.microsoft.com/exporting).
  7. LAYANAN DUKUNGAN. Karena perangkat lunak ini “sebagaimana adanya,” kami mungkin tidak menyediakan layanan dukungan untuk itu.
  8. SELURUH PERJANJIAN. Perjanjian ini, dan ketentuan untuk suplemen, pembaruan, layanan berbasis Internet, dan layanan dukungan yang Anda gunakan, adalah keseluruhan perjanjian untuk perangkat lunak dan layanan dukungan apa pun yang kami sediakan.

## 9. HUKUM YANG BERLAKU.

- a. Amerika Serikat. Jika Anda memperoleh perangkat lunak di Amerika Serikat, hukum negara bagian Washington mengatur interpretasi perjanjian ini dan berlaku untuk klaim pelanggaran itu, terlepas dari konflik prinsip hukum. Hukum negara tempat Anda tinggal mengatur semua klaim lainnya, termasuk klaim berdasarkan undang-undang perlindungan konsumen negara bagian, undang-undang persaingan tidak sehat, dan gugatan.
- b. Di luar Amerika Serikat. Jika Anda memperoleh perangkat lunak di negara lain, hukum negara itu berlaku.

10 EFEK HUKUM. Perjanjian ini menjelaskan hak-hak hukum tertentu. Anda mungkin memiliki hak lain di bawah hukum negara Anda. Anda mungkin juga memiliki hak sehubungan dengan pihak dari siapa Anda memperoleh perangkat lunak. Perjanjian ini tidak mengubah hak Anda berdasarkan hukum negara Anda jika hukum negara Anda tidak mengizinkan melakukannya.

11 PENAFIAN GARANSI. PERANGKAT LUNAK INI DILISENSIKAN “APA ADANYA.” ANDA MENANGGUNG RISIKO MENGGUNAKANNYA. MICROSOFT TIDAK MEMBERIKAN JAMINAN, JAMINAN, ATAU KETENTUAN TERSURAT. ANDA MUNGKIN MEMILIKI HAK KONSUMEN TAMBAHAN ATAU JAMINAN HUKUM BERDASARKAN UNDANG-UNDANG SETEMPAT YANG TIDAK DAPAT DIUBAH OLEH PERJANJIAN INI. SEJAUH DIIZINKAN BERDASARKAN UNDANG-UNDANG SETEMPAT ANDA, MICROSOFT MENGECEUALIKAN JAMINAN TERSIRAT ATAS KELAYAKAN UNTUK MERCHANTABILITY, KESESUAIAN UNTUK TUJUAN TERTENTU, DAN NON-PELANGGARAN.

UNTUK AUSTRALIA—ANDA MEMILIKI JAMINAN HUKUM BERDASARKAN HUKUM KONSUMEN AUSTRALIA DAN TIDAK ADA DALAM KETENTUAN INI YANG DIMAKSUDKAN UNTUK MEMENGARUHI HAK-HAK TERSEBUT.

12 PEMBatasan dan Pengecualian Solusi dan Kerusakan. ANDA DAPAT MEMULIHKAN DARI MICROSOFT DAN PEMASOKNYA HANYA KERUSAKAN LANGSUNG HINGGA US \$5,00. ANDA TIDAK DAPAT MEMULIHKAN KERUSAKAN LAIN, TERMASUK KONSEKUENSIAL, KEHILANGAN KEUNTUNGAN, KERUSAKAN KHUSUS, TIDAK LANGSUNG ATAU INSIDENTAL.

Batasan ini berlaku untuk

- segala sesuatu yang terkait dengan perangkat lunak, layanan, konten (termasuk kode) di situs Internet pihak ketiga, atau program pihak ketiga; dan
- klaim atas pelanggaran kontrak, pelanggaran garansi, jaminan atau kondisi, tanggung jawab ketat, kelalaian, atau kesalahan lainnya sejauh diizinkan oleh hukum yang berlaku.

Ini juga berlaku bahkan jika Microsoft tahu atau seharusnya tahu tentang kemungkinan kerusakan. Batasan atau pengecualian di atas mungkin tidak berlaku untuk Anda karena negara Anda mungkin tidak mengizinkan pengecualian atau pembatasan kerusakan insidental, konsekuensial atau lainnya.

## 10) gambar Docker dasar jendela—visualcppbuildtools v 14.0.25420.1

(ketentuan lisensi tersedia di: <https://www.visualstudio.com/license-terms/mt644918/>)

ALAT PEMBUATAN MICROSOFT VISUAL C ++

PERSYARATAN LISENSI PERANGKAT LUNAK MICROSOFT

ALAT PEMBUATAN MICROSOFT VISUAL C ++

-----

Ketentuan lisensi ini adalah perjanjian antara Microsoft Corporation (atau berdasarkan tempat tinggal Anda, salah satu afiliasinya) dan Anda. Mereka berlaku untuk perangkat lunak yang disebutkan di atas. Persyaratan ini juga berlaku untuk setiap layanan Microsoft atau pembaruan untuk perangkat lunak, kecuali sejauh persyaratan tersebut memiliki persyaratan yang berbeda.

-----

JIKA ANDA MEMATUHI KETENTUAN LISENSI INI, ANDA MEMILIKI HAK DI BAWAH INI.

### 1. HAK INSTALASI DAN PENGGUNAAN.

a. Satu pengguna dapat menggunakan salinan perangkat lunak untuk mengembangkan dan menguji aplikasi mereka.

2. DATA. Perangkat lunak dapat mengumpulkan informasi tentang Anda dan penggunaan perangkat lunak oleh Anda, dan mengirimkannya ke Microsoft. Microsoft dapat menggunakan informasi ini untuk menyediakan layanan dan meningkatkan produk dan layanan kami. Anda dapat memilih keluar dari banyak skenario ini, tetapi tidak semua, seperti yang dijelaskan dalam dokumentasi produk. Ada juga beberapa fitur dalam perangkat lunak yang memungkinkan Anda mengumpulkan data dari pengguna aplikasi Anda. Jika Anda menggunakan fitur ini untuk mengaktifkan pengumpulan data dalam aplikasi Anda, Anda harus mematuhi hukum yang berlaku,

termasuk memberikan pemberitahuan yang sesuai kepada pengguna aplikasi Anda. Anda dapat mempelajari lebih lanjut tentang pengumpulan dan penggunaan data dalam dokumentasi bantuan dan pernyataan privasi di <http://go.microsoft.com/fwlink/?LinkID=528096>. Penggunaan Anda atas perangkat lunak beroperasi sebagai persetujuan Anda terhadap praktik ini.

### 3. KETENTUAN UNTUK KOMPONEN TERTENTU.

- a. Bangun Server. Perangkat lunak ini mungkin berisi beberapa komponen Build Server yang tercantum dalam BuildServer file.TXT, dan/atau file apa pun yang tercantum dalam BuildServer daftar yang terletak mengikuti Persyaratan Lisensi Perangkat Lunak Microsoft ini. Anda dapat menyalin dan menginstal item tersebut, jika disertakan dalam perangkat lunak, ke mesin build Anda. Anda dan orang lain di organisasi Anda dapat menggunakan item ini di mesin build Anda semata-mata untuk tujuan mengkompilasi, membangun, memverifikasi, dan mengarsipkan aplikasi Anda atau menjalankan pengujian kualitas atau kinerja sebagai bagian dari proses pembuatan.
  - b. Platform Microsoft. Perangkat lunak ini dapat mencakup komponen dari Microsoft Windows; Microsoft Windows Server; Microsoft SQL Server; Microsoft Exchange; Microsoft Office; dan Microsoft SharePoint. Komponen-komponen ini diatur oleh perjanjian terpisah dan kebijakan dukungan produk mereka sendiri, seperti yang dijelaskan dalam ketentuan lisensi yang ditemukan di direktori instalasi untuk komponen itu atau di folder “Lisensi” yang menyertai perangkat lunak.
  - c. Komponen Pihak Ketiga. Perangkat lunak ini dapat mencakup komponen pihak ketiga dengan pemberitahuan hukum terpisah atau diatur oleh perjanjian lain, seperti yang dijelaskan dalam ThirdPartyNotices file yang menyertai perangkat lunak. Bahkan jika komponen tersebut diatur oleh perjanjian lain, penafian dan batasan dan pengecualian kerusakan di bawah ini juga berlaku.
  - d. Manajer Package. Perangkat lunak ini dapat mencakup manajer paket, seperti Nuget, yang memberi Anda opsi untuk mengunduh paket perangkat lunak Microsoft dan pihak ketiga lainnya untuk digunakan dengan aplikasi Anda. Paket-paket tersebut berada di bawah lisensi mereka sendiri, dan bukan perjanjian ini. Microsoft tidak mendistribusikan, melisensikan, atau memberikan jaminan apa pun untuk paket pihak ketiga mana pun.
4. RUANG LINGKUP LISENSI. Perangkat lunak ini berlisensi, tidak dijual. Perjanjian ini hanya memberi Anda beberapa hak untuk menggunakan perangkat lunak. Microsoft memiliki semua hak lainnya. Kecuali hukum yang berlaku memberi Anda lebih banyak hak terlepas dari batasan ini, Anda dapat menggunakan perangkat lunak hanya sebagaimana diizinkan secara tegas dalam perjanjian ini. Dengan demikian, Anda harus mematuhi batasan teknis apa pun dalam perangkat lunak yang hanya memungkinkan Anda menggunakannya dengan cara tertentu. Untuk informasi

lebih lanjut, lihat <https://docs.microsoft.com/en-us/legal/information-protection/software-license-terms#1-installation-and-use-rights>. Anda mungkin tidak

- mengatasi keterbatasan teknis apa pun dalam perangkat lunak;
  - merekayasa balik, mendekompilasi atau membongkar perangkat lunak, atau mencoba melakukannya, kecuali dan hanya sejauh yang disyaratkan oleh ketentuan lisensi pihak ketiga yang mengatur penggunaan komponen open source tertentu yang mungkin disertakan dengan perangkat lunak;
  - menghapus, meminimalkan, memblokir, atau memodifikasi pemberitahuan Microsoft atau pemasoknya;
  - menggunakan perangkat lunak dengan cara apa pun yang melanggar hukum; atau
  - berbagi, menerbitkan, menyewakan atau menyewakan perangkat lunak, atau menyediakan perangkat lunak sebagai host yang berdiri sendiri sebagai solusi untuk digunakan orang lain.
5. **PEMBATASAN EKSPOR.** Anda harus mematuhi semua undang-undang dan peraturan ekspor domestik dan internasional yang berlaku untuk perangkat lunak, yang mencakup pembatasan tujuan, pengguna akhir, dan penggunaan akhir. Untuk informasi lebih lanjut tentang pembatasan ekspor, kunjungi ([aka.ms/exporting](http://aka.ms/exporting)).
  6. **LAYANAN DUKUNGAN.** Karena perangkat lunak ini “sebagaimana adanya,” kami mungkin tidak menyediakan layanan dukungan untuk itu.
  7. **SELURUH PERJANJIAN.** Perjanjian ini, dan ketentuan untuk suplemen, pembaruan, layanan berbasis Internet, dan layanan dukungan yang Anda gunakan, adalah keseluruhan perjanjian untuk perangkat lunak dan layanan dukungan.
  8. **HUKUM YANG BERLAKU.** Jika Anda memperoleh perangkat lunak di Amerika Serikat, hukum Washington berlaku untuk interpretasi dan klaim atas pelanggaran perjanjian ini, dan hukum negara tempat Anda tinggal berlaku untuk semua klaim lainnya. Jika Anda memperoleh perangkat lunak di negara lain, hukumnya berlaku.
  9. **HAK KONSUMEN; VARIASI REGIONAL.** Perjanjian ini menjelaskan hak-hak hukum tertentu. Anda mungkin memiliki hak lain, termasuk hak-hak konsumen, berdasarkan hukum negara bagian atau negara Anda. Terpisah dan terpisah dari hubungan Anda dengan Microsoft, Anda mungkin juga memiliki hak sehubungan dengan pihak tempat Anda memperoleh perangkat lunak. Perjanjian ini tidak mengubah hak-hak lainnya jika hukum negara bagian atau negara Anda tidak mengizinkannya untuk melakukannya. Misalnya, jika Anda memperoleh perangkat lunak di salah satu wilayah di bawah ini, atau hukum negara wajib berlaku, maka ketentuan berikut berlaku untuk Anda:

- Australia. Anda memiliki jaminan hukum berdasarkan Undang-Undang Konsumen Australia dan tidak ada dalam perjanjian ini yang dimaksudkan untuk memengaruhi hak-hak tersebut.
- Kanada. Jika Anda memperoleh perangkat lunak ini di Kanada, Anda dapat berhenti menerima pembaruan dengan mematikan fitur pembaruan otomatis, memutuskan sambungan perangkat Anda dari Internet (jika dan ketika Anda terhubung kembali ke Internet, namun, perangkat lunak akan melanjutkan memeriksa dan menginstal pembaruan), atau menghapus instalasi perangkat lunak. Dokumentasi produk, jika ada, juga dapat menentukan cara menonaktifkan pembaruan untuk perangkat atau perangkat lunak tertentu Anda.
- Jerman dan Austria.
  - Garansi. Perangkat lunak berlisensi yang benar akan bekerja secara substansif seperti yang dijelaskan dalam materi Microsoft apa pun yang menyertai perangkat lunak. Namun, Microsoft tidak memberikan jaminan kontrak sehubungan dengan perangkat lunak berlisensi.
  - Batasan Tanggung Jawab. Dalam hal perilaku yang disengaja, kelalaian berat, klaim berdasarkan Undang-Undang Kewajiban Produk, serta, dalam kasus kematian atau cedera pribadi atau fisik, Microsoft bertanggung jawab sesuai dengan hukum perundang-undangan.

Tunduk pada klausul di atas (ii), Microsoft hanya akan bertanggung jawab atas kelalaian ringan jika Microsoft melanggar kewajiban kontraktual material tersebut, yang pemenuhannya memfasilitasi pelaksanaan yang wajar dari perjanjian ini, pelanggaran yang akan membahayakan tujuan perjanjian ini dan kepatuhan yang dapat dipercaya oleh suatu pihak secara konstan (disebut “kewajiban utama”). Dalam kasus kelalaian ringan lainnya, Microsoft tidak akan bertanggung jawab atas sedikit kelalaian.

10 EFEK HUKUM. Perjanjian ini menjelaskan hak-hak hukum tertentu. Anda mungkin memiliki hak lain berdasarkan hukum negara bagian atau negara Anda. Perjanjian ini tidak mengubah hak Anda berdasarkan hukum negara bagian atau negara Anda jika hukum negara bagian atau negara Anda tidak mengizinkannya melakukannya. Tanpa batasan dari hal tersebut di atas, untuk Australia, ANDA MEMILIKI JAMINAN HUKUM BERDASARKAN HUKUM KONSUMEN AUSTRALIA DAN TIDAK ADA DALAM KETENTUAN INI YANG DIMAKSUDKAN UNTUK MEMENGARUHI HAK-HAK TERSEBUT

11 PENAFIAN GARANSI. PERANGKAT LUNAK INI DILISENSIKAN “APA ADANYA.” ANDA MENANGGUNG RISIKO MENGGUNAKANNYA. MICROSOFT TIDAK MEMBERIKAN JAMINAN, JAMINAN, ATAU KETENTUAN TERSURAT. SEJAUH DIIZINKAN BERDASARKAN UNDANG-UNDANG SETEMPAT ANDA, MICROSOFT MENGEQUALIKAN JAMINAN TERSIRAT ATAS KELAYAKAN UNTUK MERCHANTABILITY, KESESUAIAN UNTUK TUJUAN TERTENTU, DAN NON-PELANGGARAN.

12. PEMBATASAN DAN PENGECEUALIAN KERUSAKAN. ANDA DAPAT MEMULIHKAN DARI MICROSOFT DAN PEMASOKNYA HANYA KERUSAKAN LANGSUNG HINGGA US \$5,00. ANDA TIDAK DAPAT MEMULIHKAN KERUSAKAN LAIN, TERMASUK KONSEKUENSIAL, KEHILANGAN KEUNTUNGAN, KERUSAKAN KHUSUS, TIDAK LANGSUNG ATAU INSIDENTAL.

Batasan ini berlaku untuk (a) apa pun yang terkait dengan perangkat lunak, layanan, konten (termasuk kode) di situs Internet pihak ketiga, atau aplikasi pihak ketiga; dan (b) klaim atas pelanggaran kontrak, pelanggaran garansi, jaminan atau kondisi, tanggung jawab ketat, kelalaian, atau kesalahan lainnya sejauh diizinkan oleh hukum yang berlaku.

Ini juga berlaku bahkan jika Microsoft tahu atau seharusnya tahu tentang kemungkinan kerusakan. Batasan atau pengecualian di atas mungkin tidak berlaku untuk Anda karena negara Anda mungkin tidak mengizinkan pengecualian atau pembatasan kerusakan insidental, konsekuensial atau lainnya.

## 11) gambar Docker dasar jendela- 3-ondemand-package.cab microsoft-windows-netfx

### PERSYARATAN LISENSI TAMBAHAN PERANGKAT LUNAK MICROSOFT

#### MICROSOFT .NET FRAMEWORK 3.5 SP1 UNTUK SISTEM OPERASI MICROSOFT WINDOWS

-----

Microsoft Corporation (atau berdasarkan tempat tinggal Anda, salah satu afiliasinya) melisensikan suplemen ini kepada Anda. Jika Anda memiliki lisensi untuk menggunakan perangkat lunak sistem operasi Microsoft Windows (yang suplemen ini berlaku) ("perangkat lunak"), Anda dapat menggunakan suplemen ini. Anda tidak boleh menggunakannya jika Anda tidak memiliki lisensi untuk perangkat lunak. Anda dapat menggunakan salinan suplemen ini dengan setiap salinan perangkat lunak berlisensi yang sah.

Ketentuan lisensi berikut menjelaskan istilah penggunaan tambahan untuk suplemen ini. Ketentuan ini dan ketentuan lisensi untuk perangkat lunak berlaku untuk penggunaan suplemen Anda. Jika ada konflik, ketentuan lisensi tambahan ini berlaku.

DENGAN MENGGUNAKAN SUPLEMEN INI, ANDA MENERIMA PERSYARATAN INI. JIKA ANDA TIDAK MENERIMANYA, JANGAN GUNAKAN SUPLEMEN INI.

-----

Jika Anda mematuhi persyaratan lisensi ini, Anda memiliki hak di bawah ini.

1. LAYANAN DUKUNGAN UNTUK SUPLEMEN. Microsoft menyediakan layanan dukungan untuk perangkat lunak ini seperti yang dijelaskan di [www.support.microsoft.com/common/international.aspx](http://www.support.microsoft.com/common/international.aspx).
2. PENGUJIAN BENCHMARK MICROSOFT .NET. Perangkat lunak ini mencakup komponen .NET Framework, Windows Communication Foundation, Windows Presentation Foundation, dan Windows Workflow Foundation dari sistem operasi Windows (.NET Components). Anda dapat melakukan pengujian benchmark internal dari .NET Components. [Anda dapat mengungkapkan hasil tes benchmark dari Komponen .NET, asalkan Anda mematuhi ketentuan yang ditetapkan di http://go.microsoft.com/fwlink/?LinkID=66406.](http://go.microsoft.com/fwlink/?LinkID=66406)

[Terlepas dari perjanjian lain yang mungkin Anda miliki dengan Microsoft, jika Anda mengungkapkan hasil pengujian benchmark tersebut, Microsoft berhak untuk mengungkapkan hasil tes benchmark yang dilakukannya terhadap produk Anda yang bersaing dengan Komponen .NET yang berlaku, asalkan memenuhi ketentuan yang sama yang ditetapkan di http://go.microsoft.com/fwlink/?LinkID=66406.](http://go.microsoft.com/fwlink/?LinkID=66406)

## 12) gambar Docker berbasis jendela — dotnet-sdk

(tersedia di <https://github.com/dotnet/core/blob/main/LICENSE.TXT>)

Lisensi MIT (MIT)

Hak Cipta (c) Microsoft Corporation

Izin dengan ini diberikan, gratis, kepada siapa pun yang memperoleh salinan perangkat lunak ini dan file dokumentasi terkait (“Perangkat Lunak”), untuk menangani Perangkat Lunak tanpa batasan, termasuk tanpa batasan hak untuk menggunakan, menyalin, memodifikasi, menggabungkan, menerbitkan, mendistribusikan, mensublisensikan, dan/atau menjual salinan Perangkat Lunak, dan untuk mengizinkan orang yang kepadanya Perangkat Lunak dilengkapi untuk melakukannya, tunduk pada ketentuan berikut:

Pemberitahuan hak cipta di atas dan pemberitahuan izin ini akan disertakan dalam semua salinan atau sebagian besar Perangkat Lunak.

**PERANGKAT LUNAK DISEDIAKAN “SEBAGAIMANA ADANYA”, TANPA JAMINAN DALAM BENTUK APA PUN, TERSURAT MAUPUN TERSIRAT, TERMASUK TETAPI TIDAK TERBATAS PADA JAMINAN KELAYAKAN UNTUK DIPERJUALBELIKAN, KESESUAIAN UNTUK TUJUAN**



TERTENTU DAN NON-PELANGGARAN. DALAM KEADAAN APA PUN PENULIS ATAU PEMEGANG HAK CIPTA TIDAK AKAN BERTANGGUNG JAWAB ATAS KLAIM, KERUSAKAN, ATAU TANGGUNG JAWAB LAINNYA, BAIK DALAM TINDAKAN KONTRAK, KESALAHAN ATAU LAINNYA, YANG TIMBUL DARI, DI LUAR ATAU SEHUBUNGAN DENGAN PERANGKAT LUNAK ATAU PENGGUNAAN ATAU TRANSAKSI LAIN DALAM PERANGKAT LUNAK.

# AWS CodeBuild Riwayat dokumen Panduan Pengguna

Tabel berikut menjelaskan perubahan penting pada dokumentasi sejak rilis terakhir AWS CodeBuild. Untuk notifikasi tentang pembaruan dokumentasi ini, Anda dapat berlangganan ke umpan RSS.

- Versi API terbaru: 2016-10-06

Perubahan	Deskripsi	Tanggal
<a href="#">Konten baru: Tambahkan acara webhook baru dan jenis filter</a>	Tambahkan dukungan untuk acara webhook baru (RELEASED dan PRERELEASED ) dan jenis filter (TAG_NAME dan RELEASE_NAME ).	Maret 15, 2024
<a href="#">Konten baru: Tambahkan acara webhook baru: PULL_REQUEST_CLOSED</a>	Tambahkan dukungan untuk acara webhook baru: PULL_REQUEST_CLOSED .	Februari 20, 2024
<a href="#">Konten yang diperbarui: Gambar Docker disediakan oleh CodeBuild</a>	Tambahkan dukungan untuk Windows Server Core 2019 (windows-base:2019-3.0 )	Februari 7, 2024
<a href="#">Konten yang diperbarui: Gambar Docker disediakan oleh CodeBuild</a>	Tambahkan dukungan untuk runtime baru untuk Amazon Linux 2023 ( ) a12/aarch64/standard/3.0	Januari 29, 2024
<a href="#">Konten baru: Kapasitas cadangan</a>	CodeBuild sekarang mendukung armada kapasitas cadangan di CodeBuild.	Januari 18, 2024
<a href="#">Jenis komputasi baru</a>	CodeBuild sekarang mendukung jenis komputasi	8 Januari 2024

---

	Linux xLarge. Untuk informasi selengkapnya, lihat <a href="#">Membangun jenis komputasi lingkungan</a> .	
<a href="#">Konten yang diperbarui: Gambar Docker disediakan oleh CodeBuild</a>	Tambahkan dukungan untuk runtime baru untuk Amazon Linux 2 (a12/standard/5.0 ) dan Ubuntu (ubuntu/standard/7.0 )	14 Desember 2023
<a href="#">Konten yang diperbarui: Gambar Docker disediakan oleh CodeBuild</a>	Tambahkan dukungan untuk gambar komputasi Lambda baru	8 Desember 2023
<a href="#">Konten baru: AWS Lambda komputasi</a>	Tambahkan konten baru untuk AWS Lambda komputasi	6 November 2023
<a href="#">Konten baru: Pelari GitHub aksi</a>	Menambahkan konten baru untuk GitHub Action runner	6 Juli 2023
<a href="#">Konten yang diperbarui: Gambar Docker disediakan oleh CodeBuild</a>	Tambahkan dukungan untuk Amazon Linux 2 (a12/standard/5.0 )	17 Mei 2023
<a href="#">Perubahan kebijakan terkelola untuk CodeBuild</a>	Detail tentang pembaruan untuk politisi AWS terkelola untuk sekarang CodeBuild tersedia. Untuk informasi selengkapnya, lihat <a href="#">CodeBuild pembaruan kebijakan AWS terkelola</a> .	16 Mei 2023

<a href="#">Konten yang diperbarui: Gambar Docker disediakan oleh CodeBuild</a>	Hapus dukungan untuk Amazon Linux 2 (a12/standard/3.0 ) dan tambahkan dukungan untuk Amazon Linux 2 (a12/standard/corretto8 ) dan Amazon Linux 2 (a12/standard/corretto11 )	9 Mei 2023
<a href="#">Konten yang diperbarui: Gambar Docker disediakan oleh CodeBuild</a>	Tambahkan dukungan untuk Ubuntu 22.04 ( ubuntu/standard/7.0 )	13 April 2023
<a href="#">Konten yang diperbarui: Gambar Docker disediakan oleh CodeBuild</a>	Hapus dukungan untuk Ubuntu 18.04 ( ubuntu/standard/4.0 ) dan Amazon Linux 2 ( a12/aarch64/standard/1.0 )	31 Maret 2023
<a href="#">Konten yang diperbarui: Gambar Docker disediakan oleh CodeBuild</a>	Hapus dukungan untuk Ubuntu 18.04 ( ubuntu/standard/3.0 ) dan Amazon Linux 2 ( a12/standard/2.0 )	30 Juni 2022
<a href="#">Contoh Amazon ECR: Batasi akses gambar</a>	Saat CodeBuild kredensya l digunakan untuk menarik gambar Amazon ECR, Anda dapat membatasi akses gambar ke proyek tertentu. CodeBuild Untuk informasi selengkapnya, lihat <a href="#">contoh Amazon ECR</a> .	10 Maret 2022

## [Ditambahkan dukungan wilayah](#)

Jenis ARM\_CONTAINER komputasi sekarang didukung di wilayah tambahan berikut: Asia Pasifik (Seoul), Kanada (Tengah), Eropa (London), dan Eropa (Paris). Untuk informasi selengkapnya, lihat [Membangun jenis komputasi lingkungan](#).

10 Maret 2022

## [Batasan VPC baru](#)

Jika Anda mengonfigurasi CodeBuild untuk bekerja dengan VPC, caching lokal tidak didukung. Mulai 02/28/22, pembuatan VPC Anda akan memakan waktu lebih lama karena instans Amazon EC2 baru akan digunakan untuk setiap build.

25 Februari 2022

## [Mode laporan Batch](#)

CodeBuild sekarang memungkinkan Anda untuk memilih bagaimana status build batch dikirim ke penyedia sumber untuk sebuah proyek. Untuk informasi selengkapnya, lihat [Mode laporan Batch](#).

4 Oktober 2021

## [Jenis komputasi baru](#)

CodeBuild sekarang mendukung tipe komputasi ARM kecil. Untuk informasi selengkapnya, lihat [Membangun jenis komputasi lingkungan](#).

13 September 2021

---

<a href="#">Proyek pembangunan publik</a>	CodeBuild sekarang memungkinkan Anda untuk membuat hasil build untuk proyek build Anda tersedia untuk umum tanpa memerlukan akses ke AWS akun. Untuk informasi selengkapnya, lihat <a href="#">Proyek pembangunan publik</a> .	Agustus 11, 2021
<a href="#">Debugging sesi untuk pembuatan batch</a>	CodeBuild sekarang mendukung debugging sesi untuk build batch. <a href="#">Untuk informasi selengkapnya, lihat build-graph dan build-list</a> .	3 Maret 2021
<a href="#">Batas pembuatan bersamaan tingkat proyek</a>	CodeBuild sekarang memungkinkan Anda untuk membatasi jumlah build bersamaan untuk proyek build. Untuk informasi selengkapnya, lihat <a href="#">Konfigurasi proyek</a> dan <a href="#">concurrentBuildLimit</a> .	16 Februari 2021
<a href="#">Properti buildspec baru: s3-prefix</a>	CodeBuild sekarang menyediakan properti <code>s3-prefix</code> buildspec untuk artefak yang memungkinkan Anda menentukan awalan jalur untuk artefak yang diunggah ke Amazon S3. Untuk informasi selengkapnya, lihat awalan <a href="#">s3</a> .	9 Februari 2021

[Properti buildspec baru:  
on-failure](#)

CodeBuild sekarang menyediakan properti `on-failure` buildspec untuk fase build yang memungkinkan Anda menentukan apa yang terjadi ketika fase build gagal. Untuk informasi selengkapnya, lihat [pada kegagalan](#).

9 Februari 2021

[Properti buildspec baru:  
kecualikan jalur](#)

CodeBuild sekarang menyediakan properti `exclude-paths` buildspec untuk artefak yang memungkinkan Anda mengecualikan jalur dari artefak build Anda. Untuk informasi selengkapnya, lihat [jalur pengecualian](#).

9 Februari 2021

[Properti buildspec baru:  
aktifkan symlink](#)

CodeBuild sekarang menyediakan properti `enable-symlinks` buildspec untuk artefak yang memungkinkan Anda mempertahankan tautan simbolis dalam artefak ZIP. Untuk informasi selengkapnya, lihat [mengaktifkan symlink](#).

9 Februari 2021

[Peningkatan nama artefak  
Buildspec](#)

CodeBuild sekarang memungkinkan `artifacts /name` properti untuk berisi informasi jalur. Untuk informasi lebih lanjut, lihat [nama](#).

9 Februari 2021

---

<a href="#">Pelaporan cakupan kode</a>	CodeBuild sekarang menyediakan laporan cakupan kode. Untuk informasi selengkapnya, lihat <a href="#">Laporan cakupan kode</a> .	30 Juli 2020
<a href="#">Batch dibangun</a>	CodeBuild sekarang mendukung menjalankan build proyek secara bersamaan dan terkoordinasi. Untuk informasi selengkapnya, lihat <a href="#">Batch build in CodeBuild</a> .	30 Juli 2020
<a href="#">Gambar Windows Server 2019</a>	CodeBuild sekarang menyediakan image build Windows Server Core 2019. Untuk informasi selengkapnya, lihat <a href="#">gambar Docker yang disediakan oleh CodeBuild</a> .	20 Juli 2020
<a href="#">Manajer Sesi</a>	CodeBuild sekarang memungkinkan Anda untuk menjeda build yang sedang berjalan dan kemudian menggunakan AWS Systems Manager Session Manager untuk terhubung ke container build dan melihat status container. Untuk informasi selengkapnya, lihat <a href="#">Pengelola Sesi</a> .	20 Juli 2020



---

<a href="#">Topik yang diperbarui</a>	CodeBuild sekarang mendukung penentuan shell untuk digunakan di lingkungan build mereka di file buildspec. Untuk informasi selengkapnya, lihat <a href="#">Referensi spesifikasi build</a> .	25 Juni 2020
<a href="#">Pelaporan pengujian dengan kerangka kerja pengujian</a>	Menambahkan beberapa topik yang menjelaskan cara menghasilkan laporan CodeBuild pengujian dengan beberapa kerangka pengujian. Untuk informasi selengkapnya, lihat <a href="#">Pelaporan pengujian dengan kerangka kerja pengujian</a> .	29 Mei 2020
<a href="#">Topik yang diperbarui</a>	CodeBuild sekarang mendukung penambahan tag ke grup laporan. Untuk informasi lebih lanjut, lihat <a href="#">ReportGroup</a> .	21 Mei 2020
<a href="#">Support untuk pelaporan pengujian</a>	CodeBuild dukungan untuk pelaporan pengujian sekarang tersedia secara umum.	21 Mei 2020

---

<a href="#">Topik yang diperbarui</a>	CodeBuild sekarang mendukung pembuatan filter webhook buat untuk Github dan Bitbucket yang memicu build hanya ketika pesan head commit cocok dengan ekspresi yang ditentukan. Untuk informasi lebih lanjut, lihat <a href="#">permintaan GitHub tarik dan sampel filter webhook</a> dan <a href="#">permintaan tarik Bitbucket dan sampel filter webhook</a> .	6 Mei 2020
<a href="#">Topik baru</a>	CodeBuild sekarang mendukung berbagi proyek build dan sumber daya grup laporan. Untuk informasi selengkapnya, lihat <a href="#">Bekerja dengan proyek bersama</a> dan <a href="#">Bekerja dengan grup laporan bersama</a> .	13 Desember 2019
<a href="#">Topik baru dan diperbarui</a>	CodeBuild sekarang mendukung pelaporan pengujian selama menjalankan proyek build. Untuk informasi selengkapnya, lihat <a href="#">Bekerja dengan pelaporan pengujian, Membuat laporan pengujian</a> , dan <a href="#">Membuat laporan pengujian menggunakan AWS CLI sampel</a> .	25 November 2019

---

<a href="#">Topik yang diperbarui</a>	CodeBuild sekarang mendukung Linux GPU dan tipe lingkungan Arm, dan tipe <code>2xlarge</code> komputasi. Untuk informasi selengkapnya, lihat <a href="#">Membangun jenis komputasi lingkungan</a> .	19 November 2019
<a href="#">Topik yang diperbarui</a>	CodeBuild sekarang mendukung nomor build pada semua build, mengekspor variabel lingkungan, dan AWS Secrets Manager integrasi. Untuk informasi selengkapnya, lihat <a href="#">Variabel yang diekspor</a> dan <a href="#">Secrets Manager</a> dalam sintaks <a href="#">Buildspec</a> .	6 November 2019
<a href="#">Topik baru</a>	CodeBuild sekarang mendukung aturan notifikasi. Anda dapat menggunakan aturan notifikasi untuk memberi tahu pengguna tentang perubahan penting dalam proyek build. Untuk informasi selengkapnya, lihat <a href="#">Membuat aturan notifikasi</a> .	5 November 2019
<a href="#">Topik yang diperbarui</a>	CodeBuild sekarang mendukung runtime Android versi 29 dan Go versi 1.13. Untuk informasi selengkapnya, lihat <a href="#">gambar Docker yang disediakan oleh CodeBuild</a> dan sintaks <a href="#">Buildspec</a> .	10 September 2019

---

<a href="#">Topik yang diperbarui</a>	Saat Anda membuat proyek, Anda sekarang dapat memilih image terkelola Amazon Linux 2 (AL2). Untuk informasi selengkapnya, lihat <a href="#">gambar Docker yang disediakan oleh CodeBuild</a> dan <a href="#">versi Runtime dalam contoh file buildspec</a> untuk CodeBuild	16 Agustus 2019
<a href="#">Topik yang diperbarui</a>	Saat Anda membuat proyek, Anda sekarang dapat memilih untuk menonaktifkan enkripsi log S3 dan, jika Anda menggunakan repositori sumber berbasis Git, sertakan submodul Git. Untuk informasi selengkapnya, lihat <a href="#">Membuat proyek build di CodeBuild</a> .	8 Maret 2019
<a href="#">Topik baru</a>	CodeBuild sekarang mendukung caching lokal. Anda dapat menentukan caching lokal dalam satu atau lebih dari empat mode saat membuat build. Untuk informasi selengkapnya, lihat <a href="#">Membangun caching di CodeBuild</a> .	21 Februari 2019

Topik baru

CodeBuild sekarang mendukung grup filter webhook untuk menentukan peristiwa yang memicu build. Untuk informasi selengkapnya, lihat [Memfilter peristiwa GitHub webhook dan Memfilter peristiwa webhook Bitbucket](#).

8 Februari 2019

Topik baru

Panduan CodeBuild Pengguna sekarang menunjukkan cara menggunakan CodeBuild dengan server proxy. Untuk informasi selengkapnya, lihat [Menggunakan CodeBuild dengan server proxy](#).

4 Februari 2019

Topik yang diperbarui

CodeBuild sekarang mendukung penggunaan gambar Amazon ECR yang ada di AWS akun lain. Beberapa topik telah diperbarui untuk mencerminkan perubahan ini, termasuk [contoh Amazon ECR untuk CodeBuild](#), [Buat proyek build](#), dan [Buat peran CodeBuild layanan](#).

24 Januari 2019

---

<a href="#">Support untuk registrasi Docker pribadi</a>	CodeBuild sekarang mendukung penggunaan gambar Docker yang disimpan dalam registri pribadi sebagai lingkungan runtime Anda. Untuk informasi selengkapnya, lihat <a href="#">Registri pribadi dengan AWS Secrets Manager sampel</a> .	24 Januari 2019
<a href="#">Topik yang diperbarui</a>	CodeBuild sekarang mendukung penggunaan token akses untuk terhubung ke GitHub (dengan token akses pribadi) dan repositori Bitbucket (dengan kata sandi aplikasi). Untuk informasi selengkapnya, lihat <a href="#">Membuat proyek build (konsol)</a> dan <a href="#">Menggunakan token akses dengan penyedia sumber Anda</a> .	6 Desember 2018
<a href="#">Topik yang diperbarui</a>	CodeBuild sekarang mendukung metrik build baru yang mengukur durasi setiap fase dalam build. Untuk informasi selengkapnya, lihat <a href="#">CodeBuild CloudWatch metrik</a> .	15 November 2018
<a href="#">Topik kebijakan titik akhir VPC</a>	Titik akhir Amazon VPC untuk CodeBuild saat ini mendukung kebijakan. Untuk informasi selengkapnya, lihat <a href="#">Membuat kebijakan titik akhir VPC</a> untuk CodeBuild	9 November 2018

---

<a href="#">Konten yang diperbarui</a>	Topik telah diperbarui untuk mencerminkan pengalaman konsol baru.	30 Oktober 2018
<a href="#">Sampel Amazon EFS</a>	CodeBuild dapat memasang sistem file Amazon EFS selama pembuatan menggunakan perintah dalam file buildspec proyek. Untuk informasi selengkapnya, lihat <a href="#">contoh Amazon EFS untuk CodeBuild</a> .	26 Oktober 2018
<a href="#">Webhook Bitbucket</a>	CodeBuild sekarang mendukung webhooks ketika Anda menggunakan Bitbucket untuk repositori Anda. Untuk informasi selengkapnya, lihat <a href="#">contoh permintaan tarik Bitbucket untuk CodeBuild</a> .	2 Oktober 2018
<a href="#">Log S3</a>	CodeBuild sekarang mendukung log build di bucket S3. Sebelumnya, Anda hanya bisa membuat log menggunakan CloudWatch Log. Untuk informasi selengkapnya, lihat <a href="#">Membuat proyek</a> .	17 September 2018

[Beberapa sumber input dan beberapa artefak keluaran](#)

CodeBuild sekarang mendukung proyek yang menggunakan lebih dari satu sumber input dan menerbitkan lebih dari satu set artefak. Untuk informasi lebih lanjut, lihat [Beberapa sumber input dan sampel artefak input dan CodePipeline integrasi dengan CodeBuild dan beberapa sumber input dan sampel artefak keluaran](#).

30 Agustus 2018

[Sampel versi semantik](#)

Panduan CodeBuild Pengguna sekarang memiliki contoh berbasis kasus penggunaan yang menunjukkan cara menggunakan versi semantik untuk membuat nama artefak pada waktu pembuatan. Untuk informasi selengkapnya, lihat [Menggunakan versi semantik untuk memberi nama contoh artefak build](#).

14 Agustus 2018



[Sampel situs web statis baru](#)

Panduan CodeBuild Pengguna 14 Agustus 2018

sekarang memiliki contoh berbasis kasus penggunaan yang menunjukkan cara meng-host keluaran build dalam bucket S3. Sampel memanfaatkan dukungan terbaru dari artefak build yang tidak terenkripsi. Untuk informasi selengkapnya, lihat [Membuat situs web statis dengan keluaran build yang dihosting di bucket S3.](#)

[Support untuk mengganti nama artefak dengan versi semantik](#)

Sekarang Anda dapat menggunakan versi semantik untuk menentukan format yang CodeBuild digunakan untuk memberi nama artefak build. Ini berguna karena artefak build dengan nama hard-code menimpa artefak build sebelumnya yang menggunakan nama hard-coded yang sama. Misalnya, jika build dipicu beberapa kali sehari, Anda sekarang dapat menambahkan stempel waktu ke nama artefaknya. Setiap nama artefak build unik dan tidak menimpa artefak bangunan sebelumnya.

7 Agustus 2018

---

<a href="#">Support artefak build yang tidak terenkripsi</a>	CodeBuild sekarang mendukung build dengan artefak build yang tidak terenkripsi. Untuk informasi selengkapnya, lihat <a href="#">Membuat proyek build (konsol)</a> .	26 Juli 2018
<a href="#">Support untuk CloudWatch metrik dan alarm Amazon</a>	CodeBuild sekarang menyediakan integrasi dengan CloudWatch metrik dan alarm. Anda dapat menggunakan CloudWatch konsol CodeBuild atau untuk memantau build di tingkat proyek dan akun. Untuk informasi selengkapnya, lihat <a href="#">Pemantauan build</a> .	19 Juli 2018
<a href="#">Support untuk melaporkan status build</a>	CodeBuild sekarang dapat melaporkan status awal dan penyelesaian build ke penyedia sumber Anda. Untuk informasi selengkapnya, lihat <a href="#">Membuat proyek build di CodeBuild</a> .	10 Juli 2018
<a href="#">Variabel lingkungan ditambahkan ke CodeBuild dokumentasi</a>	Variabel Lingkungan di <a href="#">halaman lingkungan build diperbarui dengan variabel lingkungan</a> CODEBUILD_BUILD_ID, CODEBUILD_LOG_PATH, dan CODEBUILD_START_TIME.	9 Juli 2018

[Support untuk `finally` blok di file `buildspec`](#)

CodeBuild Dokumentasi diperbarui dengan detail tentang `finally` blok opsional dalam file `buildspec`. Perintah di blok akhirnya selalu berjalan setelah perintah di blok perintah yang sesuai. Untuk informasi selengkapnya, lihat sintaks [Buildspec](#).

20 Juni 2018

[CodeBuild pemberitahuan pembaruan agen](#)

CodeBuild Dokumentasi telah diperbarui dengan rincian tentang bagaimana Anda dapat menggunakan Amazon SNS untuk diberitahu ketika versi baru dari CodeBuild agen dirilis. Untuk informasi selengkapnya, lihat [Menerima pemberitahuan untuk versi AWS CodeBuild agen baru](#).

15 Juni 2018

## Pembaruan sebelumnya

Tabel berikut menjelaskan perubahan penting dalam setiap rilis Panduan Pengguna AWS CodeBuild sebelum Juni 2018.

Perubahan	Deskripsi	Tanggal
Support untuk Windows build	CodeBuild sekarang mendukung build untuk platform Microsoft Windows Server, termasuk lingkungan build yang dikemas untuk .NET Core 2.0 di Windows. Untuk informasi	25 Mei 2018

Perubahan	Deskripsi	Tanggal
	selengkapnya, lihat <a href="#">Sampel Microsoft Windows untuk CodeBuild</a> .	
Support untuk membangun idempotensi	Saat menjalankan <code>start-build</code> perintah dengan AWS Command Line Interface (AWS CLI), Anda dapat menentukan bahwa build tersebut idempoten. Untuk informasi selengkapnya, lihat <a href="#">Jalankan build (AWS CLI)</a> .	15 Mei 2018
Support untuk mengesampingkan lebih banyak pengaturan proyek build	Sekarang Anda dapat mengganti lebih banyak setelan proyek build saat membuat build. Penggantian hanya untuk build itu. Untuk informasi selengkapnya, lihat <a href="#">Jalankan build diAWS CodeBuild</a> .	15 Mei 2018
Dukungan VPC Endpoint	Anda sekarang dapat menggunakan titik akhir VPC untuk meningkatkan keamanan build Anda. Untuk informasi selengkapnya, lihat <a href="#">Gunakan titik akhir VPC</a> .	Maret 18, 2018
Support dari pemicu	Anda sekarang dapat membuat pemicu untuk menjadwalkan build pada frekuensi reguler. Untuk informasi selengkapnya, lihat <a href="#">Buat AWS CodeBuild pemicu</a> .	28 Maret 2018

Perubahan	Deskripsi	Tanggal
Dokumentasi titik akhir FIPS	Anda sekarang dapat mempelajari cara menggunakan AWS Command Line Interface (AWS CLI) atau AWS SDK untuk memberitahu CodeBuild untuk menggunakan salah satu dari empat titik akhir Federal Information Processing Standards (FIPS). Untuk informasi selengkapnya, lihat <a href="#">Tentukan AWS CodeBuild titik akhir</a> .	28 Maret 2018
AWS CodeBuild tersedia di Asia Pasifik (Mumbai), Eropa (Paris), dan Amerika Selatan (São Paulo)	AWS CodeBuild sekarang tersedia di wilayah Asia Pasifik (Mumbai), Eropa (Paris), dan Amerika Selatan (São Paulo). Untuk informasi selengkapnya, lihat <a href="#">AWS CodeBuild</a> di Referensi Umum Amazon Web Services.	28 Maret 2018
GitHub Dukungan Server Perusahaan	CodeBuild sekarang dapat membangun dari kode sumber yang disimpan dalam repositori di GitHub Enterprise Server. Untuk informasi selengkapnya, lihat <a href="#">GitHub Contoh Server Perusahaan</a> .	Januari, 25, 2018

Perubahan	Deskripsi	Tanggal
Dukungan kedalaman klon Git	CodeBuild sekarang mendukung pembuatan klon dangkal dengan riwayat terpotong ke jumlah komit yang ditentukan. Untuk informasi selengkapnya, lihat <a href="#">Membuat proyek pembangunan</a> .	Januari, 25, 2018
Dukungan VPC	Build berkemampuan VPC sekarang dapat mengakses sumber daya di dalam VPC Anda. Untuk informasi selengkapnya, lihat <a href="#">Dukungan VPC</a> .	November, 27, 2017
Dukungan caching ketergantungan	CodeBuild sekarang mendukung cache ketergantungan. Ini memungkinkan CodeBuild untuk menyimpan bagian tertentu dari lingkungan build yang dapat digunakan kembali di cache dan menggunakannya di seluruh build.	November, 27, 2017

Perubahan	Deskripsi	Tanggal
Membangun dukungan lencana	CodeBuild sekarang mendukung penggunaan lencana build, yang menyediakan gambar (lencana) yang dapat disematkan dan dihasilkan secara dinamis yang menampilkan status build terbaru untuk sebuah proyek. Untuk informasi selengkapnya, lihat <a href="#">Membangun sampel lencana</a> .	27 November 2017
AWS Config integrasi	AWS Config sekarang mendukung CodeBuild sebagai AWS sumber daya, yang berarti layanan dapat melacak CodeBuild proyek Anda. Untuk informasi lebih lanjut tentang AWS Config, lihat <a href="#">AWS Config sampel</a> .	20 Oktober 2017
Secara otomatis membangun kembali kode sumber yang diperbarui di GitHub repositori	Jika kode sumber Anda disimpan dalam GitHub repositori, Anda dapat mengaktifkan AWS CodeBuild untuk membangun kembali kode sumber Anda setiap kali perubahan kode didorong ke repositori. Untuk informasi selengkapnya, lihat <a href="#">GitHub permintaan tarik dan sampel filter webhook</a> .	21 September 2017

Perubahan	Deskripsi	Tanggal
Cara baru untuk menyimpan dan mengambil variabel lingkungan sensitif atau besar di Amazon EC2 Systems Manager Parameter Store	Anda sekarang dapat menggunakan AWS CodeBuild konsol atau AWS CLI untuk mengambil variabel lingkungan sensitif atau besar yang disimpan di Amazon EC2 Systems Manager Parameter Store. Sekarang Anda juga dapat menggunakan AWS CodeBuild konsol untuk menyimpan jenis variabel lingkungan ini di Amazon EC2 Systems Manager Parameter Store. Sebelumnya, Anda hanya dapat mengambil jenis variabel lingkungan ini dengan memasukkannya ke dalam buildspec atau dengan menjalankan perintah build untuk mengotomatiskan. AWS CLI Anda hanya dapat menyimpan jenis variabel lingkungan ini dengan menggunakan konsol Amazon EC2 Systems Manager Parameter Store. Untuk informasi lebih lanjut, lihat <a href="#">Membuat proyek pembangunan</a> , <a href="#">Mengubah pengaturan proyek pembangunan</a> , dan <a href="#">Jalankan build</a> .	14 September 2017



Perubahan	Deskripsi	Tanggal
Membangun dukungan penghapusan	Anda sekarang dapat menghapus build in AWS CodeBuild. Untuk informasi selengkapnya, lihat <a href="#">Menghapus build</a> .	31 Agustus 2017
Cara terbaru untuk mengambil variabel lingkungan sensitif atau besar yang disimpan di Amazon EC2 Systems Manager Parameter Store dengan menggunakan buildspec	AWS CodeBuild sekarang mempermudah penggunaan buildspec untuk mengambil variabel lingkungan sensitif atau besar yang disimpan di Amazon EC2 Systems Manager Parameter Store. Sebelumnya, Anda hanya dapat mengambil jenis variabel lingkungan ini dengan menjalankan perintah build untuk mengotomatiskan. AWS CLI Untuk informasi selengkapnya, lihat parameter-store pemetaan di <a href="#">Sintaks Buildspec</a> .	10 Agustus 2017
AWS CodeBuild mendukung Bitbucket	CodeBuild sekarang dapat membangun dari kode sumber yang disimpan dalam repositori Bitbucket. Untuk informasi lebih lanjut, lihat <a href="#">Membuat proyek pembangunan</a> dan <a href="#">Jalankan build</a> .	10 Agustus 2017

Perubahan	Deskripsi	Tanggal
AWS CodeBuild tersedia di AS Barat (California N.), Eropa (London), dan Kanada (Tengah)	AWS CodeBuild sekarang tersedia di wilayah AS Barat (California Utara), Eropa (London), dan Kanada (Tengah). Untuk informasi selengkapnya, lihat <a href="#">AWS CodeBuild</a> di Referensi Umum Amazon Web Services.	29 Juni 2017
Nama dan lokasi file buildspec alternatif yang didukung	Sekarang Anda dapat menentukan nama file alternatif atau lokasi file buildspec yang akan digunakan untuk proyek build, bukan file buildspec default yang diberi nama <code>buildspec.yml</code> di root kode sumber. Untuk informasi selengkapnya, lihat <a href="#">Nama file Buildspec dan lokasi penyimpanan</a> .	27 Juni 2017
Contoh pemberitahuan build yang diperbarui	CodeBuild sekarang menyediakan dukungan bawaan untuk pemberitahuan build melalui Amazon CloudWatch Events dan Amazon Simple Notification Service (Amazon SNS). Yang sebelumnya <a href="#">Buat contoh notifikasi</a> telah diperbarui untuk menunjukkan perilaku baru ini.	22 Juni 2017

Perubahan	Deskripsi	Tanggal
Docker dalam sampel gambar khusus ditambahkan	Contoh yang menunjukkan cara menggunakan CodeBuild dan image build Docker khusus untuk membangun dan menjalankan image Docker telah ditambahkan. Untuk informasi lebih lanjut, lihat <a href="#">Docker dalam sampel gambar khusus</a> .	7 Juni 2017
Ambil kode sumber untuk permintaan GitHub tarik	Saat Anda menjalankan build CodeBuild yang bergantung pada kode sumber yang disimpan dalam GitHub repositori, Anda sekarang dapat menentukan ID permintaan GitHub tarik untuk dibuat. Anda juga dapat menentukan ID komit, nama cabang, atau nama tag sebagai gantinya. Untuk informasi selengkapnya, lihat nilai versi Sumber dalam <a href="#">Menjalankan build (konsol)</a> atau <code>sourceVersion</code> nilai dalam <a href="#">Jalankan build (AWS CLI)</a> .	6 Juni 2017

Perubahan	Deskripsi	Tanggal
Membangun versi spesifikasi diperbarui	Versi baru dari format buildspec telah dirilis. Versi 0.2 mengatasi masalah CodeBuild menjalankan setiap perintah build dalam instance terpisah dari shell default. Juga di versi 0.2, <code>environment_variables</code> diganti namanya menjadi <code>env</code> , dan <code>plaintext</code> diganti namanya menjadi <code>variables</code> . Untuk informasi selengkapnya, lihat <a href="#">Buat referensi spesifikasi untuk CodeBuild</a> .	9 Mei 2017
Dockerfiles untuk membangun gambar tersedia di GitHub	Definisi untuk banyak gambar build yang disediakan oleh AWS CodeBuild tersedia sebagai Dockerfiles di GitHub. Untuk informasi selengkapnya, lihat kolom Definisi tabel di <a href="#">Gambar Docker disediakan oleh CodeBuild</a> .	2 Mei 2017
AWS CodeBuild tersedia di Eropa (Frankfurt), Asia Pasifik (Singapura), Asia Pasifik (Sydney), dan Asia Pasifik (Tokyo)	AWS CodeBuild sekarang tersedia di wilayah Eropa (Frankfurt), Asia Pasifik (Singapura), Asia Pasifik (Sydney), dan Asia Pasifik (Tokyo). Untuk informasi selengkapnya, lihat <a href="#">AWS CodeBuild</a> di Referensi Umum Amazon Web Services.	21 Maret 2017

Perubahan	Deskripsi	Tanggal
CodePipeline dukungan tindakan uji untuk CodeBuild	Anda sekarang dapat menambahkan ke pipeline CodePipeline dalam tindakan pengujian yang menggunakan CodeBuild. Untuk informasi selengkapnya, lihat <a href="#">Menambahkan tindakan CodeBuild pengujian ke pipeline (CodePipelinekonsol)</a> .	8 Maret 2017
File buildspec mendukung pengambilan output build dari dalam direktori tingkat atas yang dipilih	File buildspec sekarang memungkinkan Anda menentukan direktori tingkat atas individual yang isinya dapat Anda instruksikan untuk disertakan dalam artefak keluaran CodeBuild build. Anda melakukan ini dengan menggunakan base-directory pemetaan. Untuk informasi selengkapnya, lihat <a href="#">Sintaks Buildspec</a> .	8 Februari 2017
Variabel lingkungan bawaan	AWS CodeBuild menyediakan variabel lingkungan bawaan tambahan untuk build Anda untuk digunakan. Ini termasuk variabel lingkungan yang menjelaskan entitas yang memulai build, URL ke repositori kode sumber, ID versi kode sumber, dan banyak lagi. Untuk informasi selengkapnya, lihat <a href="#">Variabel lingkungan di lingkungan build</a> .	30 Januari 2017

Perubahan	Deskripsi	Tanggal
AWS CodeBuild tersedia di AS Timur (Ohio)	AWS CodeBuild sekarang tersedia di wilayah AS Timur (Ohio). Untuk informasi selengkapnya, lihat <a href="#">AWS CodeBuild</a> di Referensi Umum Amazon Web Services.	Januari 19, 2017
Informasi perilaku shell dan perintah	CodeBuild menjalankan setiap perintah yang Anda tentukan dalam instance terpisah dari shell default lingkungan build. Perilaku default ini dapat menghasilkan beberapa efek samping yang tidak terduga untuk perintah Anda. Kami merekomendasikan beberapa pendekatan untuk mengatasi perilaku default ini jika diperlukan. Untuk informasi selengkapnya, lihat <a href="#">Kerang dan perintah dalam membangun lingkungan</a> .	9 Desember 2016
Informasi variabel lingkungan	CodeBuild menyediakan beberapa variabel lingkungan yang dapat Anda gunakan dalam perintah build Anda. Anda juga dapat menentukan variabel lingkungan Anda sendiri. Untuk informasi selengkapnya, lihat <a href="#">Variabel lingkungan di lingkungan build</a> .	Desember 7, 2016

Perubahan	Deskripsi	Tanggal
Topik pemecahan masalah	Informasi pemecahan masalah sekarang tersedia. Untuk informasi selengkapnya, lihat <a href="#">Pemecahan masalah AWS CodeBuild</a> .	5 Desember 2016
Rilis awal plugin Jenkins	Ini adalah rilis awal plugin CodeBuild Jenkins. Untuk informasi selengkapnya, lihat <a href="#">GunakanAWS CodeBuild dengan Jenkins</a> .	5 Desember 2016
Panduan Pengguna rilis awal	Ini adalah rilis awal Panduan Pengguna CodeBuild .	1 Desember 2016

# AWSGlosarium

Untuk AWS terminologi terbaru, lihat [AWSglosarium di Referensi](#). **Glosarium AWS**



Terjemahan disediakan oleh mesin penerjemah. Jika konten terjemahan yang diberikan bertentangan dengan versi bahasa Inggris aslinya, utamakan versi bahasa Inggris.