aws

Panduan Pengguna

AWS Transformasi

AWS Transformasi: Panduan Pengguna

Table of Contents

Apa itu AWS Transformasi?

AWS Transform adalah layanan yang membantu Anda mempercepat dan menyederhanakan transformasi infrastruktur, aplikasi, dan kode dengan pengalaman AI agen. Ini menyediakan alat khusus untuk merampingkan upaya modernisasi dan migrasi di berbagai beban kerja.

Dengan AWS Transform, Anda dapat:

- Modernisasi dan migrasi IBM z/OS dan Fujitsu GS21 ke AWS
- Migrasikan beban kerja VMware ke Amazon EC2
- Modernisasi aplikasi.NET ke Linux-ready lintas platform.NET
- Menilai beban kerja untuk kesiapan migrasi

AWS Transform membantu Anda menurunkan tugas transformasi padat karya dan kompleks di seluruh fase penemuan, perencanaan, dan eksekusi ke agen AI dengan keahlian dalam bahasa, kerangka kerja, dan infrastruktur. Pendekatan ini memungkinkan tim Anda untuk fokus pada inovasi sementara itu secara efisien mengubah proyek skala besar yang kompleks melalui pengalaman web terpadu.

Tidak ada biaya tambahan untuk menggunakan AWS Transform.

Terminologi

Dalam bagian ini, huruf miring menunjukkan istilah resmi dalam definisi istilah yang berbeda.

Koneksi akun

: Akun dalam konteks ini adalah referensi umum ke wadah milik pelanggan atau batas keamanan untuk sumber daya dalam atau layanan jarak jauh, misalnya, akun AWS atau. Akun AWS GitHub

Artifak

: Output deliverable yang dihasilkan oleh AWS Transform.

Administrator

: Administrator dapat membaca dan mengubah segala sesuatu di ruang kerja. Mereka dapat memulai obrolan dengan AWS Transform, memulai dan menghentikan pekerjaan, dan upload/

download artefak. Administrator dapat berinteraksi dengan menjalankan pekerjaan untuk tindakan human-in-the-loop (HITL), dan dapat menyetujui tindakan HITL kritis seperti penggabungan ke main, melakukan dekomposisi grafik, atau menerapkan kode ke lingkungan produksi. Administrator dapat mengubah ruang kerja, konektor, dan pengguna.

Agen

Layanan khusus tugas yang mengeksekusi jenis transformasi tertentu. Misalnya, migrasi VMware.

Penyetuju

Pemberi persetujuan memiliki izin yang mirip dengan Administrator kecuali bahwa mereka tidak memiliki izin untuk mengubah ruang kerja, konektor, atau pengguna. Pemberi persetujuan tidak dapat mengubah ruang kerja, konektor, atau pengguna.

Aset

Masukan untuk pekerjaan transformasi. Misalnya, kode sumber pelanggan, server, database, jaringan. Aset diakses melalui konektor.

Permintaan kolaborator

Sebuah tugas di mana AWS Transform meminta manusia untuk melakukan sesuatu.

Konektor

Konektor adalah penyedia aset yang memungkinkan akses ke sumber daya milik pelanggan dalam sistem eksternal untuk AWS Transform.

Saat Anda mengatur konektor, administrator akun yang Anda sambungkan harus menerima koneksi. Untuk menerima koneksi, mereka harus memiliki izin yang diberikan dalam kebijakan akseptor konektor.

Dua akun berikut harus identik, atau dalam AWS Organizations organisasi yang sama:

- Akun dari mana administrator AWS Transform mengaktifkan layanan.
- Akun yang akan berada di ujung penerima transformasi Anda. Akun ini harus diberi peran IAM yang memungkinkannya menggunakan konektor.

Kontributor

Kontributor dapat membaca semua yang ada di ruang kerja. Mereka dapat memulai obrolan dengan AWS Transform, memulai dan menghentikan pekerjaan, mengunggah atau mengunduh artefak, dan berinteraksi dengan menjalankan pekerjaan untuk tindakan HITL. Namun, mereka tidak dapat melakukan tindakan HITL kritis seperti menggabungkan ke main, melakukan

dekomposisi grafik, atau menerapkan kode ke lingkungan produksi. Kontributor juga tidak dapat mengubah ruang kerja, konektor, atau pengguna.

Tujuan

Status akhir yang ditentukan pengguna yang ingin dicapai AWS Transform. Pengguna menulis ini, dan kemudian AWS Transform mengubah tujuan menjadi serangkaian tugas yang dilakukan bersama dengan pengguna bila diperlukan.

Job

Proses yang berjalan lama (weeks/months+) yang sedang dikerjakan AWS Transform untuk memenuhi tujuan yang ditentukan oleh pengguna. Terdiri dari beberapa tugas dan permintaan kolaborator.

Rencana

Daftar tugas yang dilakukan AWS Transform (dengan bantuan dari pengguna manusia) dalam mengejar tujuan.

Pembaca

Pembaca dapat melihat status dan hasil pekerjaan AWS Transform, tetapi tidak dapat membuat perubahan apa pun. Mereka dapat membaca semua yang ada di ruang kerja, mengunduh artefak, melihat pekerjaan, dan melihat tindakan human-in-the-loop (HITL). Namun, pembaca tidak dapat melakukan tindakan mutasi di AWS Transform.

Tugas

Unit kerja individu yang merupakan bagian dari pekerjaan.

Worklog

Log tindakan apa yang dilakukan AWS Transform dan pengguna sebagai bagian dari pekerjaan.

Ruang kerja

Sumber daya AWS Transform yang berisi sumber daya lain seperti konektor dan pekerjaan. Ruang kerja berfungsi sebagai batas izin.

Memulai dengan AWS Transformasi

Topik

Penyiapan AWS Transformasi

Sebelum Anda mulai

Sebelum Anda mengatur AWS Transform, pastikan Anda memiliki AWS akun dengan akses administrator

> **Note**
>
> Jika Anda ingin mencoba AWS Transform sebagai bukti konsep atau untuk lingkungan pengujian, lihat Mulai cepat: Mencoba Transformasi. AWS

Mendaftar untuk Akun AWS

Untuk memulai AWS, Anda membutuhkan Akun AWS. Untuk informasi tentang membuat Akun AWS, lihat Memulai dengan Akun AWS di Panduan AWS Account Management Referensi.

Memulai dengan IAM-only akses

Dengan IAM-only akses, pengguna mengautentikasi dengan AWS kredensialnya yang ada.

Untuk mengatur IAM-only akses:

1. Di AWS konsol, navigasikan ke AWS Transform dan pilih Memulai.
2. Lengkapi konfigurasi pengaturan akun.

3. Di bawah Akses pengguna, pilih IAM-only akses.
4. Pilih Aktifkan untuk membuat profil Anda.

Pengguna masuk ke aplikasi web AWS Transform melalui AWS Sign-In. Pengenal pengguna mereka dalam aplikasi web didasarkan pada prinsipal IAM mereka.

> **Important**
>
> Anda tidak dapat menambahkan penyedia identitas setelah penyiapan selesai. Untuk mengelola pengguna dengan IAM Identity Center atau penyedia identitas yang ada, pilih salah satu opsi tersebut selama penyiapan awal.

Dengan IAM-only akses, setiap kepala sekolah IAM dengan `transform:AccessTransformProfile` izin pada sumber daya profil dapat mengakses AWS Transform. Anda mengelola akses pengguna melalui kebijakan IAM, bukan melalui penetapan pengguna penyedia identitas.

`transform:AccessTransformProfile`Tindakan hanya memberikan akses ke AWS Transform. Begitu berada di dalam AWS Transform, peran ruang kerja mengontrol tindakan apa yang dapat dilakukan pengguna. Peran ini termasuk Admin, Kontributor, Penyetuju, dan. Read-only Peran ruang kerja berlaku terlepas dari bagaimana pengguna diautentikasi.

Menggunakan kredensil IAM

Penanganan informasi pengguna

Ketika kepala sekolah IAM mengakses AWS Transform, layanan menangani identitas pengguna secara berbeda dibandingkan dengan akses berbasis penyedia identitas.

Identitas pengguna

Pengguna IAM diidentifikasi oleh kepala IAM mereka.

Identitas ganda

Pengguna yang memiliki identitas penyedia identitas (seperti IAM Identity Center) dan sesi IAM muncul sebagai dua pengguna terpisah di AWS Transform. Pekerjaan mereka tidak terkait antara dua identitas.

Pemberitahuan email

Pemberitahuan email hanya diaktifkan untuk pengguna SSO. Prinsipal IAM masih dapat ditambahkan ke ruang kerja tetapi mereka tidak akan menerima pemberitahuan email.

Audit

Panggilan API IAM masuk AWS CloudTrail dengan identitas IAM pemanggil. Tindakan yang dilakukan oleh pengguna IAM dalam suatu pekerjaan juga dapat dilacak di worklog AWS Transform.

Mengaktifkan akses IAM bersama penyedia identitas

Jika Anda mengatur AWS Transform dengan IAM Identity Center atau penyedia identitas pihak ketiga, Anda juga dapat mengaktifkan akses IAM sebagai cara tambahan untuk mengautentikasi. Hal ini memungkinkan prinsipal IAM untuk mengakses AWS Transform bersama pengguna penyedia identitas Anda yang ada.

Akses IAM diaktifkan secara default saat Anda membuat profil dengan Pusat Identitas IAM atau penyedia identitas pihak ketiga. Untuk mengaktifkan atau menonaktifkan akses IAM kapan saja:

1. Di konsol AWS Transform, navigasikan ke Pengaturan.
2. Di bawah Access AWS Transform dengan kredensyal IAM, aktifkan atau nonaktifkan Aktifkan akses IAM.

Ketika akses IAM diaktifkan, pengguna penyedia identitas dan prinsipal IAM dapat menggunakan Transform. AWS Pengguna IAM dan pengguna penyedia identitas dapat berkolaborasi di ruang kerja yang sama.

Memulai dengan AWS Organizations

Ikuti langkah-langkah ini untuk mengatur AWS Transform:

1. Masuk ke akun manajemen AWS Organizations Anda.
2. Arahkan ke layanan AWS Transform.
3. Pilih Aktifkan layanan bagi organisasi Anda untuk menggunakan AWS Transform.
4. Konfigurasikan izin yang diperlukan untuk akun anggota organisasi.

5. Akses pengalaman web AWS Transform dari akun anggota Anda.

> **Note**
>
> Untuk menggunakan AWS solusi Landing Zone Accelerator (LZA) untuk membangun landing zone Anda bersama dengan kemampuan AWS Transform for migration, akun AWS Transform dan instalasi LZA Anda harus berada di Organisasi yang sama. AWS Menggunakan ID Organizations terpisah untuk penerapan LZA dan AWS Transform tidak didukung karena hal ini dapat menyebabkan ketidakkonsistenan dalam manajemen organisasi dan penerapan sumber daya. Untuk mempelajari cara menyiapkan instalasi LZA Anda menggunakan Organizations, lihat Menerapkan fondasi cloud untuk mendukung beban kerja yang sangat diatur dan persyaratan kepatuhan yang kompleks dalam panduan pengguna Landing Zone Accelerator on Implementation Guide. AWS

Memulai dengan AWS IAM Identity Center

Ikuti langkah-langkah ini untuk menggunakan IAM Identity Center for AWS Transform dan untuk menambahkan pengguna dan grup.

Secara default, tidak ada pengguna yang memiliki akses ke AWS Transform saat Anda pertama kali mengaktifkannya.

> **Note**
>
> Pusat Identitas IAM tidak terbatas pada wilayah di mana ia didirikan. Jika Anda sudah menyiapkan Pusat Identitas IAM di wilayah yang tidak didukung oleh AWS Transform, Anda dapat menggunakannya untuk AWS Transform.

1. Siapkan Pusat Identitas IAM mengikuti petunjuk di Untuk mengaktifkan instance Pusat Identitas IAM.

   Konfigurasikan Pusat Identitas IAM untuk menggunakan penyedia identitas perusahaan eksternal, dan mereplikasi info pengguna dan grupnya ke Pusat Identitas IAM.
2. Di AWS konsol, pilih AWS Transform dan pilih Memulai.
3. Pilih Aktifkan layanan bagi organisasi Anda untuk menggunakan AWS Transform.

4. Pilih kunci enkripsi. Pilihan default adalah kunci AWS terkelola. Untuk menggunakan kunci kustom:

   a. Di bawah Kunci enkripsi, pilih Sesuaikan pengaturan enkripsi.

   b. Pilih Gunakan tombol AWS KMS.

   c. Pilih kunci yang ada atau buat yang baru.

   d. Pilih Kirim untuk menerapkan perubahan Anda, lalu pilih Aktifkan AWS Transformasi.

   Pilih Lihat profil untuk melihat konfigurasi. URL aplikasi Web digunakan oleh pengguna Anda untuk mengakses pengalaman web terpadu AWS Transform.

5. Pilih Pengguna di panel navigasi dan pilih Tetapkan pengguna atau grup.
6. Cari nama pengguna atau grup yang ingin Anda otorisasi untuk menggunakan AWS Transform. Pencarian referensi pengguna dan grup yang disebarkan dari penyedia identitas Anda.
7. Pilih grup atau pengguna, pilih Selesai, lalu, Tetapkan. Pengguna ini berwenang untuk menggunakan antarmuka web terpadu AWS Transform.

Menggunakan penyedia identitas pihak ketiga

AWS Transform mendukung integrasi dengan penyedia identitas pihak ketiga (IdPs) seperti Azure Active Directory (Entra ID) dan Okta Workforce Identity. Ini memungkinkan Anda untuk menggunakan sistem manajemen identitas yang ada untuk otentikasi pengguna.

Prasyarat

Sebelum mengonfigurasi integrasi penyedia identitas pihak ketiga, pastikan bahwa pengguna di penyedia identitas Anda memiliki atribut nama, email, dan nama pengguna yang dikonfigurasi

Informasi Tersimpan

Saat Anda menggunakan AWS Transform with IdPs, AWS menyimpan informasi pengguna minimal yang dienkripsi dan diamankan:

Informasi Pengguna yang Tersimpan

AWS Transform menyimpan informasi profil pengguna dasar saat login pertama, termasuk nama tampilan, alamat email, nama pengguna (preferred_username), dan pengenal pengguna yang

unik. Informasi ini dienkripsi menggunakan kunci KMS milik pelanggan atau kunci milik layanan, tergantung pada konfigurasi profil Transform pelanggan. AWS Data disimpan dalam database otentikasi AWS Transform dan hanya dikumpulkan selama sesi login awal. Ini mengisi hasil pencarian saat mengundang pengguna lain ke ruang kerja.

Siklus Hidup Data

Informasi pengguna disimpan hanya untuk pengguna yang telah masuk ke aplikasi web AWS Transform setidaknya sekali, dan mungkin menjadi basi jika pengguna memperbarui informasi mereka di penyedia identitas mereka tanpa masuk kembali ke AWS Transform. Semua informasi pengguna yang tersimpan akan dihapus saat profil AWS Transform dihapus.

Penyimpanan Rahasia Klien

Rahasia klien yang disediakan selama penyiapan disimpan menggunakan AWS Secrets Manager melalui Service Linked Secret (SLS) di akun Anda.

Penanganan Pengenal Pengguna

Entra

Menggunakan klaim “oid” (pengenal objek) sebagai pengenal pengguna unik, yang tidak dapat diubah dan secara unik mengidentifikasi pengguna di seluruh penyewa Microsoft. Nilai ini terlihat oleh pelanggan di konsol Entra dan muncul di CloudTrail log.

Identitas Tenaga Kerja Okta

Menggunakan klaim yang berbeda untuk identifikasi pengguna tergantung pada jenis token - klaim “sub” dalam token ID dan klaim “uid” di token Access. AWS Transform memvalidasi bahwa kedua klaim mengandung nilai yang sama selama otentikasi. Nilai ini terlihat oleh pelanggan di konsol Okta dan muncul di CloudTrail log.

Menyiapkan Azure Active Directory (Entra ID)

Untuk mengonfigurasi integrasi Azure Active Directory dengan AWS Transform:

1. Arahkan ke portal Azure dan pilih Azure Active Directory.
2. Di panel navigasi kiri, pilih Kelola > Pendaftaran aplikasi.
3. Pilih + Pendaftaran baru.

4. Masukkan nama aplikasi, pilih jenis akun yang didukung, biarkan URI pengalihan kosong, dan pilih Daftar.
5. Di navigasi kiri, pilih Kelola > Manifest.
6. Perbarui `requestedAccessTokenVersion` dari `null` ke 2 dan pilih Simpan.
7. Pilih Kelola > Paparkan API dan pilih Tambahkan cakupan.
8. Buat URI ID Aplikasi menggunakan struktur default`api://<client-id>`.
9. Tambahkan ruang lingkup`transform:read_write`.
10. Pilih Tambahkan sertifikat atau rahasia dan buat rahasia klien baru. Simpan nilai ini sesuai kebutuhan untuk pembuatan profil.
11. Temukan URL Penerbit dengan memilih Endpoint dan memilih dokumen metadata OpenID Connect. Bidang “penerbit” dalam metadata adalah URL Penerbit Anda.
12. Buat profil di konsol AWS Transform menggunakan Client ID, Client Secret, dan Issuer URL.
13. Setelah pembuatan profil, tambahkan URI pengalihan dengan memilih Tambahkan platform, pilih Web, dan masukkan`<web-application-url>/login/callback`.

Menyiapkan Identitas Tenaga Kerja Okta

Untuk mengonfigurasi integrasi Identitas Tenaga Kerja Okta dengan AWS Transform:

1. Arahkan ke konsol Okta Workforce Identity Anda.
2. Pilih Applications > Applications dan pilih Create App Integration.
3. Pilih OIDC - OpenID Connect dan Web Application, lalu pilih Next.
4. Beri nama aplikasi Anda, biarkan Jenis Hibah sebagai Kode Otorisasi, biarkan URI pengalihan kosong, konfigurasikan tugas pengguna, dan pilih Simpan.
5. Arahkan ke tab Masuk dan atur Penerbit ke URL Okta, bukan Dinamis.
6. Salin ID Klien dan konfigurasikan sebagai Audiens untuk Server Otorisasi Anda dengan membuka Security > API dan menambahkan Server Otorisasi.
7. Di Server Otorisasi, tambahkan ruang lingkup di `transform:read_write` bawah tab Scopes.
8. Tambahkan Kebijakan Akses yang memungkinkan Aplikasi OIDC menggunakan Server Otorisasi ini dan mengonfigurasi aturan untuk kebijakan tersebut.
9. Pada halaman Pengaturan Server Otorisasi, perhatikan URL Penerbit untuk pembuatan profil di AWS Transform.

10. Buat profil di AWS Transform menggunakan URL Penerbit, ID Klien, dan Rahasia Klien dari pengaturan aplikasi.
11. Setelah pembuatan profil, tambahkan `<web-application-url>/login/callback` sebagai URL pengalihan di tab Umum aplikasi.

> **Note**
>
> Jika Anda ingin diarahkan kembali ke aplikasi web AWS Transform setelah logout, Anda harus mengonfigurasi URL aplikasi web Anda sebagai asal tepercaya di bawah Security > API.

Orientasi pengguna

Bagian ini menjelaskan pengalaman bagi pengguna yang telah diberikan akses ke AWS Transform.

Menerima undangan

Ketika pengguna ditambahkan ke AWS Transform, mereka menerima undangan email yang berisi:

- Salam dan informasi tentang undangan
- URL aplikasi web AWS Transform
- Nama pengguna mereka
- Tautan untuk menerima undangan dan mengatur kata sandi mereka

Untuk mengatur akun mereka:

1. Pengguna mengklik tautan “Terima undangan” di email.
2. Pada halaman “Daftar pengguna baru”, mereka memasukkan dan mengonfirmasi kata sandi.
3. Kata sandi harus memenuhi persyaratan keamanan, termasuk:
    - Setidaknya 8 karakter
    - Setidaknya satu huruf besar
    - Setidaknya satu huruf kecil
    - Setidaknya satu nomor
    - Setidaknya satu karakter khusus

4. Setelah membuat kata sandi, mereka melihat konfirmasi bahwa akun mereka berhasil dibuat.

Masuk ke AWS Transformasi

Untuk masuk ke AWS Transform:

1. Arahkan ke URL aplikasi web AWS Transform yang disediakan di email undangan.
2. Masukkan nama pengguna.
3. Pilih Berikutnya.
4. Memasukkan kata sandi.
5. Pilih Masuk.

Selamat datang pengalaman

Setelah login pertama, pengguna melihat halaman selamat datang AWS Transform dengan:

- Salam yang dipersonalisasi
- Kemampuan transformasi yang tersedia
- Opsi untuk membuat ruang kerja

Halaman selamat datang menyediakan informasi tentang kemampuan transformasi yang tersedia di AWS Transform, termasuk:

- Modernisasi migrasi IBM z/OS ke AWS
- Migrasikan beban kerja VMware ke Amazon EC2
- Modernisasi aplikasi.NET ke Linux-ready lintas platform.NET
- Menilai beban kerja untuk kesiapan migrasi

Pengguna dapat memulai dengan membuat ruang kerja atau meminta tim mereka untuk menambahkannya ke ruang kerja yang ada.

Menggunakan bucket Amazon S3 Anda sendiri

Secara default, AWS Transform menggunakan bucket Amazon S3 yang dikelola layanan untuk menyimpan artefak transformasi. Anda dapat memilih untuk menggunakan bucket Amazon S3

Anda sendiri sebagai gantinya untuk kontrol yang lebih besar atas penyimpanan data, enkripsi, dan kebijakan akses.

> **Note**
>
> Jika Anda memiliki bucket Amazon S3 untuk transformasi mainframe, Anda dapat terus menggunakan konektor S3 itu dan Anda tidak perlu menggunakan fungsi ini.

> **Note**
>
> Bucket Amazon S3 Anda hanya menyimpan artefak unggahan dan transformasi yang berinteraksi dengan Anda. Artefak yang dihasilkan sistem internal tidak disimpan di bucket Anda. Selain itu, AWS Transform mengindeks artefak di bucket Anda. Ini menyimpan data yang diindeks dalam basis pengetahuan yang dikelola layanan yang digunakan untuk memberi Anda pengalaman obrolan yang diperkaya.

Prasyarat

Sebelum mengonfigurasi bucket Amazon S3 Anda sendiri, pastikan persyaratan berikut terpenuhi:

- Bucket harus berada di AWS Region yang sama di mana AWS Transform diaktifkan.
- Kebijakan bucket yang diperlukan harus diterapkan pada bucket Anda. Untuk informasi selengkapnya, lihat the section called “Kebijakan bucket yang diperlukan”.
- Jika Anda menggunakan aplikasi web AWS Transform, CORS harus dikonfigurasi pada bucket Anda. Untuk informasi selengkapnya, lihat the section called “Konfigurasi CORS yang diperlukan”.
- Jika Anda menggunakan kunci AWS KMS kustom, kebijakan kunci yang diperlukan harus diterapkan. Untuk informasi selengkapnya, lihat the section called “Kebijakan kunci KMS (opsional)”.

Konfigurasi enkripsi yang didukung

Bucket Anda harus menggunakan salah satu konfigurasi enkripsi berikut:

- SSE-S3 (AES256) - Biarkan bidang kunci KMS kosong. AWS Transform menulis objek tanpa menentukan header enkripsi, dan Amazon S3 menerapkan enkripsi S3-managed default bucket

dengan kunci. Untuk informasi selengkapnya, lihat Menggunakan enkripsi sisi server dengan kunci terkelola Amazon S3 () SSE-S3 di Panduan Pengguna Amazon S3.

- SSE-KMS dengan kunci KMS yang dikelola pelanggan - Berikan ARN kunci KMS. AWS Transform menulis objek dengan `x-amz-server-side-encryption: aws:kms` menggunakan kunci yang disediakan. Ikutithe section called “Kebijakan kunci KMS (opsional)”.

Berikut ini tidak didukung:

- SSE-KMS dengan kunci KMS yang AWS dikelola (misalnya,) `aws/s3`
- SSE-C
- DSSE-KMS
- Tombol KMS asimetris atau tombol KMS dengan spesifikasi kunci selain SYMMETRIC_DEFAULT

> **Warning**
>
> AWS-kunci KMS terkelola tidak didukung. Jika bucket Anda dienkripsi`aws/s3`, migrasi ke kunci yang dikelola pelanggan atau SSE-S3 sebelum menggunakan bucket dengan Transform. AWS

Mengonfigurasi bucket Amazon S3 Anda

Anda dapat mengonfigurasi AWS Transform untuk menggunakan bucket Amazon S3 Anda sendiri dari konsol AWS Transform.

Untuk menggunakan bucket Amazon S3 Anda sendiri

1. Di konsol AWS Transform, pilih Pengaturan.
2. Di bawah Penyimpanan Artifact, pilih Gunakan bucket S3 saya sendiri. Anda dapat menggunakan ember di akun Anda saat ini Akun AWS atau di akun lain.
3. Untuk Bucket, masukkan URI Amazon S3.
4. (Opsional) Untuk kunci KMS, masukkan ARN kunci AWS KMS untuk mengenkripsi objek di bucket. Jika Anda membiarkan bidang ini kosong, AWS Transform akan menulis objek tanpa header enkripsi, dan Amazon S3 menerapkan setelan enkripsi default bucket. Untuk informasi selengkapnya, lihat Menyetel perilaku enkripsi sisi server default untuk bucket Amazon S3 di Panduan Pengguna Amazon S3.

Jika enkripsi default bucket Anda SSE-KMS, masukkan ARN kunci KMS yang dikelola pelanggan di bidang ini dan perbarui kebijakan kunci Anda. Untuk informasi selengkapnya, lihat the section called “Kebijakan kunci KMS (opsional)”.

5. Pilih Simpan. AWS Transform memvalidasi konfigurasi bucket dan izin sebelum menerapkan konfigurasi.

Setelah Anda menyimpan konfigurasi, AWS Transform menggunakan bucket Anda untuk menyimpan artefak transformasi.

> **Note**
>
> AWS Transform memvalidasi konfigurasi Anda hanya ketika Anda menyimpan. Jika nanti Anda mengubah kebijakan bucket, konfigurasi CORS, atau kebijakan kunci KMS, simpan kembali setelan profil Anda untuk memvalidasi ulang.

Kebijakan bucket yang diperlukan

Anda harus mengonfigurasi kebijakan bucket untuk memberikan akses AWS Transform. Tambahkan kebijakan bucket berikut ke bucket Amazon S3 agar prinsipal layanan AWS Transform dapat membaca, menulis, menghapus, dan mencantumkan artefak transformasi.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": [
                    "transform.amazonaws.com"
                ]
            },
            "Action": [
                "s3:GetObject",
                "s3:PutObject",
                "s3:DeleteObject",
                "s3:AbortMultipartUpload",
                "s3:ListMultipartUploadParts",
                "s3:PutObjectTagging"
```

```
            ],
            "Resource": "arn:aws:s3:::bucket-name/AWSTransform/*",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "profile-account-id",
                    "aws:SourceArn": "profile-arn"
                }
            }
        },
        {
            "Effect": "Allow",
            "Principal": {
                "Service": [
                    "transform.amazonaws.com"
                ]
            },
            "Action": "s3:ListBucket",
            "Resource": "arn:aws:s3:::bucket-name",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "profile-account-id",
                    "aws:SourceArn": "profile-arn"
                }
            }
        }
    ]
}
```

Ganti nilai-nilai berikut:

- *bucket-name*— Nama ember Amazon S3 Anda.
- *profile-account-id*— Akun AWS ID yang terkait dengan profil AWS Transform Anda.
- *profile-arn*— ARN profil AWS Transform Anda.

Konfigurasi CORS yang diperlukan

Jika Anda menggunakan aplikasi web AWS Transform, Anda harus mengonfigurasi Berbagi Cross-Origin Sumber Daya (CORS) di bucket Amazon S3 Anda. Anda dapat menemukan domain aplikasi web setelah mengaktifkan AWS Transform. Untuk informasi selengkapnya, lihat Memulai.

```
[
```

```
    {
        "AllowedHeaders": [
            "host",
            "content-type",
            "if-none-match",
            "x-amz-checksum-sha256",
            "x-amz-expected-bucket-owner",
            "x-amz-server-side-encryption",
            "x-amz-server-side-encryption-aws-kms-key-id",
            "x-amz-server-side-encryption-context",
            "x-amz-source-account",
            "x-amz-source-arn"
        ],
        "AllowedMethods": [
            "GET",
            "PUT",
            "HEAD"
        ],
        "AllowedOrigins": [
            "webapp-domain"
        ],
        "ExposeHeaders": [
            "ETag",
            "x-amz-checksum-sha256",
            "x-amz-request-id",
            "x-amz-id-2"
        ],
        "MaxAgeSeconds": 3600
    }
]
```

Ganti nilai-nilai berikut:

- *webapp-domain*— URL asal aplikasi web Anda (misalnya,`https://1a2b3c4d5e6f7a8b9.transform.us-east-1.on.aws`), yang dapat ditemukan di halaman pengaturan AWS Transform. Jangan sertakan garis miring.

Kebijakan kunci KMS (opsional)

Bagian ini hanya berlaku jika bucket Anda dienkripsi dengan kunci KMS yang dikelola pelanggan. Jika bucket Anda menggunakan SSE-S3, lewati bagian ini.

Jika Anda menentukan kunci AWS KMS Anda sendiri untuk enkripsi bucket, tambahkan pernyataan berikut ke kebijakan kunci Anda. AWS Transform bergantung pada Forward Access Sessions (FAS) untuk membuat hibah untuk prinsipal layanannya. Hibah ini digunakan untuk memvalidasi kunci Anda dan mengakses artefak di bucket Amazon S3 Anda.

```
{
    "Sid": "AllowAWSTransformServiceAccess",
    "Effect": "Allow",
    "Principal": {
        "AWS": "arn:aws:iam::key-owner-account-id:root"
    },
    "Action": [
        "kms:CreateGrant",
        "kms:DescribeKey"
    ],
    "Resource": "*",
    "Condition": {
        "StringEquals": {
            "kms:ViaService": "transform.region.amazonaws.com"
        }
    }
}
```

Ganti nilai-nilai berikut:

- *key-owner-account-id*— Akun AWS ID yang memiliki kunci KMS.
- *region*— AWS Wilayah tempat AWS Transform diaktifkan (misalnya,`us-east-1`).

Beralih kembali ke bucket default

Anda dapat beralih kembali ke bucket yang dikelola layanan kapan saja.

Untuk beralih ke AWS Mengubah penyimpanan terkelola

1. Di konsol AWS Transform, pilih Pengaturan.
2. Di bawah Penyimpanan Artifact, pilih Edit.
3. Pilih AWS Transform managed storage.
4. Pilih Simpan.

Warning

Jika Anda mengganti konfigurasi penyimpanan saat pekerjaan transformasi sedang berlangsung, pekerjaan tersebut gagal. Artefak apa pun yang sudah dihasilkan oleh pekerjaan yang sedang berlangsung juga tidak akan dapat diakses.

Important

Saat Anda beralih kembali ke bucket yang dikelola layanan:

- Artefak di bucket Anda tidak dimigrasikan secara otomatis. Untuk menyimpan artefak apa pun, unduh artefak dari bucket Amazon S3 Anda dan unggah ulang melalui AWS aplikasi web Transform ke toko artefak untuk ruang kerja dan pekerjaan yang sesuai.
- AWS Transform menghentikan hibahnya pada kunci KMS Anda. Objek yang ditulis ke ember Anda tetap ada di ember Anda; hapus secara manual jika tidak diperlukan lagi.

Mengunggah file langsung ke bucket Anda

Anda dapat mengunggah file langsung ke bucket Amazon S3 Anda tanpa menggunakan aplikasi web AWS Transform. Untuk membuat file yang diunggah tersedia untuk agen transformasi, unggah ke `User Uploads` folder untuk pekerjaan itu. Jalur direktori untuk folder ini menggunakan format berikut:

```
AWSTransform/Workspaces/workspace-id/Jobs/job-id/User Uploads/
```

Ganti nilai-nilai berikut:

- *`workspace-id`*— ID ruang kerja AWS Transform Anda.
- *`job-id`*— ID pekerjaan transformasi.

Setiap pekerjaan di AWS Transform memiliki `User Uploads` foldernya sendiri. File yang Anda unggah ke jalur ini muncul di aplikasi web dan tersedia untuk agen transformasi saat mereka menyelesaikan pekerjaan.

Jalur file tidak boleh berisi`..`,`//`, atau spasi utama atau belakang — file yang melanggar batasan ini tidak terlihat oleh agen.

Important

Jika bucket Anda menggunakan SSE-KMS, sertakan header berikut pada setiap upload langsung:

```
x-amz-server-side-encryption: aws:kms
x-amz-server-side-encryption-aws-kms-key-id: your-kms-key-arn
```

Objek harus dienkripsi dengan kunci KMS yang terdaftar di profil Transform Anda AWS . Objek yang dienkripsi dengan kunci lain tidak dapat diakses oleh Transform. AWS

Matriks Kompatibilitas

Tabel berikut merangkum kompatibilitas AWS Transform dengan fitur bucket Amazon S3 yang umum.

Fitur	Kompatibilitas	Catatan
Pembuatan versi ember	Didukung	Mengelola versi noncurrent dengan aturan siklus hidup.
Kepemilikan Objek	Diperlukan: Pemilik bucket diberlakukan	Pemilik bucket lebih disukai dan penulis Object tidak didukung.
ACL	Tidak didukung	ACL harus dinonaktifkan di ember.
Kunci Objek (mode tata kelola)	Sebagian didukung	Penghapusan mungkin diblokir oleh pengaturan retensi.
Object Lock (mode kepatuhan)	Tidak didukung	—
Blokir Akses Publik	Didukung	—
Pembayaran oleh Pemohon	Tidak didukung	—
Titik Akses S3	Tidak didukung	Gunakan nama bucket.

Fitur	Kompatibilitas	Catatan
Titik Akses Lambda Objek S3	Tidak didukung	Gunakan nama bucket.
S3 di Outposts	Tidak didukung	—
S3 Express One Zone (ember direktori)	Tidak didukung	—
Cross-Region ember	Tidak didukung	—
Cross-account ember	Didukung	—
Cross-account Kuncinya KMS	Tidak didukung	—
Aturan siklus hidup	Didukung	—
CloudTrail peristiwa data	Didukung	—

Aktifkan AWS Transformasi

Untuk mengaktifkan AWS Transform:

1. Masuk ke Konsol AWS Manajemen.

2. Di bilah pencarian di bagian atas konsol, cari AWS Transform.
3. Pilih AWS Transform dari hasil pencarian.
4. Pilih Mulai untuk mengaktifkan layanan di Wilayah Anda saat ini.
5. Opsional: konfigurasikan Pusat Identitas IAM. Anda juga akan dapat memilih untuk menggunakan penyedia identitas pihak ketiga (iDP) di langkah selanjutnya.
6. Pilih kunci Enkripsi: kunci AWS terkelola default atau Sesuaikan pengaturan enkripsi.
7. Opsional: Gunakan bucket Amazon S3 Anda sendiri. Secara default, AWS Transform menggunakan bucket Amazon S3 yang dikelola layanan untuk menyimpan artefak transformasi.

Anda dapat memilih untuk menggunakan ember Anda sendiri sebagai gantinya. Untuk informasi selengkapnya, lihat Menggunakan bucket Amazon S3 Anda sendiri.

8. Pilih kemampuan AWS Transform yang ingin Anda aktifkan:
   - Antarmuka baris perintah (CLI), diperlukan untuk membuat dan menjalankan transformasi khusus. Untuk mengaktifkan CLI, lihat dan ikuti petunjuk unduhan.
   - Aplikasi web, antarmuka pengguna agen untuk modernisasi. Pilih Aktifkan aplikasi web untuk menggunakannya.
9. Pilih Aktifkan AWS Transformasi.
10. Opsional: pilih Aktifkan profil Tampilan untuk mengakses tab AWS Transform Users, Settings, dan Connectors, atau Kelola pengguna untuk mengelola pengguna.

    Anda dapat mengakses tab Pengguna, Pengaturan, dan Konektor kapan saja dengan memilih ikon menu di sudut kiri atas konsol.
11. Konfigurasikan akses Pengguna dengan memilih salah satu opsi berikut:
    - Pusat Identitas IAM
    - Penyedia identitas pihak ketiga (iDP)
    - IAM-only akses

    > **Note**
    >
    > Pilihan ini diselesaikan dan tidak dapat diubah saat Anda mengaktifkan AWS Transform.
12. Pilih Aktifkan aplikasi web.
13. Sistem menampilkan “Enabling AWS Transform” sementara itu menciptakan sumber daya yang diperlukan.

Setelah AWS Transform diaktifkan, tab Pengaturan menampilkan informasi berikut:

- URL aplikasi web - URL untuk mengakses aplikasi web AWS Transform
- Mulai URL untuk IDE - URL untuk mengakses AWS Transform di lingkungan pengembangan terintegrasi
- Wilayah - AWS Wilayah tempat AWS Transform diaktifkan

Mulai cepat: Mencoba AWS Transformasi

Cara termudah untuk mencoba AWS Transform adalah dengan AWS akun mandiri. Anda mungkin ingin melakukan ini sebagai bukti konsep atau untuk lingkungan pengujian. Gunakan prosedur ini:

1. Masuk ke Konsol AWS Manajemen.
2. Arahkan ke layanan AWS Transform.
3. Pilih Mulai untuk mengaktifkan layanan.
4. Pilih dan atur penyedia identitas Anda.
5. Tetapkan pengguna ke layanan AWS Transform.
6. Setelah layanan diaktifkan, Anda akan melihat URL aplikasi web AWS Transform.
7. Buka URL tersebut di jendela browser baru untuk mengakses pengalaman web AWS Transform.

Sekarang Anda siap untuk mengatur ruang kerja Anda.

Mengelola pengguna

Cara Anda mengelola akses pengguna ke AWS Transform bergantung pada model akses yang Anda pilih selama penyiapan. Jika Anda mengkonfigurasi IAM Identity Center, Anda menambahkan pengguna melalui IAM Identity Center. Jika Anda mengonfigurasi penyedia identitas pihak ketiga, Anda mengelola pengguna di penyedia tersebut. Jika Anda memilih IAM-only akses, akses pengguna dikelola melalui kebijakan IAM.

Menambahkan pengguna di Pusat Identitas IAM

Untuk menambahkan pengguna di Pusat Identitas IAM:

1. Arahkan ke konsol Pusat Identitas IAM.
2. Di panel navigasi, pilih Users (Pengguna).
3. Pilih Tambahkan pengguna.
4. Masukkan informasi yang diperlukan:
    - Nama pengguna - Pengidentifikasi unik untuk pengguna (tidak dapat diubah nanti)
    - Alamat email - Alamat email pengguna
    - Nama depan dan Nama belakang - Nama pengguna

- Nama tampilan - Nama yang muncul dalam daftar pengguna

5. Untuk Kata Sandi, pilih cara pengguna menerima kata sandi mereka:
   - Kirim email - Kirim instruksi pengaturan melalui email
   - Hasilkan kata sandi satu kali - Buat kata sandi untuk dibagikan secara manual
6. Pilih Berikutnya untuk meninjau informasi pengguna.
7. Tinjau detailnya dan pilih Tambahkan pengguna.

Setelah pengguna ditambahkan, mereka akan menerima undangan email untuk menyiapkan akun Pusat Identitas IAM mereka. Tautan undangan berlaku selama 7 hari.

Anda juga dapat mempelajari tentang bekerja dengan IAM Identity Center dan AWS Transform dalam video ini:

VideoTitle

Menambahkan pengguna ke AWS Transformasi

Setelah menambahkan pengguna ke Pusat Identitas IAM, Anda dapat memberi mereka akses ke AWS Transform:

1. Kembali ke konsol AWS Transform.
2. Di panel navigasi, pilih Pengguna dan grup.
3. Pilih tab Pengguna atau tab Grup.
4. Cari dan pilih pengguna atau grup yang ingin Anda tambahkan dari IAM Identity Center.
5. Pilih Tetapkan pengguna dan grup untuk memberikan akses kepada pengguna atau grup yang dipilih ke AWS Transform.

Setelah menambahkan pengguna, mereka muncul di daftar Pengguna dengan status “Tertunda” hingga mereka menerima undangan dan masuk.

Mengelola pengguna dengan IAM-only akses

Jika Anda mengonfigurasi AWS Transform dengan IAM-only akses, akses pengguna dikelola melalui kebijakan IAM. Setiap kepala sekolah IAM dengan `transform:AccessTransformProfile` izin pada sumber daya profil dapat mengakses AWS Transform.

Untuk memberikan akses kepada pengguna atau peran ke AWS Transform:

1. Arahkan ke konsol IAM.
2. Lampirkan kebijakan yang menyertakan `transform:AccessTransformProfile` tindakan ke pengguna atau peran IAM. Untuk kebijakan contoh, lihat Mengizinkan pengguna mengakses AWS Transform dengan kredensyal IAM.

Untuk mencabut akses, hapus kebijakan dari pengguna atau peran IAM.

Memahami izin kolaborator

AWS Transform menggunakan model izin berbasis ruang kerja untuk mengontrol akses ke sumber daya dan tindakan. Setiap pengguna diberi peran tertentu dalam ruang kerja, yang menentukan tindakan apa yang dapat mereka lakukan. Seorang pengguna dapat memiliki peran yang berbeda di ruang kerja yang berbeda.

Peran pengguna

AWS Transform mendukung lima peran pengguna dalam setiap ruang kerja. Peran ini berlaku dalam konteks ruang kerja, dan pengguna akan diberi peran di setiap ruang kerja yang menjadi anggotanya. Izin akses yang ditentukan untuk setiap peran adalah ruang kerja agnostik, sehingga pengguna A dengan peran Administrator di ruang kerja A memiliki izin yang sama dengan pengguna B dengan peran Administrator di ruang kerja B.

Izin peran

Izin terperinci untuk setiap peran:

Tindakan	ResourceType	Admin	Penyetuju	Kontributor	ReadOnly
Buat	Ruang kerja	✓	✓	✓	✓
Daftar	Ruang kerja	✓	✓	✓	✓
Dapatkan	Ruang kerja	✓	✓	✓	✓
Perbarui	Ruang kerja	✓	✗	✗	✗

Tindakan	ResourceType	Admin	Penyetuju	Kontributor	ReadOnly
Delete	Ruang kerja	✓	✗	✗	✗
Buat	ChatMessage	✓	✓	✓	✗
Baca	ChatMessage	✓	✓	✓	✓
Buat	RoleAssociation	✓	✗	✗	✗
Baca	RoleAssociation	✓	✓	✓	✓
Perbarui	RoleAssociation	✓	✗	✗	✗
Delete	RoleAssociation	✓	✗	✗	✗
Baca	CriticalHitlTask	✓	✓	✓	✓
Perbarui	CriticalHitlTask	✓	✓	✗	✗
Delete	CriticalHitlTask	✓	✓	✗	✗
Baca	HitlTask	✓	✓	✓	✓
Perbarui	HitlTask	✓	✓	✓	✗
Delete	HitlTask	✓	✓	✓	✗
Buat	Pekerjaan	✓	✓	✓	✗
Baca	Pekerjaan	✓	✓	✓	✓
Perbarui	Pekerjaan	✓	✓	✓	✗
Delete	Pekerjaan	✓	✓	✓	✗
Baca	Worklog	✓	✓	✓	✓
Buat	Artifact	✓	✓	✓	✗
Baca	Artifact	✓	✓	✓	✓

Tindakan	ResourceType	Admin	Penyetuju	Kontributor	ReadOnly
Perbarui	Artifact	✓	✓	✓	✗
Delete	Artifact	✓	✓	✓	✗
Buat	Konektor	✓	✓	✓	✗
Baca	Konektor	✓	✓	✓	✓
Perbarui	Konektor	✓	✓	✓	✗
Delete	Konektor	✓	✓	✓	✗

Human-in-the-loop (HITL) tindakan

AWS Transform menyediakan dua jenis tindakan HITL - standar dan kritis:

Tindakan HITL standar

Ini adalah tindakan rutin yang dapat dilakukan oleh pengguna dengan peran Kontributor, Penyetuju, atau Administrator.

Tindakan HITL kritis

Ini adalah tindakan dengan dampak signifikan, dan karenanya memerlukan tingkat izin yang lebih tinggi. Contohnya termasuk:

- Menggabungkan kode ke cabang utama
- Melakukan dekomposisi grafik
- Menerapkan kode ke lingkungan produksi

Tindakan HITL kritis hanya dapat dilakukan oleh pengguna dengan peran Penyetuju atau Administrator.

Untuk memastikan ada perbedaan antara tindakan HITL Standar dan HITL Kritis dalam kebijakan AuthZ, AWS Transform menyediakan dua API HITL terpisah, satu untuk menyelesaikan tindakan HITL standar, dan satu untuk menyelesaikan tindakan HITL kritis.

AWS Mengubah lingkungan

AWS Transform adalah layanan agen yang menggunakan pemrosesan bahasa alami untuk membantu Anda merencanakan dan menjalankan transformasi beban kerja Anda. Anda berinteraksi dengan AWS Transform terutama melalui antarmuka percakapan, di mana layanan beradaptasi dan merespons berdasarkan konteks diskusi Anda. Misalnya, Anda dapat memulai dengan menjelaskan arsitektur dan tujuan transformasi Anda saat ini menggunakan bahasa sehari-hari, seperti “Saya perlu memigrasikan VM lokal saya ke EC2.”

Saat Anda berkomunikasi dengan AWS Transform, layanan ini membuat rencana kerja khusus yang sesuai dengan kebutuhan spesifik Anda. Alur percakapan dinamis dan didorong oleh Anda, memungkinkan Anda untuk menyempurnakan rencana transformasi Anda secara berulang. Anda dapat memodifikasi rencana pekerjaan selama percakapan dengan membuat permintaan seperti “tambahkan fase pengujian sebelum pemotongan produksi” atau “hapus langkah cadangan karena kami sudah memiliki solusi.” Anda juga dapat kembali ke tugas sebelumnya dan melakukannya lagi. AWS Transform terus memperbarui rencana berdasarkan masukan Anda, memastikan strategi transformasi akhir memenuhi kebutuhan teknis dan bisnis Anda.

Inilah yang Anda lihat saat membuka AWS Transform.

- Lihat panel kontrol: Ini adalah panel sempit di sebelah kiri Transform. AWS Anda dapat memilih salah satu ikon untuk memilih tampilan apa yang ditampilkan di sebelah kanan panel kontrol tampilan. Arahkan kursor ke setiap ikon untuk tip alat yang menjelaskan tampilan. Tampilan standar yang diberikan, dari atas ke bawah, adalah:
  - Rencana Job
  - Dasbor
  - Persetujuan
  - Artefak
  - Worklog

  Beberapa alur kerja memberikan tampilan tambahan.

  Ketika Anda bekerja dalam rencana kerja, pandangannya adalah:
  - Obrolan
  - Persetujuan
  - Toko Artifak
  - Worklog

Saat Anda berada di ruang kerja, tampilan ada di atas:

- Lowongan
- Artefak
- Kolaborator (Pengguna)
- Konektor
- Pengaturan

- Panel Tampilan berada di sebelah kontrol tampilan.
- Panel Obrolan ada di tengah. Di sinilah Anda melakukan percakapan dengan AWS Transform.

> **Note**
>
> Pengguna dengan izin hanya-baca tidak dapat mengirim pesan dalam obrolan.

- Di sebelah kanan adalah panel Kolaborasi. Ini muncul ketika aktivitas human in the loop (HITL) dilakukan, seperti:
  - Mengunggah file data
  - Meninjau informasi dan rencana yang disediakan oleh Transform AWS

Pengaturan bahasa

AWS Transform mendukung pelokalan untuk alur kerja tertentu. Anda dapat mengubah bahasa tampilan aplikasi web dengan memilih ikon Pengaturan di sudut kanan atas dan memilih bahasa pilihan Anda dari menu Bahasa.

Bahasa-bahasa berikut didukung:

- Inggris (US)
- Inggris (UK)
- Deutsch (Jerman)
- Español (Spanyol)
- Français (Prancis)
- (Bahasa Jepang)
- Bahasa Indonesia (Indonesia)

- Italiano (Italia)
- Português (Portugis)
- (Korea)
- () (Mandarin Sederhana)
- () (Tradisional Tionghoa)
- Türkçe (Turki)

Alur kerja berikut mendukung pelokalan:

- Migrasi (termasuk VMware)

> **Note**
>
> Alur kerja AWS Transform lainnya saat ini hanya tersedia dalam bahasa Inggris. Dukungan pelokalan untuk alur kerja tambahan dapat ditambahkan di pembaruan di masa mendatang.

Mulai proyek Anda

AWS Transform memandu Anda melalui proyek modernisasi dan migrasi Anda. Untuk memulai:

1. Buat ruang kerja untuk meng-host proyek Anda. Pada halaman Workspaces, pilih Buat ruang kerja. Ikuti instruksi dan kemudian buka ruang kerja.
2. Di panel Obrolan, ketik “buat pekerjaan”
3. Pilih pekerjaan, dan ikuti petunjuk yang diberikan oleh AWS Transform. Jika input Anda diperlukan, panel Kolaborasi akan muncul dan menjelaskan apa yang diperlukan.

Fleksibilitas alur kerja

Lingkungan alur kerja AWS Transform memberikan fleksibilitas dalam perkembangan proyek modernisasi Anda. Menggunakan bahasa alami di panel obrolan, Anda dapat:

- Coba lagi tugas yang telah diselesaikan AWS Transform, berikan input manusia-in-the-loop yang berbeda di sepanjang jalan.
- Jalankan kembali pekerjaan, misalnya, jika ada perubahan dalam sumber modernisasi Anda.

Mengobrol dengan AWS Transformasi

AWS Transform chat tersedia di setiap tahap proyek Anda. Untuk membuka obrolan, klik ikon heksagonal ungu di sudut kanan bawah konsol web.

Obrolan ada di sana bagi Anda untuk menanyakan apa saja. Misalnya, Anda dapat meminta obrolan untuk menjelaskan konsep, memandu Anda melalui proses, menjelaskan permintaan atau respons AWS Transform, atau menjelaskan laporan AWS Transform.

> **Note**
>
> Pengguna dengan izin hanya-baca tidak dapat mengirim pesan dalam obrolan.

AWS Ubah integrasi obrolan

AWS Transform chat terintegrasi dengan:

Experience-Based Akselerasi

Experience-BasedAkselerasi (EBA) ditawarkan melalui obrolan AWS Transform. Ini memungkinkan Anda untuk melakukan penilaian EBA untuk beban kerja Windows dan menghasilkan rencana. Anda dapat memulai dengan mengimpor hasil penilaian dari CAST. Ini membantu Anda menemukan portofolio aplikasi Anda, setelah itu Anda dapat menggunakan AWS Transform chat untuk memilih aplikasi yang memenuhi kebutuhan bisnis Anda (misalnya, memfilter aplikasi berdasarkan kompleksitas atau baris kode). Anda kemudian dapat melakukan penilaian lebih dalam dari aplikasi yang dipilih dan menghasilkan rencana modernisasi.

AWS Hitung Mundur Premium

AWS Countdown Premium (CDP) yang terintegrasi ke dalam AWS Transform memberikan panduan ahli yang berkelanjutan dengan dukungan teknik yang ditunjuk. AWS Insinyur CDP Anda yang ditunjuk menyelam jauh ke dalam tumpukan teknologi Anda, memberikan dukungan yang dipersonalisasi dan sadar konteks saat masalah muncul. Pengguna dapat memanfaatkan AWS Transform untuk mencatat tiket dukungan langsung dari antarmuka obrolan. Tiket dukungan menyematkan rincian worklog dan rencana pekerjaan untuk membantu dukungan memahami konteks pelanggan. Jika pelanggan atau mitra adalah bagian dari Program CDP, tiket dukungan secara otomatis akan diarahkan ke CDP Engineer yang ditunjuk yang dapat membantu masalah debug, menafsirkan output transformasi, dan mendorong kemajuan untuk penyelesaian masalah yang dipercepat. Contoh:

- Untuk beban kerja.NET, CDP dapat membantu masalah konektor repositori, resolusi ketergantungan, dan penerapan pasca-transformasi.
- Dalam skenario mainframe, CDP dapat membantu memecahkan masalah kode Java refactored, mengkonfigurasi alat migrasi database, dan membangun pipeline. CI/CD
- Dalam migrasi VMware, CDP dapat mempercepat konfigurasi jaringan, penyiapan MGN, dan instalasi agen.

AWS Pembangun Keterampilan

AWS Skill Builder menyediakan modul pembelajaran yang relevan melalui obrolan. Anda dapat menanyakan obrolan AWS Transform tentang kebutuhan belajar Anda, dan itu menyajikan katalog kursus yang relevan. Skill Builder memberikan pengalaman pembelajaran mikro kontekstual saat Anda bekerja melalui tahap transformasi.

Alat developer

Anda dapat mengakses agen AWS Transform dari IDE agen, plugin agen, dan server MCP. Pilih permukaan yang sesuai dengan alur kerja Anda.

> **Note**
>
> Plugin Kiro Power dan agen secara otomatis menginstal server MCP. Anda hanya perlu menginstal server MCP secara manual jika Anda ingin menggunakannya tanpa Kiro Power atau plugin agen.

Kekuatan Kiro

AWS Transform Kiro Power menyediakan akses penuh ke agen AWS Transform dari Kiro IDE.

Untuk menginstal AWS Transformasi Kekuatan Kiro

1. Buka Kiro IDE dan arahkan ke panel Powers.
2. Temukan AWS Transform dalam daftar dan instal.
3. Buka Kiro Chat, klik pada Power, dan pilih Try power.

Atau, di panel Powers pilih Tambahkan Daya Kustom > Impor daya dari GitHub dan tempel: `https://github.com/kirodotdev/powers/tree/main/aws-transform`

Plugin agen

Plugin agen AWS Transform tersedia di awslabs/agent-plugins on GitHub, di direktori. `plugins/aws-transform` Plugin ini mendukung Claude Code, Codex, dan Cursor.

Plugin ini mencakup`.claude-plugin`,`.codex-plugin`, dan `.mcp.json` konfigurasi. Ini menyediakan akses ke semua agen AWS Transform, seperti.NET, mainframe, VMware, SQL Server, dan transformasi kustom.

Untuk menginstal di Claude Code

Jalankan perintah berikut:

```
/plugin marketplace add awslabs/agent-plugins
/plugin install aws-transform@agent-plugins-for-aws
```

Untuk menginstal di Codex atau Cursor

Ikuti petunjuk di repositori agen-plugin README.

Plugin IDE

Plugin AWS Transform IDE tersedia di VS Code dan editor yang kompatibel dengan Open VSX. Ini mengekspos fitur khusus AWS Transform, termasuk membuat dan menjalankan definisi transformasi. Ini juga memaparkan fitur mainframe, termasuk keterlacakan antara persyaratan modernisasi, aturan bisnis, dan kode sumber, serta kemampuan untuk menghasilkan dokumentasi teknis sesuai permintaan. Untuk memulai, instal menggunakan salah satu metode berikut:

- VS Code: Instal dari VS Code Marketplace.
- Buka VSX: Instal dari registri Open VSX.

Server MCP

Gunakan server MCP untuk mengintegrasikan AWS Transform secara terprogram dengan klien mana pun. MCP-compatible Anda dapat menginstal server MCP dari `awslabs.aws-transform-mcp-server` PyPI sebagai. Ini menyediakan 19 alat untuk manajemen ruang kerja, manajemen pekerjaan, tugas human-in-the-loop (HITL), penanganan artefak, manajemen konektor, obrolan, dan penelusuran sumber daya.

Prasyarat

- Python 3.10 atau yang lebih baru

Penginstalan

Kode Claude

```
claude mcp add awslabs.aws-transform-mcp-server -- uvx awslabs.aws-transform-mcp-server@latest
```

Kiro

Tambahkan ke`~/.kiro/settings/mcp.json`:

```
{
  "mcpServers": {
    "awslabs.aws-transform-mcp-server": {
      "command": "uvx",
      "args": ["awslabs.aws-transform-mcp-server@latest"]
    }
  }
}
```

Untuk detail konfigurasi untuk klien MCP spesifik Anda, lihat AWS-transform-mcp-server README aktif. GitHub

Autentikasi

AWS Transform mendukung SSO (AWS Identity and Access Management Identity Center) dan autentikasi berbasis peran IAM.

SSO (Pusat Identitas IAM)

Minta asisten AI Anda untuk “Konfigurasi AWS Transformasi dengan SSO.” Asisten menyediakan URL awal dan membuka browser Anda untuk login.

Peran IAM

Lingkungan Anda menyediakan AWS kredensil secara otomatis. Setel `AWS_PROFILE` di `env` blok konfigurasi klien MCP Anda untuk memilih profil tertentu.

Untuk informasi tentang menyiapkan otentikasi untuk AWS Transform, lihatMemulai dengan AWS Transformasi.

Membangun agen kustom untuk AWS Transformasi

Toolkit pembangun agen memungkinkan AWS Mitra dan pelanggan untuk membangun agen yang disesuaikan dengan kebutuhan modernisasi spesifik mereka yang berfungsi dalam Transform. AWS

Untuk menginstal toolkit pembangun agen

1. Buka Kiro IDE dan arahkan ke panel Powers.

2. Temukan AWS Transform Agent Toolkit dalam daftar dan instal.

Atau, di panel Powers pilih Tambahkan Daya Kustom > Impor daya dari GitHub dan tempel: `https://github.com/kirodotdev/powers/tree/main/aws-transform-agent-toolkit`

Toolkit pembuat agen memungkinkan Mitra Kompetensi Migrasi dan Modernisasi, ISV, atau pelanggan untuk membuat solusi transformasi untuk kasus penggunaan tertentu. Anda dapat mengintegrasikan agen, alat, basis pengetahuan, dan alur kerja khusus dengan kemampuan AI agen AWS Transform.

Siklus hidup agen

Build

Gunakan Kiro Power untuk menggabungkan agen dasar dan SDK untuk menulis agen khusus Anda sendiri.

Bagikan

Mendistribusikan agen dengan tim atau di seluruh jaringan mitra.

Pendaftaran

Daftarkan agen dengan AWS Transform untuk ditemukan oleh pengguna lain.

Penilaian migrasi

AWS Transform assessment membantu Anda mengevaluasi biaya, kelayakan, dan nilai bisnis dari migrasi infrastruktur lokal. AWS Penilaian memberikan rekomendasi ukuran kanan otomatis, perbandingan multi-skenario, dan penyempurnaan interaktif melalui obrolan.

Anda dapat menggunakan penilaian migrasi untuk:

- Dapatkan perkiraan biaya untuk Amazon RDS for SQL Server, Amazon EC2, Amazon EBS, Amazon S3, dan Amazon FSx
- Menerima rekomendasi ukuran kanan otomatis
- Menilai nilai bisnis dan dampak keberlanjutan
- Bandingkan beberapa skenario migrasi secara berdampingan
- Sempurnakan penilaian secara interaktif melalui obrolan
- Hasilkan presentasi eksekutif dan laporan terperinci

Prasyarat

Sebelum membuat penilaian migrasi, konfirmasikan bahwa Anda memiliki yang berikut:

- Ruang kerja AWS Transform
- Inventarisasi data dalam format yang didukung, atau Anda dapat menggunakan estimasi kasar melalui obrolan
- Izin yang sesuai untuk membuat dan menjalankan penilaian

Cara kerja penilaian migrasi

Penilaian migrasi mengikuti alur kerja terstruktur yang memandu Anda dari unggahan data awal hingga pembuatan hasil akhir yang dapat dikirim.

Untuk menyelesaikan penilaian migrasi

1. Buat pekerjaan penilaian migrasi di ruang kerja AWS Transform Anda.
2. Unggah data inventaris lokal Anda.

3. Tinjau hasil penemuan yang dihasilkan AWS Transform dari data Anda.
4. Kelola ruang lingkup inventaris dengan mengecualikan server atau menyesuaikan pengelompokan.
5. Konfigurasikan skenario penilaian Anda dengan asumsi harga dan infrastruktur.
6. Jalankan penilaian.
7. Tinjau dan perbaiki hasilnya melalui obrolan.

Setelah meninjau hasil, Anda dapat membuat skenario tambahan, membandingkan skenario, dan menghasilkan kiriman seperti presentasi dan laporan.

Mengunggah data inventaris

AWS Transform menerima data inventaris dari berbagai sumber. Tabel berikut mencantumkan format data yang didukung.

Sumber data	Deskripsi
AWS Transform ekspor Alat Penemuan	Inventaris server otomatis ditemukan oleh alat penemuan AWS Transform
RVTools	Ekspor dari lingkungan VMware dalam ZIP/CSV atau format Excel. Baik ekspor penuh dan Info-only ekspor v didukung.
Data CMDB	Ekspor basis data manajemen konfigurasi
Alat penemuan mitra	Data dari alat penemuan AWS mitra
Penilai Migrasi	File Wawasan Cepat dari konsol Penilai Migrasi
Format MPA	File impor Penilaian Portofolio Migrasi
AWS Ubah templat data	File Microsoft Excel yang dibuat dari template AWS Transform Assessment Data

AWS Transform secara otomatis mengidentifikasi format file selama konsumsi. Proses penyerapan memvalidasi data Anda, menerima sebagian data saat beberapa bidang hilang, dan mendukung

unggahan tambahan. Anda dapat mengunggah file tambahan kapan saja untuk melengkapi inventaris Anda.

Meninjau hasil penemuan

Setelah AWS Transform memproses data inventaris Anda, Transform menghasilkan ringkasan penemuan. Ringkasan mencakup informasi berikut:

- Jumlah server berdasarkan sistem operasi
- Kerusakan server fisik dan virtual
- Deteksi SQL Server
- Ringkasan penyimpanan
- Peringatan kualitas data

Tinjau hasil penemuan untuk mengonfirmasi bahwa AWS Transform mengidentifikasi infrastruktur Anda dengan benar sebelum Anda melanjutkan penilaian.

Mengelola ruang lingkup inventaris

Setelah Anda meninjau hasil penemuan, Anda dapat menyempurnakan ruang lingkup penilaian Anda. Gunakan tindakan berikut untuk mengelola cakupan inventaris Anda:

- Inventaris kueri untuk server atau beban kerja tertentu
- Kecualikan server dari lingkup penilaian
- Tinjau pengelompokan server dan dependensi aplikasi
- Identifikasi beban kerja SQL Server untuk penilaian khusus

Mengkonfigurasi skenario penilaian

Sebelum Anda menjalankan penilaian, konfigurasikan asumsi yang digunakan AWS Transform untuk menghasilkan rekomendasi. Anda dapat memodifikasi asumsi ini kapan saja dan menjalankan skenario baru dengan konfigurasi yang berbeda.

Tabel berikut mencantumkan kategori asumsi yang tersedia.

Asumsi	Deskripsi
Model penentuan harga	Database Savings Plans (untuk Amazon RDS for SQL Server) On-Demand, atau Instans Cadangan
Target Wilayah AWS	Wilayah AWS Tempat Anda berencana untuk meng-host beban kerja yang dimigrasi
Pengecualian jenis instans	Keluarga atau tipe instans Amazon EC2 untuk dikecualikan dari rekomendasi
Konfigurasi Amazon EBS	Jenis volume dan pengaturan kinerja untuk penyimpanan
Lisensi SQL Server	Bawa Media Anda Sendiri (BYOM) atau Lisensi Termasuk (LI) untuk Amazon RDS untuk SQL Server; Termasuk Lisensi (LI) atau Bawa Lisensi Anda Sendiri (BYOL) untuk SQL Server di EC2

Kemampuan penilaian

AWS Penilaian transformasi mencakup beberapa dimensi migrasi Anda. Bagian berikut menjelaskan setiap kemampuan penilaian.

Ukuran kanan Amazon EC2

AWS Transform menganalisis spesifikasi server lokal Anda dan merekomendasikan instans Amazon EC2 berukuran tepat. Rekomendasi memperhitungkan persyaratan CPU, memori, dan kinerja.

Penyimpanan Amazon EBS

AWS Transform merekomendasikan jenis dan konfigurasi volume Amazon EBS berdasarkan penggunaan penyimpanan dan persyaratan kinerja Anda saat ini.

Amazon FSx

AWS Transform mengevaluasi beban kerja penyimpanan file dan memberikan rekomendasi untuk migrasi Amazon FSx, termasuk perkiraan biaya untuk jenis sistem file Amazon FSx yang didukung.

Amazon RDS for SQL Server

AWS Transform menilai biaya migrasi database SQL Server lokal ke Amazon RDS for SQL Server. Menggunakan AI-powered agen, AWS Transform menganalisis lingkungan SQL Server lokal Anda dan mengirimkan kasus bisnis migrasi lengkap dalam hitungan menit, dengan rekomendasi komputasi dan memori yang disesuaikan dengan persyaratan beban kerja sehingga Anda menghindari penyediaan berlebih dan hanya membayar untuk apa yang Anda butuhkan.

Penilaian RDS untuk SQL Server mencakup kemampuan berikut:

- Bawa lisensi Media Anda Sendiri (BYOM), memungkinkan Anda untuk menggunakan lisensi SQL Server yang ada
- Lisensi Termasuk (LI) opsi lisensi
- Optimalisasi biaya menggunakan Database Savings Plans, yang menawarkan penghematan hingga 20% dibandingkan dengan On-Demand harga
- Penilaian kelayakan untuk AWS Migration Acceleration Program (MAP), yang memberikan kredit dan dukungan untuk mengimbangi biaya migrasi

Anda dapat memulai penilaian RDS untuk SQL Server dengan format data apa pun yang didukung, termasuk ekspor RVTools, data database manajemen konfigurasi (CMDB), ekspor dari alat penemuan AWS Transform, dan alat penemuan pihak ketiga lainnya. Buat skenario bagaimana-jika untuk membandingkan beberapa model biaya dengan asumsi yang disesuaikan termasuk wilayah, pemanfaatan sumber daya, dan persyaratan harga.

Contoh petunjuk:

- “Perkirakan biaya migrasi database SQL Server saya ke RDS untuk SQL Server”
- “Bandingkan harga BYOM vs Lisensi Termasuk untuk RDS untuk SQL Server”
- “Tunjukkan opsi Database Savings Plans untuk beban kerja RDS saya”

Server SQL Amazon EC2

AWS Transform menilai beban kerja SQL Server dan memberikan rekomendasi untuk menjalankan SQL Server di Amazon EC2, termasuk analisis lisensi dan pemetaan host khusus.

On-premises harga

AWS Transform memperkirakan biaya lokal Anda saat ini untuk memberikan garis dasar untuk perbandingan dengan harga. AWS Anda dapat melakukan penyesuaian biaya secara keseluruhan melalui obrolan. On-premises penyesuaian biaya tercermin dalam laporan PDF dan tanggapan obrolan, tetapi tidak dalam ekspor PPTX.

Contoh petunjuk:

- “Perbarui biaya server lokal menjadi $500 per server per bulan”
- “Tambahkan biaya fasilitas pusat data tahunan $50.000”

Keberlanjutan

AWS Transform memperkirakan pengurangan jejak karbon yang dihasilkan dari migrasi ke. AWS Anda dapat menjelajahi metrik keberlanjutan melalui obrolan.

Coba petunjuk seperti:

- “Tunjukkan perkiraan pengurangan karbon untuk migrasi ini”
- “Apa peningkatan efisiensi energi untuk beban kerja saya AWS?”

Penilaian nilai bisnis

AWS Transform mengevaluasi nilai bisnis migrasi yang lebih luas di luar penghematan biaya infrastruktur. Penilaian tersebut mencakup produktivitas staf, ketahanan, dan kelincahan bisnis.

Berikut adalah beberapa contoh petunjuk:

- “Perkirakan peningkatan produktivitas staf dari migrasi ke AWS”
- “Peningkatan ketahanan apa yang dapat saya harapkan setelah migrasi?”
- “Tunjukkan manfaat kelincahan bisnis dari migrasi ini”

Biaya jaringan

AWS Transform memperkirakan biaya terkait jaringan untuk migrasi Anda, termasuk transfer data dan persyaratan konektivitas.

Biaya Support

AWS Transform mencakup biaya paket AWS Support dalam penilaian berdasarkan tingkat dukungan yang Anda pilih.

Komputasi pengguna akhir

AWS Transform menilai beban kerja komputasi pengguna akhir dan memberikan rekomendasi untuk layanan komputasi pengguna AWS akhir.

Menggunakan penilaian berbasis obrolan

Anda dapat berinteraksi dengan AWS Transform melalui obrolan untuk membuat dan menyempurnakan penilaian. Obrolan mendukung tiga mode interaksi.

Anda dapat memperbaiki penilaian Anda secara berulang dengan melanjutkan percakapan. AWS Transform mempertahankan konteks di seluruh pesan dan memperbarui hasil penilaian berdasarkan masukan Anda.

Estimasi kasar dengan data terbatas

Anda bisa mendapatkan perkiraan biaya kasar dengan menjelaskan lingkungan Anda dalam obrolan tanpa mengunggah file inventaris terperinci. Estimasi kasar adalah fitur mandiri dan tidak dapat digabungkan dengan data inventaris yang diunggah atau pengayaan data melalui obrolan.

Anda dapat menggunakan petunjuk seperti ini:

- “Saya memiliki 200 server Windows dan 150 server Linux, perkirakan AWS biaya saya”
- “Beri saya perkiraan kasar untuk memigrasikan 500 VM ke” AWS
- “Perkirakan biaya untuk 50 server dengan rata-rata 8 CPU dan 32 GB RAM”

Menambahkan inventaris melalui obrolan

Anda dapat memberikan detail inventaris secara langsung melalui obrolan untuk melengkapi atau mengganti file yang diunggah.

Gunakan petunjuk seperti berikut ini:

- “Sertakan server database yang menjalankan Oracle pada 4 CPU dengan 128 GB RAM”
- “Tambahkan 20 server web yang menjalankan Linux dengan 4 CPU dan 16 GB RAM”

Memodifikasi biaya dan menambahkan layanan

Anda dapat menyesuaikan biaya lokal dan menambahkan AWS layanan ke penilaian Anda melalui obrolan.

Contoh permintaan untuk penyesuaian lokal:

- “Tingkatkan biaya lokal sebesar 15% untuk memperhitungkan penyegaran perangkat keras yang akan datang”
- “Tambahkan biaya pemeliharaan tahunan $100.000 ke baseline lokal”

Anda juga dapat menambahkan perkiraan biaya kasar untuk AWS layanan yang tidak sepenuhnya didukung oleh penilaian AWS Transform. Ini memberikan analisis yang lebih lengkap, tetapi perkiraan ini kurang akurat daripada rekomendasi otomatis.

- “Tambahkan biaya AWS Backup untuk semua server yang dimigrasi”
- “Sertakan biaya CloudWatch pemantauan Amazon dalam perkiraan”
- “Tambahkan biaya AWS Direct Connect untuk koneksi 10 Gbps”

Membandingkan skenario

Anda dapat membuat beberapa skenario penilaian dengan asumsi berbeda dan membandingkannya untuk mengidentifikasi strategi migrasi yang optimal.

Membuat skenario

Coba petunjuk seperti:

- “Buat skenario BYOM untuk memigrasi database SQL Server ke RDS untuk SQL Server”
- “Buat skenario dengan Database Savings Plans untuk RDS untuk SQL Server”
- “Buat skenario LI untuk semua beban kerja SQL Server di EC2"
- “Buat skenario dengan semua beban kerja di us-west-2"

Menjalankan perbandingan

Berikut adalah beberapa contoh petunjuk:

- “Tunjukkan perbandingan biaya di semua skenario”
- “Skenario mana yang memiliki total biaya kepemilikan terendah?”

What-if analisis

Modelkan dampak perubahan spesifik tanpa membuat skenario baru yang lengkap.

Contoh petunjuk:

- “Bagaimana jika saya memindahkan beban kerja SQL Server saya ke RDS untuk SQL Server alih-alih EC2?”
- “Apa perbedaan biaya antara BYOM dan Lisensi Termasuk untuk RDS untuk SQL Server?”
- “Bagaimana jika saya mengecualikan 50 server terkecil dari migrasi?”
- “Bagaimana jika saya hanya menggunakan instance penyimpanan yang dioptimalkan?”
- “Apa dampak perpindahan dari volume gp3 ke io2?”

Menghasilkan kiriman

Setelah menyelesaikan penilaian, Anda dapat menghasilkan kiriman dalam berbagai format. Tabel berikut menjelaskan format output yang tersedia.

Format	Deskripsi	Kustomisasi
PPTX () PowerPoint	Presentasi eksekutif dengan ringkasan temuan dan rekomendasi	Struktur tetap
XLSX (Excel)	Ekspor data terperinci dengan rekomendasi tingkat server dan rincian biaya	Struktur tetap
PDF	Melaporkan dokumen	Dapat disesuaikan melalui obrolan

Topik terkait

- Memulai
- Lowongan kerja kustom
- Alat penemuan

Khusus

Apa itu AWS Ubah kustom?

AWS Transform custom menggunakan agentic AI untuk melakukan modernisasi skala besar perangkat lunak, kode, pustaka, dan kerangka kerja untuk mengurangi utang teknis. Ini menangani beragam skenario termasuk peningkatan versi bahasa, migrasi API dan layanan, peningkatan dan migrasi kerangka kerja, refactoring kode, dan transformasi khusus organisasi.

Melalui pembelajaran berkelanjutan, agen meningkatkan dari setiap eksekusi dan umpan balik pengembang, memberikan transformasi berkualitas tinggi dan berulang tanpa memerlukan keahlian otomatisasi khusus.

Kemampuan Kunci

AWS Transform custom menyediakan kemampuan berikut:

- Definisi transformasi berbasis bahasa alami - Buat transformasi khusus menggunakan bahasa alami, dokumentasi, dan sampel kode
- Eksekusi transformasi - Terapkan transformasi secara konsisten dan andal di beberapa basis kode
- Pembelajaran berkelanjutan - Secara otomatis meningkatkan kualitas transformasi dari setiap eksekusi
- AWS-transformasi terkelola - Gunakan transformasi yang siap digunakan dan diperiksa untuk skenario AWS umum

Pola Transformasi

AWS Transform custom mendukung beragam pola transformasi untuk memenuhi kebutuhan modernisasi Anda. Setiap pola memiliki karakteristik kompleksitas yang berbeda berdasarkan ruang lingkup dan sifat perubahan yang diperlukan.

Pola	Deskripsi	Kompleksitas	Contoh
Migrasi API dan Layanan	Migrasi antara versi API atau layanan yang setara sambil	Sedang	AWS SDK v1 → v2 (Jawa, Python,), Boto2 → Boto3, JUnit

Pola	Deskripsi	Kompleksitas	Contoh
	mempertahankan fungsionalitas		4 → 5, JavaScript javax → jakarta
Upgrade Versi Bahasa	Memutakhirkan ke versi yang lebih baru dari bahasa pemrograman yang sama, mengadopsi fitur baru dan mengganti fungsionalitas yang tidak digunakan lagi	Low-Medium	Java 8 → 17, Python 3.9 → 3.13, 12 → 22, peningkatan versi Node.js TypeScript
Upgrade Kerangka	Memutakhirkan ke versi yang lebih baru dari kerangka kerja yang sama, mengatasi perubahan yang melanggar	Sedang	Spring Boot 2.x → 3.x, React 17 → 18, peningkatan sudut, peningkatan Django
Migrasi Kerangka	Bermigrasi ke kerangka kerja yang sama sekali berbeda yang melayani tujuan serupa	Tinggi	Angular → Bereaksi, Redux → Zustand, → Bereaksi Vue.js
Upgrade Perpustakaan dan Ketergantungan	Memutakhirkan pustaka pihak ketiga ke versi yang lebih baru sambil mempertahankan bahasa dan kerangka kerja yang sama	Low-Medium	Pandas 1.x → 2.x, NumPy peningkatan, peningkatan perpustakaan, peningkatan Lodash Hadoop/HBase/Hive

Pola	Deskripsi	Kompleksitas	Contoh
Refactoring Kode dan Modernisasi Pola	Memodernisasi pola kode dan mengadopsi praktik terbaik tanpa mengubah fungsionalitas eksternal	Low-Medium	Cetak → Kerangka kerja logging, penggabungan string → f-string, adopsi petunjuk tipe, instrumentasi observabilitas
Skrip dan File-by-File Terjemahan	Menerjemahkan skrip independen atau file konfigurasi di mana file sebagian besar mandiri	Low-Medium	AWS CDK → Terraform, Terraform →, Excel→not ebook Python, CloudFormation Bash→ PowerShell
Migrasi Arsitektur	Migrasi antara arsitektur perangkat keras atau lingkungan runtime dengan perubahan kode minimal	Medium-High	x86 →AWS Graviton (ARM), On-premis →Lambda, server tradisional → kontainer
Language-to-Language Migrasi	Menerjemahkan basis kode dari satu bahasa pemrograman ke bahasa pemrograman lainnya	Sangat tinggi	Java→Python, →, C → Karat, Python JavaScript → Go TypeScript
Kustom dan Organization-Specific Transformasi	Persyaratan organisasi yang unik dan kebutuhan modernisasi khusus	Bervariasi	Migrasi pustaka internal khusus, standar pengkodean khusus organisasi, migrasi kerangka kerja berpemilik

> **Note**
>
> Untuk COBOL/mainframe bahasa, gunakan AWS Transform for Mainframe. Untuk upgrade .NET Framework ke .NET Core, pertimbangkan AWS Transform untuk Windows. Untuk migrasi VMware, pertimbangkan AWS Transform for VMware. AWS

Bagaimana AWS Mengubah Karya kustom

AWS Transform custom biasanya digunakan dalam proyek skala besar di mana beberapa basis kode atau modul diubah. Tim biasanya mengikuti alur kerja empat fase:

Tentukan Transformasi - Berikan petunjuk bahasa alami, dokumentasi, dan sampel kode kepada agen, yang menghasilkan definisi transformasi awal. Definisi ini dapat disempurnakan secara iteratif melalui obrolan atau pengeditan langsung. Fase ini dapat dilewati saat menggunakan transformasi AWS-managed.

Pilot atau Proof-of-Concept - Uji transformasi pada basis kode sampel dan perbaiki berdasarkan hasil. Fase validasi ini membantu memperkirakan biaya dan upaya transformasi penuh. Pembelajaran berkelanjutan meningkatkan kualitas selama fase ini.

Eksekusi Berskala - Siapkan eksekusi massal otomatis menggunakan CLI, dengan pengembang meninjau dan memvalidasi hasil. Pantau kemajuan menggunakan aplikasi web dan melacak transformasi di beberapa repositori.

Monitor dan Tinjau - Pembelajaran berkelanjutan secara otomatis meningkatkan kualitas transformasi. Tinjau pelajaran yang telah diambil dari proses sebelumnya untuk memastikan mereka memenuhi standar kualitas, dan arsipkan apa pun yang tidak berguna.

Memahami Konsep Kunci

Bagian ini menjelaskan konsep-konsep kunci untuk bekerja dengan AWS Transform custom.

Definisi Transformasi

Definisi transformasi berisi instruksi dan pengetahuan yang diperlukan untuk melakukan transformasi kode tertentu. Ini direpresentasikan sebagai keterampilan dan terdiri dari:

- `SKILL.md`(wajib) - Instruksi dalam Penurunan Harga dengan frontmatter YAMAL (`name`edan `description` bidang) yang berisi logika transformasi inti dan instruksi eksekusi

- `references/`folder (opsional) - Dokumentasi dimuat sesuai kebutuhan selama eksekusi transformasi
- `scripts/`folder (opsional) - Skrip yang diunduh dan dijalankan transformasi selama eksekusi

AWS Transform CLI secara otomatis mengunduh definisi transformasi ke direktori saat ini bila diperlukan untuk eksekusi, inspeksi, atau modifikasi.

Important

Saat menerbitkan transformasi, direktori harus berisi hanya `SKILL.md` dan opsional `references/` folder, `scripts/` folder, atau keduanya. Tidak ada file atau subdirektori lain yang diizinkan.

Note

CLI versi 2.0 dan yang lebih baru menggunakan format keterampilan. Definisi transformasi yang ada secara otomatis ditangani oleh CLI.

Registri Transformasi

Registri transformasi adalah repositori terpusat AWS akun Anda untuk menyimpan dan mengelola definisi transformasi. Transformasi dalam registri dapat berupa:

- Terdaftar menggunakan `atx custom def list`
- Dieksekusi di beberapa basis kode
- Dibagikan dengan pengguna lain di AWS akun Anda
- Version-controlled

Important

Definisi transformasi bersifat spesifik akun. Jika Anda ingin menggunakan transformasi di AWS akun yang berbeda, Anda harus mempublikasikannya secara terpisah di akun itu.

Draf vs Transformasi yang Diterbitkan

AWS Transform custom mendukung dua status untuk transformasi dalam registri:

Transformasi draf adalah definisi transformasi yang sedang berlangsung atau belum teruji. Mereka disimpan sebagai versi tertentu dan dapat diambil, diperbarui, dan dieksekusi oleh pengguna yang mengacu pada versi tertentu. Draf berguna untuk pengembangan berulang, pengujian, dan penyempurnaan sebelum transformasi siap untuk dibagikan dengan tim Anda. Draf juga terkait dengan percakapan tertentu. Jika Anda me-restart CLI, Anda dapat menggunakan `atx --conversation-id {id}` untuk memulihkan yang sebelumnya.

Transformasi yang dipublikasikan tersedia di registri transformasi akun Anda untuk dieksekusi oleh pengguna lain dengan izin IAM yang diperlukan. Transformasi yang dipublikasikan dapat ditemukan menggunakan`atx custom def list`.

Alur kerja tipikal adalah:

1. Buat transformasi secara lokal
2. Simpan sebagai draf untuk pengujian (`atx custom def save-draft`)
3. Perbaiki dan validasi
4. Publikasikan untuk dibagikan dengan tim Anda (`atx custom def publish`)

Anda juga dapat mempublikasikan transformasi secara langsung tanpa menyimpannya sebagai draf.

Referensi vs. Pelajaran

AWS Transform custom menggunakan dua jenis pengetahuan untuk meningkatkan kualitas transformasi:

Referensi adalah dokumentasi yang disediakan pengguna yang disimpan dalam `references/` folder definisi transformasi. Referensi hanya mendukung file teks (total maksimum 10MB untuk semua file) dan biasanya berisi dokumentasi, spesifikasi API, panduan migrasi, dan contoh kode. Referensi dimuat sesuai kebutuhan selama eksekusi transformasi. Anda menambahkan referensi saat membuat atau memperbarui definisi transformasi dalam mode interaktif.

Pelajaran secara otomatis diekstraksi dari transformasi sebelumnya. Sistem pembelajaran berkelanjutan menghasilkannya dari lintasan eksekusi, umpan balik pengembang, dan perbaikan

kode yang ditemuinya selama transformasi. Ini mengelompokkan pelajaran terkait ke dalam kategori sehingga Anda dapat meninjaunya bersama. Tidak seperti referensi, yang Anda berikan di muka, pelajaran terakumulasi dari waktu ke waktu saat Anda menjalankan transformasi di berbagai basis kode, dan sistem menerapkannya secara otomatis untuk meningkatkan proses masa depan. Anda meninjau dan mengelola pelajaran secara interaktif dengan`atx custom def learnings`.

Perintah Build dan Validasi

Perintah build atau validasi adalah parameter opsional yang menentukan cara memvalidasi kode Anda selama proses transformasi. Perintah ini dijalankan di berbagai titik selama transformasi untuk memastikan integritas kode.

Memberikan perintah yang memvalidasi hasil dan mengembalikan masalah jika validasi gagal sangat penting untuk meningkatkan kualitas transformasi melalui pembelajaran berkelanjutan. Jika tidak diperlukan build atau validasi, hilangkan input Anda.

Untuk contoh dan panduan terperinci, lihat Perintah Membangun dan Validasi di bagian Alur Kerja.

Pembelajaran Terus Berkelanjutan

Pembelajaran berkelanjutan adalah sistem yang secara otomatis menangkap umpan balik dari setiap eksekusi transformasi dan meningkatkan kualitas transformasi dari waktu ke waktu. Sistem mengumpulkan informasi melalui:

- Umpan balik eksplisit - Komentar dan perbaikan kode disediakan dalam mode interaktif
- Pengamatan implisit - Masalah yang dihadapi agen saat mengubah dan men-debug kode

Sistem pembelajaran berkelanjutan memproses informasi ini untuk menciptakan pelajaran yang ditambahkan ke definisi transformasi untuk meningkatkan transformasi masa depan. Sistem menjalankan proses pembelajaran secara otomatis setelah transformasi selesai, tidak memerlukan masukan tambahan dari Anda.

> **Important**
>
> Pelajaran bersifat spesifik transformasi dan tidak dibagikan di berbagai transformasi atau akun pelanggan yang berbeda.

Mengelola Pelajaran

Sistem menghasilkan pelajaran secara otomatis, tetapi Anda tetap mengendalikan pelajaran mana yang berlaku untuk masa depan. `atx custom def learnings`Perintah membuka sesi terminal interaktif untuk meninjau dan mengkurasi pelajaran yang telah diakumulasikan oleh transformasi.

Dalam sesi ini, Anda dapat:

- Jelajahi pelajaran yang dikelompokkan ke dalam kategori, dan lihat berapa banyak pelajaran aktif yang terkandung di setiap kategori.
- Buka pelajaran untuk membaca detail lengkapnya, termasuk badan pelajaran, dampaknya, dan berapa banyak proses sebelumnya yang dikonsultasikan.
- Arsipkan pelajaran untuk mencegah sistem menerapkannya ke future run, dan mengembalikannya nanti jika diperlukan.
- Hapus pelajaran yang diarsipkan secara permanen yang tidak lagi berguna.

Untuk instruksi langkah demi langkah, lihat the section called “Pembelajaran Terus Berkelanjutan”.

Client-Side Keterampilan

Client-side keterampilan adalah kemampuan tambahan yang memperluas agen selama eksekusi transformasi. Tidak seperti definisi transformasi yang menentukan transformasi apa yang harus dilakukan, keterampilan sisi klien menyediakan alat tambahan, skrip, dan instruksi yang dapat digunakan agen bersama kemampuan bawaannya selama transformasi apa pun.

Client-side keterampilan ditemukan dari direktori tingkat proyek dan tingkat pengguna. Tempatkan keterampilan di tingkat proyek ketika mereka menegakkan standar khusus untuk repositori, dan pada tingkat pengguna ketika mereka berlaku di semua proyek Anda. Project-level keterampilan hanya tersedia ketika jalur repositori kode disediakan. Untuk konfigurasi dan penggunaan terperinci, lihatthe section called “Client-Side Keterampilan”.

Memulai

Bagian ini menjelaskan cara mengatur AWS Transform custom dan menjalankan transformasi pertama Anda.

Prasyarat

Sebelum menginstal AWS Transform custom, pastikan Anda memiliki yang berikut:

Platform yang Didukung

AWS Transform custom mendukung sistem operasi berikut:

- Linux - Dukungan penuh untuk semua distribusi Linux
- macOS - Dukungan penuh untuk sistem macOS
- Windows - Dukungan penuh untuk Windows asli menggunakan Terminal Windows dan PowerShell (`atx ct`Perintah belum didukung pada Windows)
- Windows Subsystem for Linux (WSL) - Didukung saat berjalan di bawah WSL

Perangkat Lunak yang Diperlukan

- Node.js 22 atau lebih baru - Unduh dari https://nodejs.org/en/download
- Git - Harus diinstal dan direktori kerja harus berupa repo Git yang valid untuk menjalankan transformasi. Jalankan `git init; git add .; git commit -m "Initial commit"` untuk menginisialisasi repo Git di direktori kerja Anda.

Persyaratan Jaringan

Koneksi internet dengan akses ke titik akhir berikut diperlukan:

- `transform-cli.awsstatic.com`
- `transform-custom.<region>.api.aws`
- `*.s3.amazonaws.com`

Jika Anda bekerja di lingkungan yang dibatasi internet, perbarui aturan firewall untuk mengizinkan daftar URL ini.

Menginstal AWS Transformasi CLI

Metode instalasi yang disarankan adalah menggunakan skrip instalasi. Pilih tab untuk sistem operasi Anda.

Untuk menginstal AWS Transform CLI

1. Jalankan skrip instalasi:

   Linux and macOS

   ```
   curl -fsSL https://transform-cli.awsstatic.com/install.sh | bash
   ```

   Windows (PowerShell)

   Pada Windows asli, jalankan skrip instalasi dari Terminal Windows menggunakan PowerShell.

   ```
   irm https://transform-cli.awsstatic.com/install.ps1 | iex
   ```

2. Verifikasi instalasi:

   ```
   atx --version
   ```

   Perintah harus menampilkan versi CLI AWS Transform yang diinstal.

Mengonfigurasi Autentikasi

AWS Transform custom memerlukan AWS kredensyal untuk mengautentikasi dengan layanan. Konfigurasikan otentikasi menggunakan salah satu metode berikut.

Variabel lingkungan

Mengatur variabel lingkungan berikut.

Linux and macOS

```
export AWS_ACCESS_KEY_ID=your_access_key
export AWS_SECRET_ACCESS_KEY=your_secret_key
export AWS_SESSION_TOKEN=your_session_token
```

Windows (PowerShell)

```
$env:AWS_ACCESS_KEY_ID="your_access_key"
$env:AWS_SECRET_ACCESS_KEY="your_secret_key"
```

```
$env:AWS_SESSION_TOKEN="your_session_token"
```

Anda juga dapat menentukan profil menggunakan variabel `AWS_PROFILE` lingkungan.

Linux and macOS

```
export AWS_PROFILE=your_profile_name
```

Windows (PowerShell)

```
$env:AWS_PROFILE="your_profile_name"
```

> **Note**
>
> Variabel lingkungan yang ditetapkan `$env:` dengan PowerShell berlaku untuk sesi saat ini saja. Untuk mempertahankannya di seluruh sesi, atur melalui System Properties > Environment Variables, atau gunakan`[System.Environment]::SetEnvironmentVariable()`.

AWS Berkas Kredensial

Konfigurasikan kredensyal dalam file AWS kredensyal, yang terletak di `~/.aws/credentials` Linux dan macOS, atau di Windows: `%USERPROFILE%\.aws\credentials`

```
[default]
aws_access_key_id = your_access_key
aws_secret_access_key = your_secret_key
```

Izin IAM

AWS Kredensyal Anda harus memiliki izin untuk memanggil layanan kustom AWS Transform. Sebaiknya lampirkan kebijakan `AWSTransformCustomFullAccess` AWS terkelola ke pengguna atau peran IAM Anda. Kebijakan ini menyediakan akses penuh ke kustom AWS Transform, termasuk izin untuk membuat peran terkait layanan yang diperlukan untuk emisi CloudWatch metrik ke akun Anda.

Untuk kontrol yang lebih terperinci, AWS Transform custom juga menyediakan kebijakan AWS terkelola berikut:

- `AWSTransformCustomExecuteTransformations`— Menyediakan akses untuk mengeksekusi transformasi.
- `AWSTransformCustomManageTransformations`— Menyediakan akses untuk membuat, memperbarui, membaca, dan menghapus sumber daya transformasi, serta menjalankan transformasi.

Untuk informasi selengkapnya tentang kebijakan ini, lihat kebijakan AWS terkelola untuk AWS Transform. Untuk kebijakan IAM kustom dengan izin tingkat sumber daya, lihat Panduan Referensi Otorisasi Layanan IAM Kustom AWS Transform.

> **Note**
>
> - Definisi transformasi adalah AWS sumber daya dengan ARN (Amazon Resource Names). Anda dapat menggunakan kebijakan IAM untuk mengontrol akses ke transformasi atau kelompok transformasi tertentu dengan menentukan ARN transformasi dalam laporan sumber daya kebijakan IAM Anda. Tag dapat digunakan untuk kontrol akses yang dikelompokkan.
> - AWS IAM Identity Center diperlukan untuk mengakses AWS Transform, tetapi tidak diperlukan untuk menggunakan CLI.

Melakukan konfigurasi AWS Region

AWS Kustom Transform tersedia di AWS wilayah tertentu dan secara otomatis mendeteksi konfigurasi wilayah Anda menggunakan prioritas AWS CLI standar.

Wilayah yang Didukung

AWS Transform custom tersedia di AWS wilayah berikut:

- `us-east-1`(AS Timur - Virginia Utara)
- `eu-central-1`(Eropa - Frankfurt am Main)
- `eu-west-2`(Eropa - London)

- `ca-central-1`(Kanada - Tengah)
- `ap-northeast-1`(Asia Pasifik - Tokyo)
- `ap-northeast-2`(Asia Pasifik - Seoul)
- `ap-southeast-2`(Asia Pasifik - Sydney)
- `ap-south-1`(Asia Pasifik - Mumbai)

Bagaimana Wilayah Ditentukan

CLI memeriksa sumber-sumber ini dalam urutan prioritas untuk menentukan wilayah mana yang akan digunakan:

1. `AWS_REGION`variabel lingkungan (prioritas tertinggi)
2. Variabel lingkungan `AWS_DEFAULT_REGION`
3. Profil yang dipilih dalam file AWS konfigurasi (melalui variabel `AWS_PROFILE` lingkungan)
4. Profil default dalam file AWS konfigurasi
5. Fallback default ke `us-east-1` jika tidak ada konfigurasi yang ditemukan

> **Note**
>
> File AWS konfigurasi terletak `~/.aws/config` di Linux dan macOS, `%USERPROFILE%\.aws\config` atau di Windows.

> **Note**
>
> Jika `ATX_CUSTOM_ENDPOINT` disetel, wilayah diekstraksi dari URL titik akhir dan mengganti semua pengaturan lainnya.

Mengatur Wilayah Anda

Pilih salah satu metode ini untuk mengonfigurasi wilayah Anda:

Opsi 1: Variabel lingkungan (direkomendasikan untuk penggunaan sementara):

Linux and macOS

```
export AWS_REGION=<your-region>
```

Windows (PowerShell)

```
$env:AWS_REGION="<your-region>"
```

Opsi 2: Konfigurasi AWS CLI (direkomendasikan untuk pengaturan permanen):

```
aws configure set region <your-region>
```

Opsi 3: Profile-specific konfigurasi:

Linux and macOS

```
aws configure set region <your-region> --profile your_profile_name
export AWS_PROFILE=your_profile_name
```

Windows (PowerShell)

```
aws configure set region <your-region> --profile your_profile_name
$env:AWS_PROFILE="your_profile_name"
```

Opsi 4: Pengeditan file langsung:

Edit file AWS konfigurasi secara langsung (`~/.aws/config`di Linux dan macOS, `%USERPROFILE%\.aws\config` atau di Windows):

```
[default]
region = <your-region>

[profile your_profile_name]
region = <your-region>
```

Memverifikasi Konfigurasi Wilayah Anda

Untuk memeriksa wilayah mana yang akan digunakan kustom AWS Transform:

Periksa pengaturan wilayah saat ini:

```
aws configure get region
```

Periksa variabel lingkungan:

Linux and macOS

```
echo "AWS_REGION: $AWS_REGION"
echo "AWS_DEFAULT_REGION: $AWS_DEFAULT_REGION"
echo "AWS_PROFILE: $AWS_PROFILE"
```

Windows (PowerShell)

```
echo "AWS_REGION: $env:AWS_REGION"
echo "AWS_DEFAULT_REGION: $env:AWS_DEFAULT_REGION"
echo "AWS_PROFILE: $env:AWS_PROFILE"
```

Lihat resolusi wilayah terperinci di log debug:

Linux and macOS

```
tail -f ~/.aws/atx/logs/debug.log
```

Windows (PowerShell)

```
Get-Content "$env:USERPROFILE\.aws\atx\logs\debug.log" -Wait -Tail 10
```

Contoh keluaran log yang menunjukkan sumber wilayah:

```
2026-01-07 22:34:28 [DEBUG]: Initializing FrontendServiceClient with config:
{
  "endpoint": "https://transform-custom.us-east-1.api.aws",
  "region": "us-east-1",
  "regionSource": "AWS_REGION"
}
```

`regionSource`Bidang menunjukkan dari mana wilayah itu berasal (misalnya, “aws-config (profile: default)”, “AWS_REGION”, “default”).

> **Important**
>
> Jika konfigurasi wilayah Anda menunjuk ke wilayah yang tidak didukung, CLI akan menampilkan pesan kesalahan yang jelas dengan instruksi untuk memperbarui konfigurasi Anda ke wilayah yang didukung.

Menjalankan Transformasi Pertama Anda

Cara tercepat untuk memulai adalah dengan menggunakan transformasi yang AWS dikelola. AWS Transformasi terkelola adalah transformasi pra-dibuat dan AWS diperiksa yang siap digunakan tanpa pengaturan apa pun.

Untuk menjalankan transformasi yang AWS dikelola secara interaktif:

```
atx
```

Kemudian minta agen untuk melakukan transformasi:

```
Execute the AWS/java-aws-sdk-v1-to-v2 transformation on the codebase at ./my-java-
project
```

Untuk menjalankan transformasi secara non-interaktif:

```
atx custom def exec -n AWS/java-aws-sdk-v1-to-v2 -p ./my-java-project -c "mvn clean
 install" -x -t
```

Untuk informasi rinci tentang mode eksekusi, opsi konfigurasi, dan flag perintah, lihatthe section called “Referensi Perintah”.

Memahami Hasil Transformasi

Ketika transformasi selesai, AWS Transform custom menyediakan laporan yang menjelaskan hasilnya. Jika transformasi tidak divalidasi (misalnya build yang berhasil untuk aplikasi Java), laporan memberikan rekomendasi untuk tindakan lebih lanjut.

Sebagian besar transformasi dilakukan dalam beberapa langkah, menggunakan Git untuk melakukan yang perantara. Anda dapat meninjau perubahan yang dibuat oleh transformasi menggunakan perintah Git standar:

```
git status
git log
git diff <original commit-id>
```

Membuat Transformasi Kustom

Ketika transformasi AWS-managed tidak memenuhi kebutuhan Anda, Anda dapat membuat transformasi kustom yang disesuaikan dengan kebutuhan spesifik Anda.

Untuk membuat transformasi kustom

1. Mulai AWS Transform CLI dalam mode interaktif:

   ```
   atx
   ```

2. Beri tahu agen bahwa Anda ingin membuat transformasi baru dan jelaskan kebutuhan Anda.
3. Menyediakan bahan referensi seperti dokumentasi dan sampel kode untuk meningkatkan kualitas.
4. Uji dan perbaiki transformasi secara iteratif.
5. Publikasikan ke registri transformasi organisasi Anda.

Untuk petunjuk rinci tentang membuat transformasi kustom, lihatthe section called “Alur kerja”.

AWS Transform Aplikasi Web (Opsional)

Aplikasi web AWS Transform adalah antarmuka opsional untuk memantau kampanye transformasi skala besar di beberapa repositori. Sebelum menggunakan aplikasi web AWS Transform, Anda harus mengaktifkan identitas pengguna untuk mengakses AWS Transform di organisasi Anda. Untuk informasi tentang mengaktifkan AWS Transform, lihat Menyiapkan AWS Transform.

Untuk menggunakan aplikasi web AWS Transform:

1. Kunjungi https://aws.amazon.com/transform/ dan masuk menggunakan kredensyal AWS IAM Identity Center.
2. Pilih jenis pekerjaan “Custom/code” di aplikasi web AWS Transform.
3. Beri tahu agen nama definisi transformasi yang ingin Anda gunakan.
4. Aplikasi web menyediakan perintah AWS Transform CLI yang dapat dibagikan dengan tim Anda. Perintah ini memanggil definisi transformasi dan mencatat hasil eksekusi ke aplikasi web.

5. Setelah hasil transformasi ditangkap, lihat statistik, pengukuran penghematan waktu, dan ajukan pertanyaan tentang kemajuan pekerjaan Anda di Dasbor.

Aplikasi web dirancang untuk operasi skala perusahaan di mana Anda memerlukan pemantauan transformasi terpusat di beberapa basis kode.

> **Note**
>
> AWS IAM Identity Center diperlukan untuk mengakses aplikasi web AWS Transform. CLI tidak memerlukan Pusat Identitas IAM - hanya AWS kredensyal standar yang diperlukan.

Pengantar perintah transformasi khusus

Berikut adalah beberapa perintah yang dapat Anda gunakan dengan transformasi khusus. Daftar lengkap perintah ada di referensi perintah AWS Transform custom transformations.

- `atx custom`
  - Menjalankan pengalaman interaktif transformasi kustom, memungkinkan pembuatan, penemuan, eksekusi, dan penyempurnaan transformasi.
  - Ini adalah perintah default untuk`atx`.
  - `--trust-all-tools`(`-t`) bersifat opsional dan secara implisit memungkinkan semua permintaan alat yang diminta oleh agen. Gunakan dengan hati-hati terutama di lingkungan produksi. Anda dapat mengonfigurasi kepercayaan alat untuk alat dan perintah tertentu menggunakan file pengaturan kepercayaan.
- `atx custom --help` | `atx custom -h`
  - Menampilkan menu bantuan.
  - Setiap perintah juga menyertakan menu bantuan, misalnya,`atx custom def exec --help`.
- `atx --version` | `atx -v`
  - Menunjukkan versi.
  - Nomor versi berubah dengan setiap rilis.
- `atx custom def list`
  - Mencetak daftar transformasi yang tersedia di registri transformasi.
- `atx custom def exec`

  - Mengeksekusi transformasi
- `atx custom def learnings`
  - Membuka sesi interaktif untuk melihat dan mengelola pelajaran yang telah dipelajari transformasi dari proses sebelumnya
- `atx mcp`
  - Digunakan untuk mengelola konfigurasi server MCP

Alur kerja

Melaksanakan Transformasi

Bagian ini menjelaskan berbagai cara untuk mengeksekusi transformasi dan opsi untuk mengendalikan perilaku eksekusi.

Mode Eksekusi

AWS Transform custom mendukung tiga mode eksekusi untuk mengakomodasi alur kerja yang berbeda.

Mode Percakapan Interaktif

Mulai CLI dengan `atx` dan minta agen untuk melakukan transformasi melalui bahasa alami. Mode ini memungkinkan Anda untuk melakukan percakapan penuh dengan agen, mengganggu eksekusi kapan saja, dan memberikan umpan balik selama proses transformasi.

Gunakan mode ini ketika Anda menginginkan kontrol maksimum dan kemampuan untuk memandu agen melalui skenario yang kompleks.

Eksekusi Interaktif Langsung

Gunakan `atx custom def exec -n <transformation-name> -p <path>` untuk memulai transformasi tertentu secara interaktif. Mode ini memungkinkan Anda untuk meninjau dan berinteraksi dengan agen di awal, selama, atau di akhir eksekusi. Agen akan berhenti di titik keputusan utama dan meminta masukan Anda.

Ini sangat ideal untuk menguji dan menyempurnakan transformasi sebelum menjalankannya secara mandiri.

Anda dapat menjalankan transformasi dalam mode non-interaktif atau mode tanpa kepala. Non-interactive mode menekan prompt selama transformasi bernama. Mode tanpa kepala memungkinkan Anda menjalankan agen dengan prompt teks biasa, melewati antarmuka interaktif sepenuhnya.

Non-interactive modus

Gunakan `atx custom def exec -n <transformation-name> -p <path> -x -t` untuk otomatisasi penuh. Tambahkan `-x` untuk menjalankan dalam mode non-interaktif, dan `-t` untuk mempercayai semua alat secara otomatis tanpa meminta.

Mode ini dirancang untuk integrasi CI/CD pipa dan eksekusi massal di mana tidak ada intervensi manusia yang tersedia atau diinginkan.

Mode tanpa kepala

Untuk menyelesaikan tugas tanpa berinteraksi dengan agen, jalankan `atx -x "<prompt>" -t` dan berikan instruksi Anda dalam teks biasa.

Eksekusi transformasi tanpa kepala

Gunakan mode ini untuk menerapkan definisi transformasi yang ada ke basis kode Anda. Transformasi berjalan setiap langkah secara otomatis tanpa memerlukan persetujuan Anda.

```
atx -x "apply transformation definition <transformation_definition_name> to
 <codebase_path>" -t
```

Pengembangan transformasi tanpa kepala

Membuat atau memodifikasi definisi transformasi.

Untuk mengonversi definisi transformasi lama ke format keterampilan baru (SKILL.md + referensi/), jalankan perintah berikut:

```
atx -x "convert <legacy_transformation_definition_name> transformation definition to
 skill and save as draft" -t
```

Untuk membuat definisi transformasi baru, jalankan perintah berikut:

```
atx -x "create a transformation definition to <description> with references docs
 <reference_docs_path>" -t
```

Bendera Perintah Umum

Saat menjalankan transformasi dengan`atx custom def exec`, flag berikut biasanya digunakan:

- `-n`atau `--transformation-name` - Menentukan nama transformasi untuk mengeksekusi
- `-p`atau `--code-repository-path` - Menentukan jalur ke basis kode Anda (gunakan “.” untuk direktori saat ini)
- `-c`atau `--build-command` - Menentukan perintah build atau validasi untuk menjalankan
- `-x`atau `--non-interactive` - Mengaktifkan mode non-interaktif (tidak ada permintaan pengguna)
- `-t`atau `--trust-all-tools` - Secara otomatis mempercayai semua alat tanpa meminta
- `-d`atau `--do-not-learn` - Mencegah ekstraksi pelajaran dari eksekusi ini
- `--tv`atau `--transformation-version` - Menentukan versi tertentu dari transformasi
- `-g`atau `--configuration` - Menyediakan file konfigurasi atau konfigurasi sebaris

> **Important**
>
> `--trust-all-tools`Bendera `-t` atau secara otomatis menyetujui semua eksekusi alat tanpa meminta dan melewati sebagian besar pagar pembatas keamanan, (perintah yang cocok dengan `alwaysPromptCommands` daftar Anda masih memerlukan izin eksplisit kecuali diganti oleh). `trustedShellCommands` Melewati `--non-interactive` dan `--trust-all-tools` diperlukan untuk pengalaman yang sepenuhnya otonom tetapi tidak diperlukan untuk melaksanakan transformasi. Gunakan dengan hati-hati di lingkungan produksi.

Menggunakan File Konfigurasi

AWS Transform custom mendukung file konfigurasi opsional dalam format YAMG atau JSON. File konfigurasi memungkinkan Anda menentukan parameter eksekusi dan memberikan konteks tambahan kepada agen.

Untuk menggunakan file konfigurasi:

```
atx custom def exec --configuration file://config.yaml
```

Anda juga dapat memberikan konfigurasi sebagai pasangan nilai kunci inline:

```
atx custom def exec --configuration "key=value,key2=value2"
```

Contoh file konfigurasi (config.yaml):

```
codeRepositoryPath: ./my-project
transformationName: my-transformation
buildCommand: mvn clean install
additionalPlanContext: |
  The target Java version to upgrade to is Java 17.
  Ensure compatibility with our internal logging framework version 2.3.
validationCommands: |
  mvn test
  mvn verify
```

`additionalPlanContext`Parameter memberikan konteks tambahan untuk rencana eksekusi agen. Ini sangat berguna dengan transformasi AWS yang dikelola untuk menyesuaikan perilaku mereka untuk kebutuhan spesifik Anda.

Perintah Build dan Validasi

Perintah build atau validasi adalah parameter opsional yang menentukan cara memvalidasi kode Anda selama proses transformasi. AWS Transform custom akan mencoba menyimpulkan perintah build terbaik berdasarkan transformasi jika tidak ditentukan, meskipun disarankan untuk spesifik untuk kualitas.

Contoh perintah build dan validasi:

- Java: `mvn clean install` atau `gradle build`
- Python: atau `pytest` `python -m py_compile`
- Node.js: `npm run build` atau `npm test`
- Linters: `eslint .` atau `pylint .`

Bahkan untuk bahasa atau transformasi yang tidak memerlukan pembangunan, memberikan perintah yang memvalidasi hasil dan mengembalikan masalah jika validasi gagal sangat penting untuk meningkatkan kualitas transformasi.

Jika tidak diperlukan build atau validasi, hilangkan input Anda.

Mengontrol Perilaku Belajar

Secara default, AWS Transform custom mengekstrak pelajaran dari setiap eksekusi transformasi. Anda dapat mencegah pembelajaran untuk eksekusi tertentu.

Untuk mencegah pembelajaran dari eksekusi:

```
atx custom def exec -n my-transformation -p ./my-project -d
```

`--do-not-learn`Bendera `-d` atau memilih untuk tidak mengizinkan ekstraksi pelajaran dari eksekusi saat ini.

Melanjutkan Percakapan

AWS Transform custom memungkinkan Anda untuk melanjutkan percakapan sebelumnya dalam 30 hari pembuatan.

Untuk melanjutkan percakapan terbaru:

```
atx --resume
```

Untuk melanjutkan percakapan tertentu:

```
atx --conversation-id <conversation-id>
```

> **Important**
>
> Percakapan hanya dapat dilanjutkan dalam waktu 30 hari setelah pembuatan. Setelah 30 hari, percakapan tidak dapat dilanjutkan lagi.

Menit Agen Pelacakan

AWS Transform custom track menit agen yang dikonsumsi selama sesi transformasi. Menit agen terakumulasi sepanjang siklus hidup percakapan dan ditampilkan saat percakapan berakhir:

```
Agent minutes used: 12.50
```

Menit agen bertahan di seluruh interupsi. Jika Anda menyela sesi dengan Ctrl+C dan melanjutkannya nanti, menit yang sebelumnya terakumulasi terbawa dan terus terakumulasi dalam sesi yang dilanjutkan.

Untuk memeriksa Risalah Agen selama sesi interaktif:

Ketik `/usage` pada prompt input untuk menampilkan Menit Agen yang terakumulasi saat ini tanpa mengakhiri percakapan.

Untuk menetapkan batas anggaran Risalah Agen:

```
atx custom def exec -n my-transformation -p ./my-project --limit 30
```

`--limit`Opsi menetapkan anggaran Menit Agen maksimum untuk sesi tersebut. Agent Minutes mencerminkan waktu kerja agen aktif, bukan waktu jam dinding. Ketika batas tercapai, CLI menampilkan pesan dan keluar dengan instruksi untuk melanjutkan:

```
## Budget limit reached: 30.00 / 30.00 Agent Minutes. Exiting.
```

Anda dapat melanjutkan percakapan nanti dengan batas yang meningkat:

```
atx --conversation-id <conversation_id> -t --limit <increased_limit>
```

Pembelajaran Terus Berkelanjutan

Bagian ini menjelaskan cara meninjau dan mengelola pelajaran yang dibuat oleh pembelajaran berkelanjutan.

Memahami Pelajaran

Sistem pembelajaran berkelanjutan secara otomatis mengekstrak pelajaran dari transformasi sebelumnya. Sistem membuatnya secara asinkron berdasarkan:

- Umpan balik pengembang disediakan dalam mode interaktif
- Masalah kode yang dihadapi selama transformasi

Pelajaran terakumulasi dari waktu ke waktu saat Anda menjalankan transformasi di berbagai basis kode. Sistem menerapkannya secara otomatis untuk meningkatkan future run. Setiap pelajaran

termasuk dalam kategori yang berisi semua pelajaran dari domain yang sama sehingga Anda dapat meninjau pelajaran terkait bersama-sama. Untuk pelajaran yang tidak ingin Anda gunakan, Anda dapat mengarsipkan atau menghapus pelajaran sepenuhnya.

Melihat dan Mengelola Pelajaran

Gunakan `learnings` perintah untuk membuka sesi interaktif untuk menjelajah dan mengelola pelajaran definisi transformasi.

Untuk membuka penampil pelajaran:

```
atx custom def learnings -n my-transformation
```

Penampil membuka daftar kategori pelajaran, masing-masing menunjukkan berapa banyak pelajaran aktif yang dikandungnya. Pilih kategori untuk melihat pelajarannya, lalu pilih pelajaran untuk melihat detail lengkapnya, termasuk badan pelajaran, dampaknya, dan berapa banyak proses sebelumnya yang dikonsultasikan.

Pengarsipan dan Pemulihan Pelajaran

Sistem menerapkan pelajaran secara otomatis. Jika Anda tidak ingin sistem menerapkan pelajaran, Anda dapat mengarsipkannya. Sistem mempertahankan pelajaran yang diarsipkan tetapi tidak menerapkannya ke masa depan. Semua pelajaran yang diarsipkan dikelompokkan bersama sehingga Anda dapat meninjaunya dan mengembalikan apa pun ke penggunaan aktif.

Menghapus Pelajaran

Hapus pelajaran yang tidak berguna secara permanen. Penghapusan tidak dapat dibatalkan, dan sistem mungkin mempelajari kembali pelajaran yang dihapus dari future run.

Pelajaran harus diarsipkan sebelum dapat dihapus.

Konfigurasi Lanjutan

Bagian ini menjelaskan fitur lanjutan dan opsi konfigurasi untuk AWS mengubah kustom.

Variabel lingkungan

Anda dapat menyesuaikan perilaku CLI menggunakan variabel lingkungan.

> **Note**
>
> Contoh berikut menunjukkan sintaks Linux dan macOS (). `export` Di Windows, atur variabel lingkungan dalam PowerShell menggunakan`$env:NAME="value"`. Lihat tab Windows (PowerShell) untuk perintah yang setara.

ATX_SHELL_TIMEOUT

Ganti batas waktu default untuk perintah shell (900 seconds/15 menit).

Linux and macOS

```
export ATX_SHELL_TIMEOUT=1800  # 30 minutes
```

Windows (PowerShell)

```
$env:ATX_SHELL_TIMEOUT=1800  # 30 minutes
```

Ini berguna untuk basis kode besar atau proses build yang berjalan lama.

ATX_DISABLE_UPDATE_CHECK

Nonaktifkan pemeriksaan versi otomatis dan perbarui pemberitahuan selama eksekusi perintah.

Linux and macOS

```
export ATX_DISABLE_UPDATE_CHECK=true
```

Windows (PowerShell)

```
$env:ATX_DISABLE_UPDATE_CHECK="true"
```

ATX_GIT_COMMITTER_NAME dan ATX_GIT_COMMITTER_EMAIL

Konfigurasikan identitas penulis yang digunakan untuk komit pos pemeriksaan yang dibuat kustom AWS Transform di repositori Anda saat menerapkan perubahan selama transformasi. Ketika variabel-

variabel ini tidak disetel, komit pos pemeriksaan dikaitkan dengan identitas default ()`ATX Bot <checkpoint@atx.bot>`. Tetapkan kedua variabel untuk mengaitkan pos pemeriksaan ke penulis tertentu.

Linux and macOS

```
export ATX_GIT_COMMITTER_NAME="Jane Developer"
export ATX_GIT_COMMITTER_EMAIL="jane@example.com"
```

Windows (PowerShell)

```
$env:ATX_GIT_COMMITTER_NAME="Jane Developer"
$env:ATX_GIT_COMMITTER_EMAIL="jane@example.com"
```

Pengaturan Kepercayaan

Pengaturan kepercayaan memungkinkan Anda untuk menyetujui alat dan perintah tertentu untuk dijalankan tanpa permintaan. Anda juga dapat meminta izin eksplisit untuk perintah shell tertentu terlepas dari tingkat kepercayaan. Pengaturan ini dikonfigurasi dalam `~/.aws/atx/trust-settings.yaml` file.

File berisi tiga daftar:

- `trustedTools`- Alat yang dapat mengeksekusi tanpa meminta
- `trustedShellCommands`- Perintah Shell yang dapat dijalankan tanpa meminta
- `alwaysPromptCommands`- Pola perintah Shell yang memerlukan izin eksplisit kecuali diganti oleh`trustedShellCommands`, terlepas dari `-t` bendera atau kepercayaan sesi. Pola-pola ini tidak ditegakkan dalam mode non-interaktif ()`-x`.

Alat tepercaya default:

- `file_read`
- `get_transformation_from_registry`
- `list_available_transformations_from_registry`

Mengedit pengaturan kepercayaan:

Anda dapat mengedit file trust-settings.yaml secara manual untuk menambah atau menghapus alat dan perintah tepercaya. Keduanya `trustedShellCommands` dan `alwaysPromptCommands` mendukung pola wildcard glob menggunakan. *

> **Note**
>
> Jika perintah cocok dengan kedua daftar, `trustedShellCommands` ambil prioritas.

Berikut ini menjelaskan setiap daftar perintah dan memberikan contoh:

- `trustedShellCommands`- Perintah yang cocok dengan pola-pola ini dijalankan tanpa meminta, melewati semua pagar pembatas lainnya. Pola dicocokkan dengan string perintah penuh.

  Contoh:

  - `cd *`- Cocokkan perintah majemuk dimulai dengan cd
  - `*&&*`- Mempercayai semua perintah dengan && operator
- `alwaysPromptCommands`- Perintah yang cocok dengan pola ini memerlukan izin eksplisit kecuali diganti oleh`trustedShellCommands`, terlepas dari `-t` bendera atau kepercayaan sesi. Pola-pola ini tidak ditegakkan dalam mode non-interaktif ()`-x`. Pola dicocokkan dengan setiap sub-perintah dalam ekspresi majemuk (&&, | |, substitusi perintah).

  Contoh:

  - `rm -rf *`- Selalu meminta perintah hapus paksa rekursif
  - `sudo *`- Selalu meminta perintah yang dijalankan dengan sudo
  - `find * -exec *`- Selalu meminta perintah find dengan -exec

Session-level kepercayaan:

Selama permintaan interaktif, Anda dapat memilih:

- `(y)es`- Jalankan sekali
- `(n)o`- Menyangkal
- `(t)rust`- Percaya untuk sesi saat ini saja

Session-level pengaturan trust bersifat sementara dan diatur ulang saat CLI dimulai ulang, memberikan persetujuan sementara tanpa memodifikasi trust-settings.yaml secara permanen.

> **Note**
>
> Kepercayaan sesi tidak tersedia untuk perintah yang cocok dengan `alwaysPromptCommands` daftar Anda.

Server Protokol Konteks Model (MCP)

AWS Transform CLI mendukung server Model Context Protocol (MCP), yang memperluas fungsinya dengan alat tambahan.

Konfigurasi:

Konfigurasikan server MCP dalam `~/.aws/atx/mcp.json` file. AWS Transform CLI mendukung dua jenis server MCP: server berbasis perintah lokal dan server HTTP jarak jauh.

Server berbasis perintah lokal:

Server lokal berjalan sebagai proses anak di mesin Anda. Konfigurasikan mereka dengan `command` properti:

```
{
  "mcpServers": {
    "my-local-server": {
      "command": "npx",
      "args": ["-y", "@example/mcp-server"]
    }
  }
}
```

Server HTTP jarak jauh:

Server jarak jauh terhubung ke server MCP yang dihosting di URL HTTP atau HTTPS. Konfigurasikan mereka dengan `url` properti:

```
{
  "mcpServers": {
```

```
    "my-remote-server": {
      "url": "https://api.example.com/mcp",
      "headers": {
        "Authorization": "Bearer ${MCP_API_TOKEN}"
      }
    }
  }
}
```

`headers`Properti ini opsional dan mendukung ekspansi variabel lingkungan menggunakan `${VAR_NAME}` sintaks. Ini memungkinkan Anda untuk menyimpan nilai sensitif seperti token API dalam variabel lingkungan daripada di file konfigurasi.

Properti konfigurasi:

Server berbasis perintah lokal mendukung properti berikut:

- `command`(wajib) - Perintah untuk menjalankan server
- `args`(opsional) - Array argumen baris perintah
- `env`(opsional) - Variabel lingkungan untuk diteruskan ke proses server

Server HTTP jarak jauh mendukung properti berikut:

- `url`(wajib) - URL HTTP atau HTTPS dari server MCP jarak jauh
- `headers`(opsional) - Header HTTP untuk disertakan dalam permintaan, dengan dukungan untuk ekspansi variabel `${VAR_NAME}` lingkungan

Mengelola server MCP:

Lihat daftar server MCP yang dikonfigurasi:

```
atx mcp tools
```

Daftar alat yang tersedia yang ditawarkan oleh server MCP tertentu:

```
atx mcp tools --server <server-name>
```

Pelacakan penggunaan:

CLI secara otomatis melacak penggunaan alat MCP selama eksekusi transformasi. Statistik penggunaan dipertahankan seperti `mcp_usage.json` di direktori percakapan di sampingmetadata.json. File merekam metrik per-alat untuk setiap eksekusi, termasuk:

- Jumlah pemanggilan per alat
- Jumlah kesalahan per alat
- Total waktu eksekusi per alat
- Detail kesalahan terakhir (jika ada)

Client-Side Keterampilan

Client-side keterampilan adalah kemampuan tambahan yang memperluas agen selama eksekusi transformasi. Mereka memungkinkan Anda untuk menyediakan alat khusus, skrip, dan instruksi yang dapat digunakan agen bersama kemampuan bawaannya.

Direktori penemuan keterampilan:

Keterampilan ditemukan dari empat direktori dalam urutan prioritas. Jika keterampilan dengan nama yang sama ada di beberapa direktori, direktori pertama dalam daftar diprioritaskan:

1. `<project>/.aws/atx/skills/`- Project-level, AWS Transformasi CLI-specific
2. `<project>/.agents/skills/`- Project-level, cross-client (tersedia untuk perkakas agen yang kompatibel)
3. `~/.aws/atx/skills/`- User-level, AWS Transformasi CLI-specific
4. `~/.agents/skills/`- User-level, cross-client (tersedia untuk perkakas agen yang kompatibel)

`.aws/atx/skills/`Direktori khusus untuk AWS Transform CLI. `.agents/skills/`Direktori adalah cross-client, yang berarti keterampilan yang ditempatkan di sana tersedia untuk perkakas agen yang kompatibel di luar Transform AWS CLI.

Struktur direktori keterampilan:

Setiap skill adalah direktori yang berisi `SKILL.md` file dengan frontmatter YAMM:

```
~/.aws/atx/skills/
### my-skill/
    ### SKILL.md          # Required: frontmatter + instructions
    ### references/       # Optional: reference docs the agent can read
```

```
    #   ### guide.md
    ### scripts/          # Optional: scripts the agent can execute
        ### validate.py
```

SKILL.md format:

```
---
name: my-skill
description: When to use this skill
---
# Skill Title

Instructions for the agent...
```

`name`Bidang harus cocok dengan nama direktori induk.

Menonaktifkan keterampilan:

Untuk mencegah keterampilan dimuat tanpa menghapus file-nya, tambahkan `disable-model-invocation: true` ke frontmatter:

```
---
name: my-skill
description: When to use this skill
disable-model-invocation: true
---
```

Ketika properti ini diatur, CLI melewatkan keterampilan selama penemuan. Agen tidak dapat melihat atau menggunakan keterampilan kecuali definisi transformasi secara eksplisit menginstruksikannya untuk membaca file keterampilan. Gunakan ini untuk menonaktifkan sementara keterampilan, menandainya sebagai pekerjaan yang sedang berlangsung, atau menyimpan materi referensi yang ditujukan hanya untuk pembaca manusia.

> **Note**
>
> File keterampilan yang dinonaktifkan tetap ada di disk. Jika definisi transformasi menginstruksikan agen untuk membaca jalur file tertentu, agen masih dapat mengakses konten. `disable-model-invocation`Properti mencegah penemuan otomatis dan injeksi konteks, bukan akses sistem file.

Ketersediaan keterampilan dengan mode eksekusi:

- Mode Exec (`atx custom def exec`dengan`--code-repository-path`) - Menemukan keterampilan dari direktori tingkat pengguna dan tingkat proyek.
- Mode interaktif (`atx`) - Hanya keterampilan tingkat pengguna yang ditemukan pada awalnya. Saat Anda menyediakan jalur repositori kode selama sesi, keterampilan tingkat proyek juga dimuat.

Memverifikasi penemuan keterampilan:

Periksa log debug CLI setelah dijalankan untuk memverifikasi keterampilan mana yang ditemukan:

Linux and macOS

```
grep -i "skill" ~/.aws/atx/logs/debug.log | tail -20
```

Windows (PowerShell)

```
Select-String -Pattern "skill" "$env:USERPROFILE\.aws\atx\logs\debug.log" | Select-
Object -Last 20
```

Keterampilan yang gagal validasi dilewati dengan peringatan di log debug.

> **Note**
>
> Client-side keterampilan membutuhkan CLI versi 2.0 atau yang lebih baru.

Memilih Antara Project-Level dan User-Level Keterampilan

Di mana Anda menempatkan keterampilan menentukan siapa yang mendapat manfaat darinya dan kapan itu diaktifkan.

Project-level keterampilan (`<project>/.aws/atx/skills/`):

Komit ini ke kontrol versi sehingga setiap anggota tim yang menjalankan transformasi terhadap repositori secara otomatis menemukannya. Gunakan keterampilan tingkat proyek untuk:

- Repository-specific pemeriksaan kepatuhan (aturan Dockerfile, kebijakan Terraform, validator keamanan migrasi)

- Standar pengkodean organisasi yang berlaku untuk basis kode ini (pola observabilitas, penanganan kesalahan, konvensi penamaan)
- Buat atau uji skrip yang unik untuk proyek (linter khusus, fungsi kebugaran arsitektur)
- Panduan migrasi API untuk pustaka internal yang digunakan dalam repositori ini

User-level keterampilan (`~/.aws/atx/skills/`):

Ini tetap ada di mesin Anda dan diaktifkan selama semua transformasi terlepas dari repositori mana yang Anda targetkan. Gunakan keterampilan tingkat pengguna untuk:

- Alat alur kerja pribadi (generator changelog, format pesan komit)
- Cross-project preferensi (pola pengujian yang disukai, pengingat gaya dokumentasi)
- Kepatuhan lisensi memeriksa bahwa organisasi Anda memerlukan di semua repositori
- Ambang batas cakupan atau gerbang kualitas yang Anda terapkan pada setiap basis kode yang Anda gunakan

Kiat untuk keterampilan yang efektif:

- Tulis `description` bidang yang jelas di `SKILL.md` frontmatter Anda. Agen menggunakan bidang ini untuk memutuskan kapan suatu keterampilan relevan.
- Keluar dari skrip validasi dengan kode 0 pada keberhasilan dan bukan nol pada kegagalan. Agen menafsirkan kode keluar untuk menentukan kepatuhan.
- Cetak pesan kesalahan yang jelas dan dapat ditindaklanjuti dalam skrip. Agen membaca output untuk memahami apa yang harus diperbaiki.
- Tempatkan keterampilan di direktori lintas klien (`.agents/skills/`) di kedua level untuk membagikannya dengan alat pengembangan AI lainnya di luar AWS Transform CLI.

Client-Side Contoh Keterampilan

Contoh-contoh ini menunjukkan dua pola umum: keterampilan validasi berbasis skrip dan keterampilan referensi saja.

Contoh: Dockerfile Compliance Checker () Script-Based

Keterampilan ini memvalidasi Dockerfiles terhadap praktik terbaik keamanan dan operasional. Ini menggunakan skrip validasi yang dijalankan agen sebelum dan sesudah membuat perubahan.

Struktur direktori:

```
.aws/atx/skills/
### dockerfile-compliance/
    ### SKILL.md
    ### scripts/
    #   ### lint_dockerfile.sh
    ### references/
        ### dockerfile-best-practices.md
```

SKILL.md:

```
---
name: dockerfile-compliance
description: Validates Dockerfiles against security and operational best practices
---
# Dockerfile Compliance Checker

When a transformation creates or modifies Dockerfiles, run the compliance checker.

## When to use

- After creating a new Dockerfile
- After modifying FROM, RUN, USER, or EXPOSE directives
- When containerizing an application as part of a transformation

## How to use

Run: `bash scripts/lint_dockerfile.sh <path-to-Dockerfile>`

If violations are found, consult `references/dockerfile-best-practices.md`
for compliant patterns.
```

Skrip validasi memeriksa tag gambar dasar yang tidak disematkan, berjalan sebagai root, rahasia hardcode dalam `ENV` arahan, dan definisi yang hilang. `HEALTHCHECK` Agen menjalankan skrip, memperbaiki pelanggaran menggunakan pola dari file referensi, dan menjalankan kembali skrip untuk mengonfirmasi kepatuhan.

Contoh: API Deprecation Helper () Reference-Only

Keterampilan ini memandu agen untuk mengganti panggilan API yang tidak digunakan lagi selama transformasi pemutakhiran. Ini hanya menggunakan file referensi tanpa skrip.

Struktur direktori:

```
.aws/atx/skills/
### api-deprecation-helper/
    ### SKILL.md
    ### references/
        ### aws-sdk-v2-to-v3.md
        ### react-class-to-hooks.md
```

SKILL.md:

```
---
name: api-deprecation-helper
description: Guides the agent through replacing deprecated API calls with modern
 equivalents
---
# API Deprecation Helper

When performing upgrade transformations, use this skill to identify and replace
deprecated API calls with their modern equivalents.

## When to use

- During any version upgrade transformation
- When build warnings mention deprecated APIs
- When transforming code that uses legacy patterns

## Process

1. Identify deprecated API calls in the codebase
2. For each deprecated call, find the replacement in `references/`
3. Apply the replacement, preserving the original behavior
4. Verify the replacement compiles and tests pass
```

File referensi berisi contoh kode sebelum dan sesudah. Misalnya, pola `aws-sdk-v2-to-v3.md` peta seperti yang `s3.putObject(params).promise()` setara dengan modular v3 menggunakan `S3Client` dan`PutObjectCommand`.

Tag dan Organisasi

Anda dapat mengatur transformasi dengan tag untuk kontrol akses dan kategorisasi.

> **Note**
>
> Beberapa perintah ini memerlukan penentuan Nama Sumber Daya Amazon (ARN) untuk Definisi Transformasi. Struktur ARN adalah: `arn:aws:transform-custom:<region>:<account-id>:package/<td-name>`

Untuk membuat daftar tag untuk transformasi:

```
atx custom def list-tags --arn <transformation-arn>
```

Untuk menambahkan tag ke transformasi:

```
atx custom def tag --arn <transformation-arn> --tags '{"env":"prod","team":"backend"}'
```

Untuk menghapus tag dari transformasi:

```
atx custom def untag --arn <transformation-arn> --tag-keys "env,team"
```

Tag dapat digunakan untuk kontrol akses yang dikelompokkan dalam kebijakan IAM. Anda dapat membuat kebijakan yang memberikan izin untuk semua transformasi dengan tag tertentu (misalnya, semua transformasi yang ditandai dengan atau). `team:frontend` `environment:production`

Beberapa catatan

AWS Transform CLI memelihara tiga jenis log untuk pemecahan masalah dan debugging.

Log percakapan:

Linux and macOS

```
~/.aws/atx/custom/<conversation_id>/logs/<timestamp>-conversation.log
```

Windows

```
%USERPROFILE%\.aws\atx\custom\<conversation_id>\logs\<timestamp>-conversation.log
```

Log ini berisi riwayat percakapan lengkap untuk sesi tertentu.

Log subagent:

Linux and macOS

```
~/.aws/atx/custom/<conversation_id>/logs/subagents/<name>.log
```

Windows

```
%USERPROFILE%\.aws\atx\custom\<conversation_id>\logs\subagents\<name>.log
```

Log ini berisi output dari subagen yang ditimbulkan oleh agen utama selama transformasi. Anda tidak perlu mengelola subagen secara langsung.

Log debug pengembang:

Linux and macOS

```
~/.aws/atx/logs/debug*.log
~/.aws/atx/logs/error.log
```

Windows

```
%USERPROFILE%\.aws\atx\logs\debug*.log
%USERPROFILE%\.aws\atx\logs\error.log
```

Log ini memberikan informasi pemecahan masalah lanjutan untuk CLI itu sendiri.

> **Note**
>
> Mungkin ada beberapa file log debug di direktori log (yaitu debug1.log, debug2.log). Tinjau dan berikan semua log yang relevan misalnya, ~/. aws/atx/custom/ <conversation-id>/* dan ~/. aws/atx/logs/ *, saat membuka tiket dukungan untuk resolusi yang lebih cepat.

Pembaruan CLI

Perbarui CLI Anda untuk mengakses fitur dan peningkatan baru.

Untuk memeriksa pembaruan:

```
atx update --check
```

Untuk memperbarui ke versi terbaru:

```
atx update
```

Untuk memperbarui ke versi tertentu:

```
atx update --target-version <version>
```

Buat Transformasi Kustom

Bagian ini menjelaskan cara membuat, memodifikasi, dan mengelola definisi transformasi kustom.

Menciptakan Transformasi Baru

Gunakan CLI interaktif untuk membuat definisi transformasi baru.

Untuk membuat definisi transformasi

1. Mulai AWS Transform CLI:

   ```
   atx
   ```
2. Beri tahu agen bahwa Anda ingin membuat transformasi baru.
3. Berikan penjelasan yang jelas dan terperinci tentang tujuan transformasi. Termasuk:
   - Status sumber dan target (misalnya, “upgrade dari versi X ke versi Y”)
   - Perubahan khusus diperlukan (misalnya, “perbarui pernyataan impor, ganti metode yang tidak digunakan lagi”)
   - Setiap pertimbangan atau kendala khusus
4. Ketika agen meminta klarifikasi atau informasi tambahan, berikan contoh spesifik dan bahan referensi.
5. Tinjau definisi transformasi awal yang dibuat oleh agen.
6. Uji transformasi pada basis kode sampel.
7. Iterasi dengan memberikan umpan balik, perbaikan kode, atau contoh tambahan.
8. Simpan transformasi secara lokal atau publikasikan ke registri.

Praktik terbaik untuk membuat transformasi:

- Mulailah dengan transformasi yang sederhana dan terdefinisi dengan baik sebelum mencoba yang kompleks
- Menyediakan bahan referensi yang komprehensif termasuk panduan migrasi dan contoh kode
- Uji pada beberapa basis kode sampel sebelum dipublikasikan
- Gunakan perintah build atau validasi deterministik untuk mengaktifkan pembelajaran berkelanjutan
- Pertimbangkan untuk memecah transformasi kompleks menjadi beberapa langkah yang lebih kecil
- Tandai informasi penting dengan “KRITIS:” atau “PENTING:” dalam definisi transformasi Anda untuk memastikan agen memprioritaskan persyaratan ini
- Bila Anda membutuhkan persyaratan yang tepat diikuti (seperti menggunakan perintah tertentu atau nilai string), secara eksplisit menentukan string lengkap dalam definisi transformasi Anda. Anda dapat membungkusnya dalam tanda kutip bash untuk menunjukkan dengan jelas bahwa itu adalah perintah terminal atau string literal, yang mengurangi variabilitas dan memastikan eksekusi yang konsisten

Menyediakan Bahan Referensi

Anda dapat memberikan file referensi ke AWS Transform custom dengan menentukan path file selama percakapan. File-file ini disimpan dalam `references/` folder definisi transformasi.

Jenis file referensi yang direkomendasikan:

- Before/after kode contoh
- Dokumentasi untuk API, pustaka, atau fitur yang terlibat
- Human-readable panduan migrasi

Untuk menyediakan file referensi:

```
Take a look at the documentation here: /path/to/migration-guide.md
```

Anda juga dapat menyediakan direktori yang berisi beberapa file referensi:

```
Take a look at the docs we have here: /path/to/docs/
```

> **Note**
>
> Hanya file berbasis teks (.md, .html, .txt, file kode) yang didukung. File biner, gambar, dan file teks kaya (mis., .pdf, .png, .docx) saat ini tidak didukung. Seringkali mungkin untuk mengekstrak konten teks dan menggunakannya sebagai referensi. Jika Anda memiliki banyak file teks kecil, pertimbangkan untuk menggabungkannya menjadi beberapa file bernama deskriptif. Ada batas total 10MB untuk semua file.

Memodifikasi Transformasi yang Ada

Anda dapat memodifikasi transformasi kustom sebelum dan sesudah menyimpannya sebagai draf atau menerbitkannya. Anda tidak dapat memodifikasi transformasi AWS-managed. Jika Anda perlu menyesuaikannya, Anda dapat memberikan konteks tambahan menggunakan file konfigurasi.

Untuk memodifikasi transformasi yang ada

1. Mulai AWS Transform CLI:

   ```
   atx
   ```

2. Beri tahu agen bahwa Anda ingin memodifikasi transformasi yang ada.
3. Pilih apakah akan:

   - Berikan jalur file ke transformasi yang disimpan secara lokal (yaitu bukan draf yang disimpan atau dipublikasikan)
   - Minta daftar transformasi dari registri
4. Jika memilih dari registri, pilih transformasi yang ingin Anda modifikasi.
5. Bekerja dengan agen untuk menggambarkan perubahan yang ingin Anda buat.
6. Uji transformasi yang diperbarui pada basis kode sampel.
7. Publikasikan pembaruan Anda ke registri jika diinginkan.

Menerbitkan dan Mengelola Transformasi

Anda dapat mempublikasikan dan mengelola transformasi Anda menggunakan pengalaman interaktif atau dengan perintah berikut.

Untuk menyimpan transformasi sebagai konsep:

```
atx custom def save-draft -n my-transformation --description "Description of the
 transformation" --sd ./transformation-directory
```

Untuk mempublikasikan transformasi:

```
atx custom def publish -n my-transformation --description "Description of the
 transformation" --sd ./transformation-directory
```

Untuk membuat daftar transformasi yang tersedia:

```
atx custom def list
```

Untuk mengunduh definisi transformasi:

```
atx custom def get -n my-transformation
```

Ini mengunduh definisi transformasi ke direktori kerja Anda saat ini. Anda dapat menentukan direktori target dengan `--td` bendera dan versi dengan `--tv` bendera.

Untuk menghapus definisi transformasi:

```
atx custom def delete -n my-transformation
```

> **Important**
>
> Ini secara permanen menghapus definisi transformasi yang ditentukan dari akun Anda.

Mengelola Versi Transformasi

AWS Transform custom mempertahankan versi definisi transformasi Anda. Anda dapat menentukan versi saat menjalankan atau mengunduh transformasi.

Untuk menjalankan versi tertentu:

```
atx custom def exec -n my-transformation --tv v1 -p ./my-project
```

Untuk mengunduh versi tertentu:

```
atx custom def get -n my-transformation --tv v1
```

Jika tidak ada versi yang ditentukan, versi terbaru digunakan.

Referensi Perintah

Bagian ini memberikan referensi komprehensif dari semua perintah CLI kustom AWS Transform.

Perintah Interaktif

atx

Mulai percakapan interaktif dengan AWS Transform CLI.

```
atx                          # Start new conversation
atx --resume                 # Resume most recent conversation
atx --conversation-id <id>   # Resume specific conversation
atx -t                       # Start with all tools trusted
```

Perintah sesi interaktif

Perintah berikut dapat diketik pada prompt input selama sesi interaktif:

- `/usage`- Menampilkan menit agen akumulasi untuk sesi saat ini

Perintah Definisi Transformasi

atx kustom def exec

Jalankan definisi transformasi pada repositori kode.

Opsi	Bentuk Panjang	Parameter	Deskripsi
`-p`	`--code-repository-path`	`<path>`	Jalur ke repositori kode untuk diubah. Untuk direktori saat ini, gunakan “.”

Opsi	Bentuk Panjang	Parameter	Deskripsi
`-c`	`--build-command`	`<command>`	Perintah untuk dijalankan saat membangun repositori
`-n`	`--transformation-name`	`<name>`	Nama definisi transformasi dalam registri
`-x`	`--non-interactive`	`-`	Menjalankan transformasi tanpa bantuan pengguna
`-t`	`--trust-all-tools`	`-`	Mempercayai semua alat (tidak ada petunjuk alat)
`-d`	`--do-not-learn`	`-`	Menyisih untuk tidak mengizinkan ekstraksi pelajaran dari eksekusi ini
`-g`	`--configuration`	`<config>`	Jalur ke file konfigurasi (JSON atau YAMG) atau pasangan kunci=nilai
`--tv`	`--transformation-version`	`<version>`	Versi definisi transformasi yang akan digunakan

Opsi	Bentuk Panjang	Parameter	Deskripsi
`--limit`	`--limit`	`<limit>`	Tetapkan batas anggaran Risalah Agen. Transformasi keluar ketika batas tercapai dan dapat dilanjutkan dengan batas yang meningkat
`-h`	`--help`	-	Tampilkan bantuan untuk perintah

daftar def kustom atx

Daftar definisi transformasi yang tersedia.

```
atx custom def list
atx custom def list --json    # Output in JSON format
```

atx custom def dapatkan

Dapatkan detail definisi transformasi tertentu.

Opsi	Bentuk Panjang	Parameter	Deskripsi
`-n`	`--transformation-name`	`<name>`	Nama definisi transformasi untuk diambil
`--tv`	`--transformation-version`	`<version>`	Versi definisi transformasi untuk mengambil
`--td`	`--target-directory`	`<directory>`	Direktori target untuk menyimpan definisi transformasi (default: ".")

Opsi	Bentuk Panjang	Parameter	Deskripsi
`--json`	`--json`	-	Respons keluaran dalam format JSON
`-h`	`--help`	-	Tampilkan bantuan untuk perintah

hapus def kustom atx

Hapus definisi transformasi secara permanen.

```
atx custom def delete -n <transformation-name>
```

atx kustom def publikasikan

Publikasikan definisi transformasi ke registri.

Opsi	Bentuk Panjang	Parameter	Deskripsi
`-n`	`--transformation-name`	`<name>`	Nama definisi transformasi untuk dipublikasikan
`--tv`	`--transformation-version`	`<version>`	Versi definisi transformasi yang akan digunakan (tidak berlaku dengan --description atau --source-directory)
`--sd`	`--source-directory`	`<directory>`	Direktori sumber untuk membaca file definisi transformasi lokal (tidak berlaku dengan --transformation-version)

Opsi	Bentuk Panjang	Parameter	Deskripsi
`--description`	`--description`	`<description>`	Deskripsi untuk definisi transformasi (tidak berlaku dengan --transformation-version)
`--json`	`--json`	-	Respons keluaran dalam format JSON
`-h`	`--help`	-	Tampilkan bantuan untuk perintah

draf simpan def kustom atx

Simpan definisi transformasi sebagai draf di registri. Draf kedaluwarsa setelah 30 hari dan tidak muncul di registri. Perintah ini mengembalikan nomor versi draf. Anda harus mengeksekusi dan mengambil draf secara eksplisit menggunakan nama transformasi dan nomor versi.

```
atx custom def save-draft -n <transformation-name> --description "Description" --sd
 <directory>
```

Perintah Pelajaran

pembelajaran def kustom atx

Buka sesi interaktif untuk melihat dan mengelola pelajaran yang telah dipelajari definisi transformasi dari proses sebelumnya. Jelajahi pelajaran berdasarkan kategori, lihat detail pelajaran, dan arsipkan, pulihkan, atau hapus pelajaran.

```
atx custom def learnings -n <transformation-name>
```

Perintah Tag

tag daftar def kustom atx

Daftar tag untuk definisi transformasi.

```
atx custom def list-tags --arn <transformation-arn>
```

tag def kustom atx

Tambahkan tag ke definisi transformasi.

```
atx custom def tag --arn <arn> --tags '{"env":"prod","team":"backend"}'
```

atx kustom def untag

Hapus tag dari definisi transformasi.

```
atx custom def untag --arn <arn> --tag-keys "env,team"
```

Perbarui Perintah

pembaruan atx

Perbarui CLI AWS Transform.

```
atx update                    # Update to latest version
atx update --check            # Check for updates only
atx update --target-version <version>  # Update to specific version
```

Perintah MCP

alat atx mcp

Memungkinkan klien untuk konfigurasi alat MCP view/confirm mereka. Pembaruan konfigurasi harus dikelola langsung melalui file ~/. aws/atx/mcp.json

```

atx mcp tools     # View list of MCP servers
atx mcp tools -s    # List tools for the specified server
```

AWS-Transformasi Terkelola

AWS Transformasi terkelola adalah transformasi yang telah dibuat sebelumnya dan AWS diperiksa untuk kasus penggunaan umum yang siap digunakan tanpa pengaturan tambahan.

Gambaran umum

AWS-transformasi terkelola memiliki karakteristik sebagai berikut:

- Divalidasi oleh AWS - Transformasi ini diperiksa oleh AWS untuk menjadi berkualitas tinggi
- Siap digunakan - Tidak diperlukan pengaturan tambahan
- Terus tumbuh - Transformasi tambahan terus ditambahkan
- Dapat disesuaikan - Pre-built transformasi dapat disesuaikan dengan memberikan panduan tambahan atau persyaratan khusus untuk kebutuhan organisasi Anda menggunakan parameter konfigurasi `additionalPlanContext`
- Dukungan akses awal - Beberapa transformasi dapat ditandai sebagai akses awal karena mereka menjalani pengujian dan penyempurnaan lebih lanjut

Available AWS-Transformasi Terkelola

Katalog berikut mencantumkan transformasi AWS-managed yang saat ini tersedia, dikelompokkan berdasarkan kasus penggunaan. Setiap tabel menunjukkan nama transformasi, bahasa atau tumpukan yang diterapkan, dan statusnya. Untuk menemukan transformasi dengan cepat, gunakan kotak pencarian di bagian atas halaman ini untuk mencari berdasarkan nama, bahasa, atau kasus penggunaan.

Katalog ini adalah daftar kanonik transformasi AWS-managed. Kiro Power, plugin agen, plugin VS Code, dan CLI semuanya menjalankan transformasi yang sama. Untuk membuat daftar transformasi yang tersedia di registri Anda, jalankan`atx custom def list`.

> **Note**
>
> Transformasi ditandai Akses awal berfungsi tetapi mungkin sering diperbarui berdasarkan umpan balik pelanggan.

Upgrade runtime

Tingkatkan bahasa atau runtime ke versi yang lebih baru, atau ubah distribusi runtime.

Transformasi	Bahasa/tumpukan	Status	Deskripsi
`AWS/java-version-upgrade`	Java	Umumnya tersedia	Tingkatkan aplikasi Java menggunakan sistem build apa pun dari versi JDK sumber apa pun ke versi JDK target apa pun, dengan modernisasi ketergantungan termasuk migrasi Jakarta EE, driver database, kerangka kerja ORM, dan pembaruan ekosistem Spring. Tentukan versi JDK target dalam obrolan atau dengan parameter `.additionalPlanContext`
`AWS/python-version-upgrade`	Python	Umumnya tersedia	Migrasikan proyek Python dari Python 3.8 atau 3.9 ke Python 3.11, 3.12, atau 3.13 sambil mempertahankan fungsionalitas dan kinerja. Tentukan versi Python target dalam obrolan atau dengan parameter `.additionalPlanContext`

Transformasi	Bahasa/tumpukan	Status	Deskripsi
`AWS/nodejs-version-upgrade`	Node.js	Umumnya tersedia	Tingkatkan Node.js aplikasi dari Node.js versi sumber apa pun ke Node.js versi target apa pun. Tentukan versi target dalam obrolan atau dengan `additionalPlanContext` parameter.
`AWS/lambda-nodejs-runtime-upgrade`	Node.js (Lambda)	Akses awal	Tingkatkan fungsi AWS Lambda dari Node.js runtime yang lebih lama (nodejs4.3 hingga nodejs22.x) ke nodejs24.x, mengatasi perubahan yang melanggar di Lambda Runtime Interface Client (RIC) dan runtime 24 bahasa. Node.js

Transformasi	Bahasa/tumpukan	Status	Deskripsi
`AWS/oracle-java-to-corretto`	Java (distribusi JDK)	Akses awal	Migrasikan proyek Java dari Oracle JDK ke Amazon Corretto. Mengganti API Oracle-specific internal dengan setara Java standar, memperbarui konfigurasi build (Maven dan Gradle) dan gambar dasar kontainer, menghapus flag JVM komersial , dan menghasilkan laporan lisensi.

Transformasi	Bahasa/tumpukan	Status	Deskripsi
`AWS/ruby-upgrade`	Ruby	Akses awal	Tingkatkan aplikasi Ruby dari Ruby 2.x ke Ruby 4.0, dan kerangka kerjanya ke Rails 8.0 atau Sinatra 4.1. Anda dapat menggunakan transformasi ini dengan Rails dan ActiveRecord aplikasi, aplikasi Sinatra, dan permata mandiri. Transformasi memasangkan setiap benjolan versi Ruby dengan peningkatan kerangka kerja yang cocok dan menjalankan rangkaian pengujian. Kemudian mengisolasi kegagalan sebelum maju ke versi berikutnya.

Migrasi SDK

Migrasi antar versi utama AWS SDK.

Transformasi	Bahasa/tumpukan	Status	Deskripsi
`AWS/java-aws-sdk-v1-to-v2`	Java	Umumnya tersedia	Tingkatkan AWS SDK dari v1 ke v2 untuk proyek Java

Transformasi	Bahasa/tumpukan	Status	Deskripsi
			menggunakan Maven atau Gradle.
`AWS/python-boto2-to-boto3`	Python	Umumnya tersedia	Migrasikan aplikasi Python dari boto2 ke boto3, berdasarkan dokumentasi migrasi resmi. AWS
`AWS/nodejs-aws-sdk-v2-to-v3`	Node.js	Umumnya tersedia	Tingkatkan Node.js aplikasi dari AWS SDK untuk JavaScrip t v2 ke v3 untuk arsitektur modular, TypeScript dukungan kelas satu, dan peningkatan kinerja, tanpa mengubah versi. Node.js

Upgrade dan migrasi kerangka kerja

Tingkatkan kerangka kerja ke versi yang lebih baru, atau migrasi ke kerangka kerja yang berbeda.

Transformasi	Bahasa/tumpukan	Status	Deskripsi
`AWS/spring-boot-version-upgrade`	Boot Java /Musim Semi	Umumnya tersedia	Tingkatkan aplikasi Spring Boot yang lebih lama ke versi Spring Boot target, termasuk file build dan starter, properti, paket Jackson 3, Spring Security 7 DSL, infrastru

Transformasi	Bahasa/tumpukan	Status	Deskripsi
			ktur pengujian, dan dependensi observabi litas.
`AWS/JBoss-to-Spring-Boot`	Java (JBoss atau WildFly ke Spring Boot)	Akses awal	Migrasikan aplikasi perusahaan Java EE atau Jakarta EE yang berjalan di JBoss EAP atau WildFly ke Spring Boot, menghilangkan dependensi server aplikasi untuk model penerapan kontainer cloud-native.
`AWS/early-access-angular-to-react-migration`	Angular untuk Bereaksi	Akses awal	Ubah aplikasi Angular menjadi React.
`AWS/angular-version-upgrade`	Sudut	Akses awal	Tingkatkan aplikasi Angular yang lebih lama ke versi Angular target, termasuk komponen, layanan, templat, dan perutean.

Transformasi	Bahasa/tumpukan	Status	Deskripsi
`AWS/vue.js-version-upgrade`	Vue.js	Akses awal	Lakukan peningkatan versi utama dari Vue.js 2 menjadi Vue.js 3, memodernisasi komponen, manajemen status, perutean, dan API global. Pembaruan minor dan patch berada di luar cakupan.

GenAI dan migrasi model

Migrasikan beban kerja AI generatif ke.

Transformasi	Bahasa/tumpukan	Status	Deskripsi
`AWS/GenAI-to-Bedrock-Migration-Assessment`	Beban kerja AI generatif	Akses awal	Nilai beban kerja AI generatif untuk migrasi dari penyedia pihak ketiga (OpenAI, Google Gemini, Anthropic direct, dan model sumber terbuka) ke. Menemukan penggunaan dan model SDK, mengklarifikasi persyaratan, merancang pemetaan model, memperkirakan biaya dan risiko, dan menghasilkan

Transformasi	Bahasa/tumpukan	Status	Deskripsi
			artefak penilaian. Assessment-only; tidak mengubah kode sumber. Juga mencakup kerangka kerja agen seperti CreWai,, LangGraph dan Strands.

Language-to-language migrasi

Terjemahkan basis kode dari satu bahasa pemrograman ke bahasa pemrograman lainnya.

Transformasi	Bahasa/tumpukan	Status	Deskripsi
`AWS/vba-to-python-migration`	VBA KE Python	Akses awal	Migrasikan makro, modul UserForms, dan logika tertanam Excel VBA ke skrip dan modul Python yang setara, menggunakan openpyxl, pandas, tkinter atau 6, dan pustaka standar. PyQt Bahasa target adalah Python 3.8 atau yang lebih baru. Gunakan saat mengonversi kode VBA .bas, .cls, .frm, atau .xlsm-embedded.

Migrasi basis data

Saat ini tidak ada transformasi AWS terkelola khusus basis data yang tersedia. Untuk memodernisasi database Microsoft SQL Server dan aplikasi.NET terkait ke Amazon Aurora PostgreSQL, lihat. the section called “Modernisasi SQL Server”

Observabilitas

Memodernisasi pencatatan dan pemantauan ke layanan AWS-native.

Transformasi	Bahasa/tumpukan	Status	Deskripsi
`AWS/early-access-log4j-to-slf4j-migration`	Java (pencatatan)	Akses awal	Migrasikan aplikasi Java dari Log4j (1.x atau 2.x) ke SLF4J dengan backend Logback. Menangani kode sumber, manajemen ketergantungan (Maven dan Gradle), dan mencatat file konfigurasi, dan memvalidasi dengan kompilasi, pengujian, dan pemindaian impor sisa.
`AWS/datadog-monitors-to-cloudwatch-alarms`	Infrastruktur sebagai kode/pemantauan	Akses awal	Migrasikan monitor DataDog metrik yang melacak metrik AWS layanan dan metrik khusus ke CloudWatch alarm asli sebagai infrastruktur sebagai kode. Menghasilkan CloudFormation YAMAL, CDK

Transformasi	Bahasa/tumpukan	Status	Deskripsi
			TypeScript, dan Terraform HCL.

Arsitektur

Re-architect, platform ulang, atau optimalkan aplikasi untuk AWS.

Transformasi	Bahasa/tumpukan	Status	Deskripsi
`AWS/java-performance-optimization`	Java (kinerja)	Umumnya tersedia	Optimalkan kinerja aplikasi Java dengan menganalisis data profil JFR untuk mendeteksi hotspot CPU dan memori serta anti-pola, kemudian menerapkan perbaikan kode yang ditargetkan. Untuk petunjuk tentang mengumpulkan data JFR, lihat panduan runtime JFR.
`AWS/early-access-java-x86-to-graviton`	Jawa (Arm64/Graviton)	Akses awal	Validasi kompatibilitas aplikasi Java dengan arsitektur Arm64 untuk prosesor AWS Graviton, dan selesaikan ketidakcocokan dengan memperbarui dependensi, mendeteksi pola kode khusus arsitektur, dan

Transformasi	Bahasa/tumpukan	Status	Deskripsi
			mengkompilasi ulang pustaka asli saat kode sumber tersedia. Banyak aplikasi Java modern sudah ada Arm64-compatible.
`AWS/oracle-service-bus-to-aws`	Java /integrasi ke tanpa server	Akses awal	Migrasikan konfigurasi proses Oracle Service Bus (OSB) dan BPEL ke arsitektur tanpa server AWS-native, menghasilkan proyek CDK yang dapat di-deploy dengan TypeScript Amazon API Gateway, Lambda, dan Step Functions. AWS AWS

Transformasi	Bahasa/tumpukan	Status	Deskripsi
`AWS/mulesoft-to-aws-native`	Java MuleSoft /ke tanpa server	Akses awal	Ubah aplikasi MuleSoft Mule 3.x dan 4.x menjadi arsitektur AWS tanpa server (Amazon API Gateway, AWS Lambda, dan AWS Step Functions) yang menargetkan Java 17 di Lambda dengan. AWS SnapStart Aliran peta transformasi memicu sumber AWS peristiwa, mengonversi DataWeave dan ekspresi MEL (MuleSoft Expression Language) menjadi AWS setara, dan menggantikan konektor dengan panggilan SDK for AWS Java v2. Ini menghasilkan template SAM, penangan AWS Lambda, dan tes JUnit 5.

Transformasi	Bahasa/tumpukan	Status	Deskripsi
`AWS/payshield-hsm-to-aws-payment-cryptography`	Java (HSM ke Kriptografi AWS Pembayaran)	Umumnya tersedia	Migrasikan aplikasi Java dari protokol modul keamanan PayShield perangkat keras Thales (HSM) ke AWS Payment Cryptography SDK v2. Migrasi mencakup pemetaan perintah, migrasi kunci, pengaturan ketergantungan, dan pengujian paritas.

Analisis basis kode

Menganalisis basis kode dan portofolio untuk merencanakan modernisasi. Transformasi ini menghasilkan laporan dan tidak mengubah kode Anda.

Transformasi	Bahasa/tumpukan	Status	Deskripsi
`AWS/comprehensive-codebase-analysis`	Beberapa bahasa	Umumnya tersedia	Lakukan analisis statis mendalam dari basis kode untuk menghasilkan dokumentasi hierarkis dan referensi silang yang menggabungkan analisis perilaku, dokumentasi arsitektur, dan intelijen bisnis, dengan penekanan pada wawasan utang teknis.

Transformasi	Bahasa/tumpukan	Status	Deskripsi
`AWS/agentic-readiness-analysis`	Beberapa bahasa	Akses awal	Evaluasi apakah sistem siap dipanggil dengan aman oleh agen AI, yang mencakup API, identitas, manajemen negara, manusia-in-the-loop, dan observabilitas.
`AWS/modernization-readiness-analysis`	Beberapa bahasa	Akses awal	Pindai portofolio untuk kesenjangan kematangan cloud-native dan petakan temuan ke jalur modernisasi. AWS
`AWS/portfolio-agentic-readiness-analysis`	Portofolio	Akses awal	Agregat laporan analisis kesiapan agen individu di seluruh portofolio aplikasi untuk mengidentifikasi pemblokir lintas sektoral, pola remediasi bersama, dan rekomendasi yang diprioritaskan.

Transformasi	Bahasa/tumpukan	Status	Deskripsi
`AWS/portfolio-modernization-readiness-analysis`	Portofolio	Akses awal	Agregat laporan analisis modernisasi individu di seluruh portofolio ke dalam peta jalan terkonsolidasi dengan gelombang migrasi yang diprioritaskan dan jalur modernisasi yang direkomendasikan.

Transformasi	Bahasa/tumpukan	Status	Deskripsi
`AWS/business-rules-extraction`	Beberapa bahasa	Akses awal	Transformasi ini secara statis menganalisis basis kode monolitik untuk menghasilkan dokumentasi siap tulis ulang tanpa memerlukan modifikasi build, run, atau kode sumber. Ini menguraikan sistem menjadi domain terbatas dan mengekstrak aturan bisnis per domain, alur kerja, masalah lintas sektoral, dan struktur basis data. Outputnya mencakup manifes yang dapat dibaca mesin, dasbor interaktif, dan spesifikasi implementasi netral bahasa per domain untuk memandu modernisasi atau penulisan ulang penuh.

Menyesuaikan AWS-Transformasi Terkelola

Anda dapat menyesuaikan transformasi yang AWS dikelola untuk memenuhi kebutuhan spesifik organisasi Anda dengan menyediakan konteks tambahan melalui parameter `additionalPlanContext` konfigurasi.

Contoh: Menyesuaikan upgrade versi Java

```
codeRepositoryPath: ./my-project
transformationName: AWS/java-version-upgrade
buildCommand: mvn clean install
additionalPlanContext: |
  The target Java version to upgrade to is Java 17.
  Update all internal library dependencies to versions compatible with Java 17.
  Ensure compatibility with our custom authentication framework.
```

Contoh: Menyesuaikan migrasi AWS SDK

```
codeRepositoryPath: ./my-project
transformationName: AWS/java-aws-sdk-v1-to-v2
buildCommand: gradle build
additionalPlanContext: |
  Maintain our existing error handling patterns.
  Use our organization's standard credential provider chain.
  Update logging to use our internal logging framework.
```

Kasus Penggunaan Umum

Bagian ini memberikan panduan langkah demi langkah untuk skenario transformasi umum menggunakan AWS Transform CLI.

Peningkatan Runtime Lambda

Lambda secara berkala menghentikan runtime bahasa yang lebih lama, mengharuskan tim untuk meningkatkan fungsi Lambda mereka ke versi yang didukung. AWS Transform custom dapat mengotomatiskan upgrade ini menggunakan transformasi upgrade versi bahasa yang AWS dikelola. Misalnya, untuk memutakhirkan fungsi Lambda yang ditulis dengan Python 3.9 ke Python 3.13, Anda akan menggunakan transformasi. `AWS/python-version-upgrade` Transformasi serupa tersedia untuk Java (`AWS/java-version-upgrade`) dan Node.js (`AWS/nodejs-version-upgrade`), yang mencakup bahasa runtime Lambda yang paling umum.

Untuk memutakhirkan satu fungsi Lambda secara interaktif, navigasikan ke root repositori sumber fungsi Anda dan mulai Transform CLI dengan. AWS `atx` Dalam sesi interaktif, jelaskan peningkatan yang Anda butuhkan—misalnya, “Tingkatkan fungsi Lambda Python 3.9 ini ke Python 3.13.” Agen akan memilih transformasi AWS-managed yang sesuai, menganalisis kode Anda, memperbarui sintaks bahasa, menyesuaikan dependensi, dan memodernisasi panggilan API yang tidak digunakan

lagi. Anda dapat memberikan perintah build atau validasi seperti `pytest` atau `python -m py_compile *.py` agar agen dapat memverifikasi kompilasi kode yang diubah dan lulus tes. Sepanjang sesi interaktif, Anda dapat meninjau perubahan yang diusulkan agen, memberikan umpan balik, dan meminta penyesuaian sebelum menerima hasil akhir.

Saat Anda perlu memutakhirkan runtime Lambda di banyak repositori, gunakan mode non-interaktif untuk menjalankan transformasi dalam skala besar. Untuk setiap repositori, panggil CLI dengan flag yang sesuai untuk dijalankan tanpa input pengguna. Misalnya, untuk meningkatkan fungsi Node.js 16 Lambda ke Node.js 22:

```
atx custom def exec \
  -n AWS/nodejs-version-upgrade \
  -p ./my-lambda-repo \
  -c "npm test" \
  -g "additionalPlanContext='Upgrade from Node.js 16 to Node.js 22'" \
  -x -t
```

`-x`Bendera menjalankan transformasi secara non-interaktif, dan `-t` mempercayai semua eksekusi alat sehingga tidak ada permintaan yang diperlukan. Anda dapat membuat skrip perintah ini di lusinan atau ratusan repositori menggunakan loop shell sederhana. Setelah setiap transformasi selesai, tinjau perbedaan Git untuk memverifikasi perubahan dan jalankan rangkaian pengujian Anda untuk mengonfirmasi bahwa fungsi yang ditingkatkan berfungsi dengan benar. Anda juga dapat mengintegrasikan kustom AWS Transform ke dalam CI/CD pipeline untuk mengidentifikasi fungsi Lambda yang tidak digunakan lagi secara berkelanjutan. Pendekatan ini tidak hanya mengatasi masalah langsung runtime yang tidak digunakan lagi, tetapi juga menetapkan proses berkelanjutan untuk mendeteksi dan meningkatkan fungsi sebelum mencapai status dukungan akhir.

Tim platform pusat dapat membuat kampanye melalui aplikasi web AWS Transform untuk menentukan transformasi, menentukan target repositori Lambda, dan melacak kemajuan di seluruh organisasi. Untuk panduan mendetail dalam membangun solusi penerapan skala tingkat produksi menggunakan AWS Batch, lihat Membangun solusi modernisasi kode yang dapat diskalakan dengan Kustom Transform. AWS

AWS Transform titik akhir kustom dan antarmuka (AWS PrivateLink)

Anda dapat membuat koneksi pribadi antara VPC dan AWS Transform kustom Anda dengan membuat antarmuka VPC endpoint. Endpoint antarmuka didukung oleh AWS PrivateLink, teknologi

yang memungkinkan Anda mengakses layanan AWS Transform khusus secara pribadi tanpa gateway internet, perangkat NAT, koneksi VPN, atau koneksi. Direct Connect Lalu lintas antara VPC dan AWS Transform kustom Anda tidak meninggalkan jaringan Amazon.

Setiap titik akhir antarmuka diwakili oleh satu atau beberapa Antarmuka Jaringan Elastis di subnet Anda.

Untuk informasi selengkapnya, lihat Antarmuka VPC endpoint (AWS PrivateLink) dalam Panduan Pengguna Amazon VPC.

> **Note**
>
> - Anda harus mengaktifkan AWS PrivateLink integrasi untuk Amazon S3 karena AWS Transform kustom membuat panggilan API S3. Untuk petunjuk terperinci, lihat dokumentasi AWS PrivateLink untuk Amazon S3. Jika Anda mengalami masalah akses S3 saat menggunakan AWS Transform kustom, lihat panduan pemecahan masalah kami.
> - Jika Anda tidak menggunakan fitur DNS AWS PrivateLink Pribadi (lihat DNS Pribadi), Anda harus:
>   - Konfigurasikan perutean ke titik akhir antarmuka VPC (lihat dokumentasi titik akhir antarmuka Routing ke VPC)
>   - Tetapkan variabel `ATX_CUSTOM_ENDPOINT` lingkungan untuk menentukan domain kustom Anda, misalnya:
>
>     ```
>     ATX_CUSTOM_ENDPOINT=https://transform-custom.<region>.api.aws atx
>     ```

Pertimbangan untuk AWS Transform titik akhir VPC kustom

Sebelum menyiapkan titik akhir VPC antarmuka untuk AWS Transform kustom, pastikan Anda meninjau properti dan batasan titik akhir Antarmuka di Panduan Pengguna Amazon VPC.

AWS Transform kustom mendukung panggilan ke semua tindakan API-nya melalui titik akhir antarmuka.

Prasyarat

Sebelum Anda memulai salah satu prosedur di bawah ini, pastikan Anda memiliki yang berikut:

- AWS Akun dengan izin yang sesuai untuk membuat dan mengonfigurasi sumber daya.
- VPC sudah dibuat di akun Anda AWS .
- Keakraban dengan AWS layanan, terutama Amazon VPC AWS Transform dan custom.

Membuat titik akhir VPC antarmuka untuk AWS Transform kebiasaan

Anda dapat membuat titik akhir VPC untuk layanan AWS Transform kustom menggunakan konsol VPC Amazon atau (). AWS Command Line Interface AWS CLI Untuk informasi selengkapnya, lihat Membuat titik akhir antarmuka dalam Panduan Pengguna Amazon VPC.

Buat titik akhir VPC berikut untuk AWS Transform kustom menggunakan nama layanan ini:

- com.amazonaws. *region*.transform-kustom

Ganti *region* dengan AWS Wilayah tempat Anda ingin menggunakan CLI AWS Transform khusus, misalnya, com.amazonaws.us-east-1.transform-custom.

Untuk informasi selengkapnya, lihat Mengakses layanan melalui titik akhir antarmuka dalam Panduan Pengguna Amazon VPC.

Membuat kebijakan titik akhir VPC untuk AWS Transform kebiasaan

Anda dapat melampirkan kebijakan titik akhir ke titik akhir VPC yang mengontrol akses ke kustom. AWS Transform Kebijakan titik akhir menentukan informasi berikut:

- Prinsipal yang dapat melakukan tindakan.
- Tindakan yang dapat dilakukan.
- Sumber daya yang menjadi target tindakan.

Untuk informasi selengkapnya, lihat Mengontrol Akses ke Layanan dengan titik akhir VPC dalam Panduan Pengguna Amazon VPC.

Contoh: Kebijakan titik akhir VPC untuk AWS Transform tindakan kustom

Berikut ini adalah contoh kebijakan endpoint untuk AWS Transform kustom. Saat dilampirkan ke titik akhir, kebijakan ini memberikan akses ke tindakan AWS Transform kustom yang tercantum untuk semua prinsipal di semua sumber daya.

```
{
   "Statement":[
      {
         "Principal":"*",
         "Effect":"Allow",
         "Action":[
            "transform-custom:*"
         ],
         "Resource":"*"
      }
   ]
}
```

Menggunakan komputer lokal untuk menyambung ke AWS Transform titik akhir kustom

Bagian ini menjelaskan proses penggunaan komputer lokal untuk terhubung ke AWS Transform kustom melalui AWS PrivateLink titik akhir di AWS VPC Anda.

1. Buat koneksi VPN antara perangkat lokal dan VPC Anda.
2. Buat titik akhir VPC antarmuka untuk kustom. AWS Transform
3. Siapkan titik akhir Amazon Route 53 masuk. Ini akan memungkinkan Anda untuk menggunakan nama DNS titik akhir AWS Transform kustom Anda dari perangkat lokal Anda.

Pemecahan masalah

Bagian ini memberikan panduan untuk memecahkan masalah umum dengan AWS mengubah kustom.

Lokasi Log

AWS Transform CLI memelihara log di lokasi berikut:

Log percakapan:

Linux and macOS

```
~/.aws/atx/custom/<conversation_id>/logs/<timestamp>-conversation.log
```

Windows

```
%USERPROFILE%\.aws\atx\custom\<conversation_id>\logs\<timestamp>-conversation.log
```

Log ini berisi riwayat percakapan lengkap untuk men-debug eksekusi transformasi tertentu.

Log subagent:

Linux and macOS

```
~/.aws/atx/custom/<conversation_id>/logs/subagents/<name>.log
```

Windows

```
%USERPROFILE%\.aws\atx\custom\<conversation_id>\logs\subagents\<name>.log
```

Log ini berisi output dari subagen yang ditimbulkan oleh agen utama selama transformasi. Anda tidak perlu mengelola subagen secara langsung.

Log debug pengembang:

Linux and macOS

```
~/.aws/atx/logs/debug*.log
~/.aws/atx/logs/error.log
```

Windows

```
%USERPROFILE%\.aws\atx\logs\debug*.log
%USERPROFILE%\.aws\atx\logs\error.log
```

Log ini memberikan informasi rinci tentang operasi dan kesalahan CLI. Setiap file log memiliki batas 5MB sebelum berguling. Mungkin ada beberapa log debug di direktori ini, seperti debug1.log dan debug2.log.

Masalah Umum

Masalah instalasi:

Jika instalasi gagal, pastikan Anda memiliki Node.js 22 atau lebih baru diinstal:

```
node --version
```

Unduh Node.js dari https://nodejs.org/en/download jika diperlukan.

Masalah otentikasi:

Verifikasi AWS kredensil Anda dikonfigurasi dengan benar:

```
aws sts get-caller-identity
```

Pastikan pengguna atau peran IAM Anda memiliki `transform-custom:*` izin yang diperlukan.

Masalah konektivitas jaringan:

Jika Anda mengalami kesalahan koneksi, verifikasi akses jaringan ke titik akhir yang diperlukan:

- `transform-cli.awsstatic.com`
- `transform-custom.<region>.api.aws`
- `*.s3.amazonaws.com`

Jika bekerja di lingkungan yang dibatasi internet, perbarui aturan firewall untuk mengizinkan daftar URL ini.

Masalah konfigurasi wilayah:

Jika Anda menemukan kesalahan terkait wilayah:

- Verifikasi wilayah Anda didukung
- Periksa variabel lingkungan yang dapat menggantikan konfigurasi Anda: `echo $AWS_REGION $AWS_DEFAULT_REGION`
- Periksa konfigurasi wilayah Anda: `aws configure get region`
- Perbarui wilayah Anda jika diperlukan: `aws configure set region <your-region>`
- Periksa log debug untuk detail resolusi wilayah

Masalah Git:

Pastikan Git diinstal dan repositori Anda berada di bawah kontrol sumber git:

```
git --version
git status
```

AWS Transform custom membutuhkan repositori berada di bawah kontrol sumber git.

Masalah eksekusi transformasi:

Jika transformasi gagal:

1. Tinjau log percakapan di `~/.aws/atx/custom/<conversation_id>/logs/`
2. Periksa kegagalan build atau pengujian dalam output transformasi
3. Verifikasi bahwa perintah build sudah benar untuk project Anda
4. Coba jalankan transformasi dalam mode interaktif untuk memberikan umpan balik

Masalah dimulainya kembali percakapan:

Jika Anda tidak dapat melanjutkan percakapan:

- Pastikan percakapan berusia kurang dari 30 hari
- Periksa ID percakapan sudah benar
- Pastikan Anda memiliki konektivitas jaringan

Mendapatkan Dukungan

Untuk bantuan tambahan, kunjungi AWS Support melalui AWS Konsol.

Saat membuka tiket dukungan, sertakan:

- Log percakapan dari `~/.aws/atx/custom/<conversation_id>/logs/`
- Log subagent dari `~/.aws/atx/custom/<conversation_id>/logs/subagents/`
- Debug log dari `~/.aws/atx/logs/`
- Langkah-langkah untuk mereproduksi masalah
- AWS Transform versi CLI () `atx --version`

Masalah akses S3

Layanan kustom AWS Transform menjual URL S3 yang telah ditandatangani sebelumnya untuk memfasilitasi pengunggahan dan pengunduhan mesin klien file transformasi yang berpotensi besar. to/from Ini berarti bahwa CLI tidak hanya berinteraksi dengan titik akhir layanan AWS Transform tetapi juga berinteraksi dengan titik akhir layanan S3. Interaksi ini menggunakan URL yang telah ditandatangani sebelumnya sehingga kredensyal IAM klien Anda tidak memerlukan izin S3 apa pun, tetapi terkadang konfigurasi server proxy mesin lokal Anda dan Kebijakan Titik Akhir VPC S3 jaringan Anda dapat membatasi lalu lintas ini.

Silakan periksa the section called “Lokasi Log” untuk pesan kesalahan yang lebih rinci untuk membantu root menyebabkan kegagalan alat apa pun atau akses S3 ditolak terkait dengan mengunduh dan mengunggah file transformasi.

Jika Anda menggunakan VPN/Proxy server yang memanfaatkan variabel `https_proxy` dan `no_proxy` lingkungan, pertimbangkan untuk menambahkan nilai berikut ke nilai variabel `no_proxy` lingkungan Anda untuk melewati proxy untuk titik akhir layanan kustom S3 dan AWS Transform, misalnya:

Linux and macOS

```
export no_proxy=.s3.amazonaws.com,transform-custom.<region>.api.aws
```

Windows (PowerShell)

```
$env:no_proxy=".s3.amazonaws.com,transform-custom.<region>.api.aws"
```

Jika Anda menggunakan AWS PrivateLink untuk Amazon S3 di VPC Anda, ini mungkin memiliki kebijakan yang ditetapkan untuk membatasi lalu lintas S3. Pastikan kebijakan titik akhir VPC S3 Anda `GetObject` mengizinkan `PutObject` dan beroperasi untuk AWS bucket milik layanan khusus Transform. Ini dapat dicapai dengan menambahkan pernyataan berikut ke kebijakan titik akhir VPC Anda:

```
{
    "Effect": "Allow",
    "Principal": "*",
    "Action": ["s3:PutObject","s3:GetObject"],
    "Resource": "arn:aws:s3:::aws-transform-custom-*/*"
```

```
}
```

DenyPernyataan eksplisit dalam kebijakan lebih diutamakan daripada pernyataan. `Allow` Tinjau kebijakan Titik Akhir VPC S3 `Deny` untuk pernyataan yang mungkin membatasi akses.

AWS riwayat versi Transform CLI

Tinjau tabel berikut untuk detail tentang rilis CLI AWS Transform saat ini dan sebelumnya. Tabel mencakup versi, tanggal rilis, dan catatan rilis untuk setiap versi.

Versi	Tanggal rilis	Catatan rilis
3.6.0 (terbaru)	Juli 17, 2026	• Menambahkan `atx custom def learnings` subperintah, yang membuka sesi TUI interaktif untuk melihat dan mengelola pembelajaran yang dihasilkan secara otomatis untuk definisi transformasi berdasarkan proses sebelumnya. • Menambahkan koleksi telemetri penggunaan selama eksekusi transformasi. Untuk menonaktifkan telemetri, atur variabel `ATX_DISABLE_TELEMETRY=true` lingkungan. Untuk informasi selengkapnya, lihat Penggunaan Telemetri. • Memperbaiki masalah pada Windows di mana operasi file menggunakan jalur ~ (direktori home) diselesaikan ke direktori yang tidak terduga. Hal ini menyebabkan read/write kegagalan file selama sesi. • Memperbaiki masalah di mana server MCP dan nama alat yang berisi karakter khusus dapat menyebabkan kegagalan koneksi. CLI sekarang secara otomatis membersihkan nama untuk hanya menggunakan karakter yang valid.
3.5.0	Juli 8, 2026	• Menambahkan dukungan Windows asli. Anda sekarang dapat menjalankan perintah AWS Transform Custom langsung di Windows. (`atx ct`Perintah belum didukung pada Windows) • Input kemudi yang ditambahkan ke analisis `atx ct analysis run` kesiapan agen dan kesiapan modernisasi mencerminkan preferensi Anda. Gunakan `--context`

Versi	Tanggal rilis	Catatan rilis
		untuk pembingkaian strategis, `--agent-scope` untuk menetapkan akses agen untuk kesiapan agen, `--prefer` dan/atau untuk mengarahkan rekomendasi modernisasi `--avoid` menuju atau menjauh dari teknologi tertentu. • Definisi transformasi dalam format keterampilan sekarang dapat menyertakan `scripts/` folder, memungkinkan Anda menggabungkan skrip pembantu di samping referensi Anda`SKILL.md`. • Ditambahkan `--wait` ke`atx ct remediation create`, yang menunggu sampai remediasi selesai. • Menambahkan mode tanpa kepala untuk eksekusi satu tembakan. Jalankan `atx -x "prompt" -t` untuk meneruskan prompt, mendapatkan respons, dan keluar tanpa petunjuk interaktif. Mode tanpa kepala membutuhkan kepercayaan semua alat.
3.4.1	Juli 2, 2026	• Memperbaiki masalah sesi interaktif yang diperkenalkan versi 3.4.0. Setelah agen selesai merespons, CLI mungkin tetap pada prompt “Berpikir” (ditampilkan dengan elipsis di UI) alih-alih kembali ke prompt input. CLI sekarang mengembalikan Anda ke prompt input setelah setiap respons.

Versi	Tanggal rilis	Catatan rilis
3.4.0 (tidak disarankan)	Juni 29, 2026	⚠ Masalah yang diketahui Dalam sesi interaktif, CLI mungkin tetap pada prompt “Berpikir” (ditampilkan dengan elipsis di UI) setelah agen selesai merespons. Ini mencegah Anda memasukkan pesan berikutnya. Untuk mengatasi masalah ini, tingkatkan ke versi 3.4.1 atau yang lebih baru dengan menjalankan`atx update`. • Ditambahkan`atx ct schema`, manifes yang dapat dibaca mesin dari permukaan `atx ct` perintah penuh untuk skrip dan otomatisasi. • Menambahkan validasi cakupan token saat Anda menambahkan atau memperbarui sumber, segera memunculkan masalah kredensi atau izin alih-alih gagal nanti selama analisis. • Permintaan tarik remediasi otomatis yang ditingkatkan untuk menyertakan konteks pencarian dan mendukung mode draf untuk PR yang lebih dapat ditinjau. • Umpan balik kemajuan CLI yang ditingkatkan dengan konteks waktu nyata tentang aktivitas saat ini, perkiraan menit agen, dan aktivitas sub-agen. • Ditambahkan pos pemeriksaan yang dapat dikonfigu rasi identitas penulis komit melalui variabel `ATX_GIT_C OMMITTER_NAME` dan `ATX_GIT_COMMITTER_EMAIL` lingkungan. • Berbagai perbaikan bug kecil dan perbaikan.

Versi	Tanggal rilis	Catatan rilis
3.3.0	Juni 18, 2026	• Ditambahkan konsisten `--json` dan `--help` dukungan di seluruh `atx ct` perintah. • Pengaturan agen keamanan tetap sehingga ruang agen dibuat selama pengaturan dan dihapus saat pembongkaran. • Berbagai perbaikan bug kecil dan perbaikan.
3.2.0	Juni 15, 2026	• Menambahkan perintah modernisasi berkelanjutan (`atx ct`) ke distribusi CLI standar untuk semua pengguna. • Peningkatan eksekusi sub-agen untuk menampilkan pemintal “Berpikir...” sambil menunggu respons server, memberikan umpan balik visual yang jelas bahwa pekerjaan sedang berlangsung. • Pemintal yang ditingkatkan untuk memulai ulang secara otomatis dengan “Mengerjakannya...” setelah 3 detik hening mengikuti keluaran agen, sehingga terminal tidak pernah tampak beku di antara langkah-langkah pemrosesan. • Pemintal berpikir yang ditingkatkan dengan animasi kilau halus untuk tampilan visual yang lebih halus.

Versi	Tanggal rilis	Catatan rilis
3.1.0	Juni 3, 2026	• Menambahkan variabel `ATX_MAX_VALIDATION_LOOPS` lingkungan untuk mengkonfigurasi loop validasi maksimum. • Pemeriksaan pembaruan tetap tidak lagi menunjukkan peringatan “pembaruan tersedia” palsu ketika versi yang diinstal sudah lebih baru dari versi registri. • Diperbaiki `delete --transformation-name ""` tidak lagi menunjukkan prompt konfirmasi yang tidak masuk akal; nilai kosong sekarang ditolak dengan kesalahan validasi yang jelas. • Subperintah yang tidak dikenal tetap sekarang menampilk an “perintah tidak dikenal” alih-alih pesan kesalahan internal yang membingungkan. • Perbandingan versi tetap untuk menggunakan perbandin gan numerik alih-alih ketidaksetaraan string untuk pemeriksaan pembaruan.
3.0.0	21 Mei 2026	• Menambahkan validasi nama transformasi pra-penerbangan yang menyarankan nama serupa dari registri Anda ketika nama tidak ditemukan, membantu menangkap kesalahan ketik tanpa memerlukan perintah daftar terpisah. • Menambahkan penjelasan opsional untuk perintah shell, menunjukkan apa yang mereka lakukan dan mengapa. • Spanduk pengaturan kepercayaan yang ditingkatkan untuk menyertakan informasi lebih lanjut tentang izin alat. • Menambahkan spanduk yang dinonaktifkan pembelajaran saat menggunakan flag do-not-learn, memberikan visibilitas ke fitur pembelajaran dan cara mengaktifkannya. • Perubahan melanggar: Node.js 22 atau lebih baru sekarang diperlukan. Tingkatkan dari https://nodejs.org/ jika Anda menggunakan versi yang lebih lama.

Versi	Tanggal rilis	Catatan rilis
2.0.0	14 Mei 2026	• Menambahkan dukungan server MCP jarak jauh - sambungkan ke server HTTP dan HTTPS MCP dengan fallback SSE otomatis, otentikasi token Bearer dengan ekspansi variabel lingkungan di header, dan dukungan proxy (). HTTP_PROXY/HTTPS_PROXY • Ditambahkan `atx resume` subperintah sebagai alias untuk. `atx --resume` • `grep`Alat yang ditambahkan untuk pencarian teks multi-file dengan kontrol output - mendukung pola regex, pemfilteran gumpalan file, pencarian case-insensitive, batas yang dapat dikonfigurasi, dan pemotongan output. depth/result • Implementasi MCP yang ditingkatkan untuk menggunakan SDK MCP standar untuk perilaku yang lebih konsisten dan dukungan yang lebih mudah untuk fitur MCP masa depan. • Penanganan keluaran shell yang ditingkatkan - output sekarang terpotong dengan anggun pada 30K karakter per aliran alih-alih menyebabkan kesalahan keras, mempertah ankan awal dan akhir output untuk konteks. • Memperbaiki masalah di mana Ctrl+C selama sesi interaktif dapat keluar dari CLI sebelum waktunya alih-alih membatalk an operasi Linux/macOS saat ini dan kembali ke prompt. • Memperbaiki pesan penolakan kepercayaan untuk menunjukkan alasan khusus mengapa kepercayaan tidak tersedia, terpisah dari mengapa perintah memerlukan otorisasi. • Peringatan perintah berbahaya yang diperbaiki tidak lagi menyebutkan kepercayaan saat perintah masuk. `alwaysPromptCommands`

Versi	Tanggal rilis	Catatan rilis
1.10.0	April 29, 2026	• Deskripsi alat yang disempurnakan dengan panduan lintas alat, membantu agen memilih alat terbaik untuk setiap tugas dan mengurangi penggunaan shell yang tidak perlu. • Ditambahkan `alwaysPromptCommands` dalam `trust-settings.yaml` - konfigurasikan pola perintah shell yang selalu memerlukan izin eksplisit, terlepas dari pengaturan kepercayaan. Mendukung wildcard glob dan cocok dengan setiap sub-perintah dalam ekspresi majemuk. `trustedShellCommands` dapat mengesampingkan pola-pola ini. Tidak diberlakukan dalam mode non-interaktif untuk menghindari kerusakan jaringan pipa otomatis. • Jalur relatif tetap tidak diselesaikan menjadi jalur absolut.
1.9.0	April 23, 2026	• Permintaan persetujuan alat yang ditingkatkan - setiap opsi sekarang menunjukkan apa yang dilakukannya secara inline, pesan “Mengapa” menjelaskan mengapa persetuju an diperlukan, dan konfirmasi kepercayaan mencakup `trust-settings.yaml` petunjuk tentang kepercayaan permanen.
1.8.0	April 16, 2026	• Mengubah awalan keluaran agen dari `>` ke `ATX:` untuk perbedaan visual yang lebih jelas antara input pengguna dan output agen. • Memperbaiki tampilan perintah resume saat percobaan ulang CLI habis. • Resume tetap dari direktori kerja yang berbeda. Dalam mode interaktif, meminta jalur yang dikoreksi; dalam mode non-interaktif, gagal cepat dengan kesalahan deskriptif.
1.7.1	April 9, 2026	• Memperbaiki kegagalan login SSO (“Token sesi tidak ditemukan atau tidak valid”) ketika wilayah penyedia kredensi berbeda dari wilayah profil. `sso_region` • Caching kredenal tetap sehingga sesi yang berjalan lama tidak lagi gagal saat kredensional diputar pada disk.

Versi	Tanggal rilis	Catatan rilis
1.7.0	April 2, 2026	• `exec`Perintah tetap untuk keluar dengan kode 1 pada kesalahan validasi input, membuat CLI dapat diandalkan dalam skrip CI/CD dan saluran pipa. • Menambahkan tampilan pesan peringatan dari layanan. • Menambahkan dukungan MFA prompt untuk profil AWS kredensi. • Memperbaiki panggilan kredensi STS untuk menghormati pengaturan proxy HTTP (S).
1.6.0	Maret 18, 2026	• Ditambahkan `--limit <minutes>` pilihan untuk menetapkan anggaran Agent Minutes untuk `exec` dan `interactive` perintah. Ketika batas tercapai, CLI menampilkan penggunaan dan keluar dengan instruksi resume. • Pencatatan tetap dari pesan pengguna awal di log percakapan.
1.5.0	Maret 11, 2026	• Menambahkan dukungan untuk eksekusi tugas bersamaan, memungkinkan transformasi yang lebih cepat. • Menambahkan tampilan penggunaan Agent Minutes di akhir setiap percakapan, dan `/usage` perintah dalam mode interaktif untuk memeriksa penggunaan kapan saja.
1.4.0	Maret 2, 2026	• Pesan kesalahan yang ditingkatkan saat menyimpan definisi transformasi dengan file yang tidak didukung. • Menambahkan batas waktu ke panggilan layanan untuk mencegah CLI macet saat koneksi terputus. • Proses CLI tetap hang setelah berhasil diselesaikan. • Validasi wilayah tetap saat melanjutkan percakapan.

Versi	Tanggal rilis	Catatan rilis
1.3.0	Februari 19, 2026	• Pembatasan yang dihapus untuk jalur server MCP - pengguna sekarang dapat menentukan jalur apa pun untuk `command` properti dalam konfigurasi MCP. • Peningkatan stabilitas koneksi server MCP.
1.2.0	Februari 9, 2026	• Menambahkan dukungan untuk konfigurasi proxy melalui variabel lingkungan (`https_proxy` ,`no_proxy`, dll.). • Peningkatan stabilitas umum dan perbaikan bug.
1.1.1	Januari 16, 2026	Perbaikan bug dan peningkatan stabilitas.
1.1.0	Januari 13, 2026	Menambahkan dukungan konfigurasi wilayah AWS CLI, pemeriksaan validasi kredenal awal, dan fungsionalitas resume kampanye tetap.
1.0.1	Desember 18, 2025	Shutdown yang anggun, log kesalahan yang lebih bersih, dan perbaikan umum.
1.0.0	Desember 1, 2025	Rilis awal AWS Transform CLI.

AWS Mengubah modernisasi berkelanjutan

Apa itu AWS Mengubah modernisasi berkelanjutan?

AWS Modernisasi berkelanjutan Transform menyediakan analisis dan remediasi untuk repositori kode sumber Anda. Anda dapat menghubungkan GitHub organisasi, GitLab grup, ruang kerja Bitbucket, dan repositori lokal, lalu menjalankan analisis otomatis untuk mengidentifikasi utang teknis, kerentanan keamanan, peluang modernisasi, dan kesiapan agen di seluruh basis kode Anda.

Semua analisis dan remediasi berjalan di Anda Akun AWS menggunakan kredensil Anda. Kode sumber Anda tetap di bawah kendali Anda.

> **Note**
>
> Sebaiknya mulai dengan Kiro Power atau plugin agen. Keterampilan agen mengatur pengaturan, orientasi, dan penggunaan modernisasi berkelanjutan yang berkelanjutan termasuk penyediaan infrastruktur, konfigurasi sumber, eksekusi analisis, triase temuan, dan remediasi. Untuk instruksi instalasi, lihat Alat developer.

Kemampuan kunci

AWS Transform modernisasi berkelanjutan memberikan kemampuan berikut:

- Analisis Utang Teknologi - Pindai repositori untuk dependensi yang sudah ketinggalan zaman, kerentanan keamanan, masalah kualitas kode, dan peluang modernisasi. Jalankan pemindaian metadata cepat di seluruh manifes paket atau analisis tingkat kode yang komprehensif. Tentukan kriteria analisis khusus menggunakan definisi transformasi yang disesuaikan dengan lingkungan Anda.
- Remediasi Otonom — Hasilkan permintaan tarik yang divalidasi dalam skala besar. Setiap temuan dapat diperbaiki secara otomatis menggunakan definisi transformasi yang terkait. Remediasi membuat cabang dan terbuka PRs/MRs secara otomatis di seluruh GitHub, GitLab, dan Bitbucket.
- Pelaporan — Menghasilkan laporan HTML yang menunjukkan temuan berdasarkan tingkat keparahan, repositori, dan jenis analisis. Lacak kemajuan remediasi dan temukan resolusi dari waktu ke waktu.

- Pemantauan Berkelanjutan - Jadwalkan analisis berulang menggunakan Amazon EventBridge Scheduler. Konfigurasikan jadwal cron harian, mingguan, atau kustom untuk terus memantau portofolio Anda untuk masalah baru.

Jenis analisis

AWS Transform modernisasi berkelanjutan mendukung jenis analisis berikut untuk memenuhi kebutuhan modernisasi Anda.

Tipe	Deskripsi
`rapid-techdebt-analysis`	Pemindaian metadata-only cepat dari manifes paket (`pom.xml`,`package.json` ,`requirements.txt`) untuk mengidentifikasi versi basi dan dependensi yang sudah ketinggalan zaman. Tidak menganalisis kode sumber.
`tech-debt-comprehensive`	Analisis utang teknis tingkat kode yang mendalam menggunakan agen AWS Transform . Memeriksa kode sumber untuk mengidentifikasi pola utang, masalah kualitas kode, masalah arsitektur, dan peluang peningkatan.
`security`	Kerentanan keamanan dan deteksi CVE menggunakan Agen AWS Keamanan. Memindai kode sumber dan dependensi untuk kerentanan yang diketahui, pola pengkodean yang tidak aman, dan kelemahan yang dapat dieksploitasi. Membutuhkan pengaturan infrastruktur satu kali.
`agentic-readiness`	Penilaian kesiapan integrasi AI dan agen. Skor 56 kriteria di lima kategori: Infrastruktur & Platform, Arsitektur Aplikasi, Yayasan Data Identity/Security/Governance,, dan Operasi & Pengamatan.

Tipe	Deskripsi
`modernization-readiness`	Penilaian peluang modernisasi cloud. Mengevaluasi kesiapan di seluruh dimensi infrastruktur, aplikasi, data, keamanan, dan operasi. Mengidentifikasi kandidat untuk kontainerisasi, migrasi tanpa server, dan peningkatan platform.
`custom`	Jalankan definisi transformasi (TD) apa pun sebagai analisis. Gunakan tipe ini untuk menentukan kriteria analisis Anda sendiri atau untuk menjalankan transformasi AWS terkelola yang tidak tercakup oleh tipe bawaan.

Memahami konsep-konsep kunci

Sumber

Sebuah sumber memberi tahu modernisasi berkelanjutan di mana repositori Anda berada. Jenis sumber yang didukung:

- GitHub— Organizations dengan token akses pribadi (klasik)
- GitLab— Grup atau pengguna, termasuk instans yang dihosting sendiri
- Bitbucket - Ruang kerja (Cloud) atau proyek (Pusat Data)
- Lokal - Direktori induk yang berisi repositori git

Repositori

Repositori ditemukan dengan memindai sumber. Setelah ditemukan, Anda dapat memfilter berdasarkan sumber, label, atau kriteria lainnya; menerapkan label untuk organisasi (tim, prioritas, gelombang migrasi); dan menargetkan repositori khusus untuk analisis atau remediasi.

Analisis

Analisis adalah pemindaian satu atau lebih repositori menggunakan jenis analisis tertentu. Setiap analisis menghasilkan temuan yang mengidentifikasi masalah dalam kode Anda. Anda dapat menjalankan analisis sesuai permintaan atau menjadwalkannya untuk berjalan secara otomatis. Jenis analisis meliputi analisis hutang teknologi cepat, teknologi-utang-komprehensif, keamanan, kesiapan agen, kesiapan modernisasi, dan kebiasaan. Setiap analisis melacak statusnya (tertunda, berjalan, selesai, dibatalkan, atau gagal) dan repositori yang dipindai.

Temuan

Temuan adalah hasil analisis. Setiap temuan mencakup tingkat keparahan (tinggi, sedang, atau rendah), status (terbuka, diberhentikan, atau usang), dan transformasi perbaikan yang dapat memperbaikinya (jika dapat diperbaiki secara otomatis). Temuan baru dimulai sebagai terbuka. Pengguna dapat mengabaikan temuan dengan alasan. Re-analysis menandai temuan yang diselesaikan sebagai usang secara otomatis.

Remediasi

Remediasi menerapkan definisi transformasi untuk memperbaiki temuan. Tiga mode: berbasis temuan (setiap temuan menggunakan transformasi perbaikannya sendiri), penggantian TD (mengesampingkan definisi transformasi untuk temuan tertentu), dan TD langsung (jalankan transformasi terhadap repositori tanpa temuan). Output tergantung pada penyedia sumber (GitHub PR, GitLab MR, Bitbucket PR, atau cabang lokal).

Definisi transformasi

Definisi transformasi berisi instruksi dan pengetahuan yang diperlukan untuk melakukan transformasi kode tertentu. Modernisasi berkelanjutan menggunakan definisi transformasi untuk analisis kustom (`--type custom --transformation-name` *`name`*) dan remediasi (). `--transformation-name` *`name`* Buat daftar definisi transformasi yang tersedia dengan`atx custom def list`.

Bagaimana AWS Mengubah karya modernisasi berkelanjutan

AWS Modernisasi berkelanjutan transformasi biasanya digunakan dalam proyek skala besar di mana beberapa basis kode memerlukan analisis dan remediasi berkelanjutan. Tim biasanya mengikuti alur kerja ini:

1. Connect Sources — Tambahkan GitHub organisasi, GitLab grup, ruang kerja Bitbucket, atau direktori lokal sebagai sumber. Berikan token otentikasi dengan cakupan yang sesuai.
2. Temukan Repositori — Jalankan pemindaian penemuan untuk menghitung semua repositori di sumber Anda. Gunakan label untuk mengatur repositori berdasarkan tim, prioritas, atau gelombang migrasi.
3. Jalankan Analisis — Jalankan analisis di seluruh portofolio Anda. Pilih pemindaian cepat untuk hasil cepat atau analisis komprehensif untuk temuan terperinci. Analisis keamanan membutuhkan penyiapan infrastruktur satu kali.
4. Temuan Triase - Tinjau temuan berdasarkan tingkat keparahan, repositori, atau jenis analisis. Singkirkan positif palsu dengan alasan yang terdokumentasi. Re-runanalisis untuk menandai masalah yang diselesaikan sebagai usang.
5. Remediasi — Buat remediasi untuk memperbaiki temuan. Modernisasi berkelanjutan secara otomatis membuat cabang dan membuka pull/merge permintaan dengan perbaikan. Lacak status remediasi dan coba lagi kegagalan.
6. Siapkan Analisis Berkelanjutan — Jadwalkan analisis berulang menggunakan Amazon EventBridge Scheduler. Konfigurasikan irama analisis (harian, mingguan, atau cron khusus) untuk terus memantau portofolio Anda untuk masalah baru. Kombinasikan dengan remediasi otomatis untuk menjaga kesehatan kode dari waktu ke waktu.

Opsi Komputasi

Modernisasi berkelanjutan mendukung tiga opsi komputasi untuk menjalankan analisis dan remediasi. Keterampilan agen di Kiro Power dan plugin agen membantu Anda mengatur infrastruktur untuk setiap opsi.

> **Note**
>
> Terlepas dari opsi komputasi, semua analisis dan remediasi terjadi di Anda Akun AWS menggunakan kredensil Anda. Kode sumber Anda tetap di bawah kendali Anda.

Lokal (default)

Secara default, analisis berjalan di mesin lokal Anda. Opsi ini tidak memerlukan infrastruktur tambahan. Server berjalan secara lokal dan mengeksekusi analisis menggunakan sumber daya komputasi lokal. Baik untuk mencoba alat, repositori kecil, atau penggunaan individu.

Amazon EC2

Jalankan analisis pada instans Amazon EC2 persisten di instans Anda. Akun AWS Opsi ini melepaskan komputasi dari mesin lokal Anda, mendukung analisis yang lebih besar, dan memungkinkan analisis terjadwal berulang. Instance tetap berjalan di antara kiriman.

Keterampilan agen menyediakan AWS CloudFormation tumpukan yang mencakup:

- Instans Amazon EC2 (Amazon Linux 2023) dengan Docker dan wadah modernisasi berkelanjutan
- Peran IAM dengan izin untuk AWS Transform, Amazon S3, KMS AWS , Secrets Manager, dan agen keamanan
- `AmazonSSMManagedInstanceCore`untuk akses shell melalui SSM (tidak diperlukan key pair SSH atau port masuk)
- Grup keamanan tanpa aturan masuk

Pengguna atau peran IAM Anda memerlukan izin untuk manajemen siklus hidup Amazon EC2, operasi tumpukan AWS CloudFormation , pembuatan peran IAM, operasi bucket Amazon S3, manajemen rahasia Secrets Manager, dan perintah SSM.

Untuk mengatur eksekusi EC2, instal Kiro Power atau plugin agen (lihatAlat developer), lalu tanyakan kepada agen: “Siapkan instans EC2 untuk analisis modernisasi berkelanjutan”. Agen menyediakan infrastruktur, memverifikasi kontainer sehat, dan mengirimkan analisis Anda melalui SSM - tidak diperlukan SSH.

AWS Batch (Fargate)

Jalankan analisis sebagai pekerjaan terisolasi pada AWS Batch dengan Fargate untuk komputasi tanpa server. Setiap analisis berjalan dalam wadahnya sendiri tanpa mengelola infrastruktur yang persisten. Baik untuk analisis paralel dari berbagai sumber atau jenis analisis.

Opsi ini menggunakan kembali tumpukan infrastruktur AWS Transform CDK. Keterampilan agen menyebarkan infrastruktur yang diperlukan termasuk:

- AWS Antrian pekerjaan batch dan lingkungan komputasi
- Definisi Job dengan gambar kontainer modernisasi berkelanjutan
- Peran IAM untuk eksekusi pekerjaan batch
- Fungsi Lambda untuk pengajuan pekerjaan

Untuk mengatur eksekusi Batch, instal Kiro Power atau plugin agen (lihatAlat developer), lalu tanyakan kepada agen: “Jalankan analisis saya di Fargate” atau “Siapkan eksekusi Batch untuk modernisasi berkelanjutan”. Agen menyebarkan tumpukan CDK, menyimpan kredensi sumber Anda di Secrets Manager, dan mengirimkan pekerjaan analisis ke Batch. AWS

Pengaturan agen keamanan

Jenis `security` analisis menggunakan layanan Agen AWS Keamanan untuk kerentanan dan deteksi CVE. Tidak seperti jenis analisis lainnya, ini memerlukan pengaturan infrastruktur satu kali di Anda Akun AWS.

Perintah penyiapan menyediakan AWS CloudFormation tumpukan yang mencakup:

- Bucket Amazon S3 untuk mengunggah kode sumber
- Peran IAM yang diasumsikan oleh `securityagent.amazonaws.com`
- Kebijakan terkelola untuk peran agen keamanan

Pengguna atau peran IAM Anda harus memiliki izin tambahan berikut untuk menjalankan penyiapan:

- `cloudformation:CreateStack`, `cloudformation:UpdateStack`, `cloudformation:DescribeStacks`
- `iam:CreateRole`, `iam:PutRolePolicy`, `iam:AttachRolePolicy`, `iam:CreatePolicy`
- `s3:CreateBucket`, `s3:PutBucketEncryption`, `s3:PutBucketPublicAccessBlock`

Gunakan perintah berikut untuk mengelola infrastruktur agen keamanan:

```
# Provision the security agent infrastructure
atx ct setup security-agent

# Check the status
atx ct setup security-agent --status

# Delete the infrastructure
atx ct setup security-agent --delete
```

Setelah penyiapan selesai, Anda dapat menjalankan analisis keamanan dengan cara yang sama seperti jenis analisis lainnya.

Memulai

Prasyarat

Untuk menggunakan modernisasi berkelanjutan secara efektif, Anda harus memiliki pemahaman dasar tentang kontrol versi Git, keakraban dengan platform kode sumber Anda (GitHub,, atau Bitbucket) GitLab, pengetahuan kerja tentang izin AWS IAM, dan pengalaman dengan alat baris perintah.

Sebelum Anda menggunakan modernisasi berkelanjutan, verifikasi bahwa Anda memiliki yang berikut:

- AWS Transform CLI diinstal:

```
curl -fsSL https://transform-cli.awsstatic.com/install.sh | bash
```

- Node.js 22 atau yang lebih baru
- AWS Kredensi yang valid (`AWS_PROFILE`atau variabel lingkungan)
- Kebijakan AWS terkelola yang `AWSTransformCustomFullAccess` dilampirkan pada pengguna atau peran IAM Anda

Quick start

1. Mulai server modernisasi berkelanjutan:

   ```
   atx ct server &
   ```

2. Tambahkan sumber:

   ```
   atx ct source add --name name --provider provider --org org-or-group --token token
   ```

   Di mana *`name`* pengenal deskriptif untuk sumber Anda, *`provider`* adalah platform (`github`,,`gitlab`, atau`local`)`bitbucket`, *`org-or-group`* adalah nama organisasi atau grup Anda, dan *`token`* merupakan token otentikasi Anda.

3. Temukan repositori:

   ```
   atx ct discovery scan --source name
   ```

4. Jalankan analisis:

```
atx ct analysis run --type type --source name --wait
```

5. Jelajahi temuan:

```
atx ct findings list --json
```

6. Remediasi temuan:

```
atx ct remediation create --ids id1,id2 --name "remediation-name"
```

Bekerja dengan modernisasi berkelanjutan

Manajemen sumber

Gunakan `atx ct source` perintah untuk menghubungkan repositori. Penyedia yang didukung: GitHub, GitLab, Bitbucket, lokal.

GitHub organisasi

Token: token akses pribadi (klasik) dengan `repo` ruang lingkup. Read-only untuk analisis, repo lengkap untuk remediasi.

```
atx ct source add --name name --provider github --org org --token pat
```

GitLab grup dan pengguna

Token: token akses pribadi dengan `api` ruang lingkup.

```
atx ct source add --name name --provider gitlab --org group-or-user --token pat
# Self-hosted:
atx ct source add --name name --provider gitlab --org group-or-user --token pat --url
 https://gitlab.example.com
```

Ruang kerja dan proyek Bitbucket

Bitbucket Cloud — cakupan:`read:repository:bitbucket`,,`write:repository:bitbucket`,`read:pullrequest:bitb` `write:pullrequest:bitbucket` Juga kebutuhan `--email` dan`--username`.

```
atx ct source add --name name --provider bitbucket --org workspace --token api-token --
email email --username username
```

Pusat Data Bitbucket:

```
atx ct source add --name name --provider bitbucket --org project-key --token http-
access-token --url https://bitbucket.example.com
```

Repositori lokal

```
atx ct source add --name name --provider local --path parent-directory
```

> **Important**
>
> `--path`harus menunjuk ke direktori induk yang berisi repo git sebagai subdirektori, bukan ke repo tunggal.

Mengelola sumber

```
atx ct source list
atx ct source remove --name name
```

Penemuan dan manajemen repositori

```
atx ct discovery scan --source name
atx ct discovery status --source name
atx ct discovery scan --source name --path new-directory
```

Setelah penemuan:

```
atx ct repository list
atx ct repository list --source name
atx ct repository list --labels "team:frontend,priority:high"
atx ct repository update --source name --repo "source::slug" --labels
 "team:frontend,priority:high"
atx ct repository update --source name --labels "migration:wave-1"
```

Menjalankan analisis

`--type`Bendera menentukan jenis analisis yang akan dijalankan:

- `rapid-techdebt-analysis`— Ketergantungan usang dan kemenangan mudah.
- `tech-debt-comprehensive`— AI-powered Analisis yang lebih dalam mencakup ketergantungan, keamanan, pola, kinerja, pemeliharaan, arsitektur, kualitas kode, dan temuan infrastruktur.
- `security`— Kerentanan dan eksposur keamanan.
- `agentic-readiness`— Kesiapan repositori Anda untuk agen AI (kerangka kerja, API, dokumentasi).
- `modernization-readiness`— Peluang modernisasi di seluruh dimensi infrastruktur, aplikasi, data, keamanan, dan operasi Anda.

```
atx ct analysis run --type type --source name [--repo source::slug] [--wait]
atx ct analysis get --id id --json
atx ct analysis list --json
atx ct analysis list --status pending|running|complete|cancelled|failed --json
atx ct analysis list --type type --json
atx ct analysis cancel --id id
atx ct analysis delete --id id [--cascade-findings]
```

Analisis kustom

```
atx ct analysis run --type custom --transformation-name name --source source --
repo source::slug --wait
```

Konfigurasi dengan `-g` bendera: nilai kunci, JSON, atau jalur file.

Daftar TD: `atx custom def list`

Mengelola temuan

```
atx ct findings list --json
atx ct findings list --repo source::slug --source name --severity high|medium|low
 --type analysis-type --status open|dismissed|obsolete --analysis-id id --fix-
transform transform-name --json
```

Menemukan status

- `open`— Aktif
- `dismissed`— Dihentikan secara manual (membutuhkan alasan)
- `obsolete`— System-set ketika analisis ulang tidak lagi menghasilkan temuan

```
atx ct findings update --id id --status dismissed --reason "reason"
atx ct findings update --id id --status open
atx ct findings batch-update --ids id1,id2 --status dismissed --reason "reason"
atx ct findings get --id id
atx ct findings delete --id id
```

Menemukan keusangan

Re-analysis menandai temuan yang diselesaikan sebagai usang. Tidak dapat dibuka kembali. Dipertahankan untuk audit.

Membuat remediasi

Tiga mode: berbasis temuan, penggantian TD, TD langsung.

```
atx ct remediation create --ids id1,id2 --name "name"
atx ct remediation create --ids id1,id2 --transformation-name TD
atx ct remediation create --transformation-name TD --repo source::slug
```

Keluaran oleh penyedia: GitHub PR, GitLab MR, Bitbucket PR, Cabang lokal.

> **Note**
>
> Token harus memiliki akses tulis untuk PR/MR pembuatan.

Eksekusi lokal dengan `--local` bendera.

```
atx ct remediation create --transformation-name TD --repo source::slug -g
 "additionalPlanContext=Upgrade to Node.js 22"
atx ct remediation list
atx ct remediation status --id id
atx ct remediation retry --id id
atx ct remediation delete --id id
```

AWS Transform aplikasi web (Opsional)

Aplikasi web AWS Transform adalah antarmuka opsional untuk memantau analisis modernisasi berkelanjutan, temuan, dan remediasi di seluruh portofolio sumber kode Anda.

Sebelum menggunakan aplikasi web AWS Transform, Anda harus mengaktifkan identitas pengguna untuk mengakses AWS Transform di organisasi Anda. Untuk informasi tentang mengaktifkan AWS Transform, lihat Menyiapkan AWS Transform.

Untuk menggunakan aplikasi web AWS Transform:

1. Kunjungi `https://aws.amazon.com/transform/` dan masuk menggunakan kredensil Pusat AWS Identitas IAM.
2. Jika modernisasi berkelanjutan tidak muncul setelah Anda masuk, masuk dengan kredensi IAM sebagai gantinya. Untuk mengaktifkan login kredenal IAM:
   a. Di AWS Management Console, buka AWS Transform dan pilih Settings.
   b. Di bagian Access AWS Transform with IAM credentials, aktifkan akses kredenal IAM.
   c. Pada halaman pengaturan yang sama, salin tautan masuk yang ditampilkan di bawah URL aplikasi Web (dengan IAM).
   d. Rekatkan URL masuk ke jendela browser yang sama tempat Konsol AWS Manajemen terbuka. Ini memastikan aplikasi web menggunakan AWS kredensil dari akun itu.
3. Buka menu navigasi kiri dan pilih modernisasi berkelanjutan. Dasbor menampilkan statistik ringkasan termasuk sumber, repositori, temuan total berdasarkan tingkat keparahan, dan jenis analisis.
4. Gunakan tab Analisis, Temuan, dan Remediasi untuk melihat hasil terperinci.
5. Mengobrol dengan AWS Transform langsung dari aplikasi web untuk mengajukan pertanyaan tentang analisis, temuan, atau remediasi Anda.

Aplikasi web dirancang untuk operasi skala perusahaan di mana Anda memerlukan visibilitas terpusat ke dalam analisis modernisasi berkelanjutan dan remediasi di beberapa basis kode.

Alat developer

Anda dapat mengakses modernisasi berkelanjutan dari IDE agen menggunakan AWS Transform Power dan plugin agen. Agen mengatur pengaturan, orientasi, dan penggunaan berkelanjutan modernisasi berkelanjutan melalui interaksi percakapan.

- Kiro IDE — Instal AWS Transform Kiro Power. Buka Kiro Chat dan minta agen untuk membantu alur kerja modernisasi berkelanjutan.
- Claude Code, Codex, Cursor — Instal awslabs/agent-plugins (direktori). `plugins/aws-transform` Plugin agen menyediakan kemampuan yang sama dengan Kiro Power.

Keterampilan agen

Plugin Kiro Power dan agen mencakup keterampilan yang mengatur alur kerja modernisasi berkelanjutan melalui interaksi percakapan:

Keterampilan	Deskripsi
Panduan	Orientasi interaktif yang memandu Anda melalui alur kerja penuh langkah demi langkah
Sumber	Tambahkan, daftar, dan hapus koneksi sumber
Penemuan	Pindai sumber untuk menemukan repositori
Analisis	Jalankan dan kelola analisis di semua jenis
Temuan	Kueri, filter, dan kelola temuan analisis
Remediasi	Membuat dan mengelola remediasi untuk memperbaiki temuan
Eksekusi EC2	Menyediakan dan mengelola instans Amazon EC2 untuk analisis jarak jauh

Keterampilan	Deskripsi
Eksekusi Batch	Kirim pekerjaan analisis ke AWS Batch dengan Fargate
Jadwal	Siapkan analisis berulang pada jadwal cron menggunakan Amazon Scheduler EventBridge
Pelaporan	Hasilkan laporan HTML mandiri dengan bagan yang menunjukkan temuan, distribusi keparahan, dan kemajuan remediasi

Referensi CLI

Tabel berikut merangkum semua `atx ct` subperintah.

Subperintah	Deskripsi
`atx ct server`	Mulai server modernisasi berkelanjutan
`atx ct status`	Lihat status sistem termasuk sumber, repositori, analisis, temuan, dan remediasi
`atx ct source add`	Tambahkan sumber (GitHub, GitLab, Bitbucket, atau lokal)
`atx ct source list`	Daftar sumber yang dikonfigurasi
`atx ct source remove`	Hapus sumber
`atx ct discovery scan`	Temukan repositori dari sumber
`atx ct discovery status`	Periksa status pemindaian penemuan
`atx ct repository list`	Daftar repositori dengan filter opsional
`atx ct repository get`	Dapatkan satu repositori
`atx ct repository update`	Perbarui label repositori

Subperintah	Deskripsi
`atx ct repository delete`	Hapus sebuah repositori
`atx ct analysis run`	Jalankan analisis pada repositori
`atx ct analysis get`	Dapatkan detail analisis
`atx ct analysis list`	Daftar analisis dengan status opsional dan jenis filter
`atx ct analysis cancel`	Batalkan analisis yang sedang berjalan
`atx ct analysis delete`	Hapus analisis
`atx ct findings list`	Buat daftar temuan dengan filter untuk repo, sumber, tingkat keparahan, jenis, status, dan analisis
`atx ct findings get`	Dapatkan satu temuan
`atx ct findings update`	Perbarui status pencarian (terbuka atau diberhentikan)
`atx ct findings batch-update`	Batch memperbarui beberapa temuan
`atx ct findings delete`	Hapus temuan yang diberhentikan atau usang
`atx ct remediation create`	Buat remediasi dari temuan atau definisi transformasi
`atx ct remediation list`	Daftar semua remediasi
`atx ct remediation status`	Periksa status remediasi dan lihat tautan PR/MR
`atx ct remediation retry`	Coba lagi remediasi yang gagal
`atx ct remediation delete`	Hapus remediasi

Subperintah	Deskripsi
`atx ct setup security-agent`	Menyediakan atau mengelola infrastruktur agen keamanan
`atx ct schema`	Cetak manifes JSON yang dapat dibaca mesin dari permukaan `atx ct` perintah penuh — setiap perintah, opsi, dan argumen — sehingga agen dan otomatisasi dapat menemukan CLI tanpa menguraikan teks bantuan

Pemecahan Masalah

Koneksi ditolak atau server tidak berjalan

CLI modernisasi berkelanjutan membutuhkan server yang berjalan. Mulailah dengan `atx ct server &` dan verifikasi dengan`atx ct status --health`.

Pemindaian penemuan mengembalikan nol repositori

Untuk sumber lokal, verifikasi bahwa `--path` menunjuk ke direktori induk yang berisi repositori sebagai subdirektori (misalnya,`/home/user/repos`), bukan ke repositori secara langsung (misalnya,). `/home/user/repos/my-app` Pemindai mencari direktori anak yang berisi `.git` folder.

Kesalahan SETUP_REQUIRED

Sumber ada di akun Anda tetapi kredensialnya tidak dikonfigurasi pada mesin ini. Jalankan `atx ct source add --name` *`name`* untuk mengonfigurasi kredensil secara lokal.

Kesalahan AUTH_REQUIRED

Tidak ada token valid yang ditemukan untuk sumbernya. Verifikasi token Anda benar dan memiliki cakupan yang diperlukan. Untuk GitHub, pastikan token memiliki `repo` ruang lingkup. Untuk GitLab, pastikan `api` ruang lingkup.

Kesalahan INVALID_INPUT

Pastikan Anda menggunakan nama jenis analisis yang benar. Nilai yang valid adalah:`rapid-techdebt-analysis`,`tech-debt-comprehensive`,`security`,`agentic-readiness`,`modernization-readiness`,`custom`.

Kesalahan izin selama remediasi

Remediasi menciptakan cabang dan pull/merge permintaan, yang membutuhkan akses tulis ke repositori. Perbarui token Anda dengan izin menulis. Untuk GitHub, pastikan `repo` ruang lingkup penuh. Untuk GitLab, pastikan `api` ruang lingkup. Untuk Bitbucket, pastikan `write:repository:bitbucket` dan `write:pullrequest:bitbucket` cakupan.

Pengaturan agen keamanan gagal

Verifikasi bahwa pengguna atau peran IAM Anda memiliki izin yang diperlukan AWS CloudFormation, IAM, dan Amazon S3 yang tercantum di bagian penyiapan agen keamanan. Bagaimana AWS Mengubah karya modernisasi berkelanjutan Periksa statusnya dengan`atx ct setup security-agent --status`.

Modernisasi berkelanjutan tidak terlihat di aplikasi web

Jika modernisasi berkelanjutan tidak muncul setelah Anda masuk ke aplikasi web AWS Transform, masuk dengan kredensi IAM di Akun AWS mana AWS Transform diaktifkan, bukan Pusat Identitas IAM. AWS Untuk langkah, lihat AWS Transform aplikasi web (Opsional).

Alat penemuan

Alat AWS Transform penemuan membantu Anda secara otomatis menemukan inventaris server di organisasi Anda untuk mempersiapkan migrasi. Alat ini mendukung VMware vCenter, Hyper-V Microsoft, dan server yang diimpor. Untuk menggunakan alat penemuan, Anda mengonfigurasi satu atau beberapa sumber penemuan, biarkan alat berjalan, lalu tinjau hasilnya di panel inventaris Ditemukan.

Setelah Anda mengonfigurasi sumber penemuan, alat penemuan mulai mengumpulkan informasi. Waktu yang dibutuhkan alat penemuan untuk sepenuhnya menganalisis jaringan Anda tergantung pada ukuran lingkungan Anda. Untuk kasus bisnis migrasi terarah, Anda dapat menggunakan file Penilaian Portofolio Migrasi (MPA) yang dihasilkan alat setelah pengumpulan server selesai.

Alur kerja alat penemuan

Alur kerja alat penemuan terdiri dari dua jenis kegiatan:

- Kegiatan konfigurasi
- Tinjauan dan penggunaan data

Langkah-langkah berikut menjelaskan cara menginstal dan menggunakan alat penemuan dan bekerja dengan data yang dikumpulkan:

1. Instal alat penemuan.
2. Konfigurasikan sumber penemuan. Anda dapat mengonfigurasi satu atau beberapa hal berikut: VMware vCenter, Hyper-V Microsoft host, atau server melalui impor file CSV. Penemuan data dimulai setelah Anda mengonfigurasi sumber apa pun.
3. Siapkan akses OS dan kemudian tinjau status pengumpulan server, database, dan koneksi jaringan.
    - Sesuaikan kredensyal OS sesuai kebutuhan.
4. (Opsional) Konfigurasikan kredensyal Oracle untuk mengaktifkan pengumpulan SQL langsung, atau gunakan OS-level deteksi fallback melalui SSH atau WinRM.
5. Untuk menghasilkan kasus bisnis migrasi, unggah file.zip ke penilaian Migrasi, atau unzip dan unggah file MPA dari direktori mpa_exports. Ekspor mencakup data dari semua sumber yang dikonfigurasi dan berisi file MPA untuk VMware, Hyper-V, dan server yang diimpor.

Alat penemuan mendukung jalur penemuan berikut:

- VMware vCenter auto-discovery - Secara otomatis menemukan server yang dikelola oleh VMware vCenter.
- Hyper-V penemuan otomatis - Secara otomatis menemukan server yang dikelola oleh Hyper-V host Microsoft.
- Impor server — Mengimpor inventaris server secara manual melalui file CSV.

Menyiapkan alat penemuan

Prasyarat

Berikut ini adalah prasyarat untuk menggunakan alat penemuan: AWS Transform

Prasyarat umum

- Alat ini membutuhkan 4 vCPU, 16 GB RAM, dan 35 GB hard disk.
- DHCP harus tersedia di jaringan untuk alat penemuan VM.
- Alat ini mengumpulkan data dengan menggunakan pendekatan terpusat. Server dalam lingkup harus mengizinkan konektivitas masuk dari alat penemuan VM (port default, konfigurasi port khusus didukung):
  - Linux — SSH TCP/22
  - Windows - TCP/5985 untuk HTTP, TCP/5986 untuk HTTPS
  - SNMP — UDP/161 (hanya digunakan untuk pengumpulan jaringan, bukan metrik OS)
- Untuk Linux, akun pengguna yang dapat menggunakan SSH untuk terhubung ke server. Alat penemuan menjalankan perintah melalui SSH untuk pengumpulan jaringan dan metrik OS. Sebagian besar perintah berjalan sebagai pengguna biasa. Sejumlah kecil perintah mencoba sudo dan mundur secara otomatis jika sudo tidak tersedia: `dmidecode` (server UUID dan pabrikan), (deteksi LVM), dan `ss` atau `lvdisplay` `netstat` (koneksi jaringan dengan informasi proses). Tanpa sudo, alat penemuan masih mengumpulkan sebagian besar data tetapi UUID, LVM, dan detail jaringan tingkat proses akan hilang. Kami merekomendasikan untuk mengonfigurasi sudo tanpa kata sandi untuk pengguna SSH untuk memastikan pengumpulan data lengkap. Untuk rincian lengkap, lihatthe section called “Izin yang diperlukan”.

Prasyarat VMware

- VMware vCenter Server versi 6.5, 6.7, 7.0, atau 8.0.
- Izin untuk menyebarkan OVA ke VMware vCenter Anda.
- Untuk pengaturan VMware vCenter Server, pengguna vCenter dengan Read-Only peran yang ditetapkan pada tingkat pusat data root. Lihat perinciannya di the section called “Izin yang diperlukan”.

Hyper-V prasyarat

- Windows Server dengan Hyper-V peran diaktifkan.
- WinRM diaktifkan pada Hyper-V host.
- Akun pengguna yang merupakan anggota grup Pengguna Manajemen Jarak Jauh, Hyper-V Administrator, dan Pengguna Monitor Kinerja di setiap Hyper-V host, dengan akses baca WMI ke namespace. `root\cimv2` Lihat perinciannya di the section called “Izin yang diperlukan”.
- Otentikasi yang didukung: NTLM (hanya HTTPS) dan Kerberos (HTTP atau HTTPS).

Prasyarat impor server

- File CSV dengan nama host server atau alamat IP dan nama kredensi (opsional) yang dipetakan ke nama ramah kredenal OS yang dikonfigurasi atau untuk dikonfigurasi pada alat penemuan. CSV harus menggunakan header berikut:

```
hostname_or_ip,os_credential_name,oracle_credential_name
```

- Server harus dapat dijangkau dari alat penemuan VM. Port default: Port SSH 22 untuk Linux, 5985/5986 port WinRM untuk Windows. Port khusus didukung.

Prasyarat penginstal Linux

- Distribusi Linux yang didukung: Amazon Linux 2, Amazon Linux 2023, RHEL 8-9, Rocky Linux 8—9, 9, Ubuntu 20.04—24.04, Debian AlmaLinux 10—12, atau SLES 15 SP5.
- Minimal 4 vCPU, 16 GB RAM, 35 GB ruang disk yang tersedia.
- Port 5000 harus tersedia (tidak digunakan oleh layanan lain).
- systemd diperlukan untuk manajemen layanan.

- Alat penemuan VM harus memiliki akses jaringan ke infrastruktur target Anda. Port default: vCenter pada port 443, Hyper-V host pada port 5985/5986, server pada port 22. Port khusus didukung untuk SSH, WinRM, dan SNMP.

Prasyarat Oracle Database

- Akses jaringan dari alat penemuan ke host Oracle pada port 1521 (atau port khusus).
- Versi Oracle yang didukung: Anda dapat mengumpulkan Oracle Database 12c Release 1 (12.1) dan yang lebih baru melalui koneksi SQL langsung. OS-level deteksi fallback berfungsi dengan semua versi Oracle.
- Akun layanan Oracle read-only dengan hibah SELECT_CATALOG_ROLE. Lihat perinciannya di the section called “Database Oracle (SQL)”.
- Untuk deteksi OS-level fallback: Akses SSH atau WinRM ke host Oracle (menggunakan kredensyal OS yang ada).

Menyebarkan di VMware

Spesifikasi mesin virtual VMware

- Sistem Operasi - Amazon Linux 2023
- RAM - 16 GB
- CPU - 4 core
- Disk - 35 GB
- Persyaratan VMware - Lihat persyaratan host VMware untuk menjalankan AL2023 di VMware

Menyebarkan VMware OVA

1. Unduh file OVA dari URL ini: https://s3.us-east-1.amazonaws.com/atx.discovery.collector.bundle/releases/latest/AWS-Transform-discovery-tool.ova
2. Masuk ke vCenter sebagai administrator VMware.
3. Gunakan salah satu cara ini untuk menginstal file OVA:

    a. Gunakan UI: Pilih File, pilih Deploy OVF Template, pilih alat penemuan file OVA yang Anda unduh di langkah 1, lalu selesaikan wizard. Pastikan pengaturan proxy di dasbor manajemen server dikonfigurasi dengan benar.

b. Gunakan baris perintah: Untuk menginstal alat penemuan file OVA dari baris perintah, unduh dan gunakan VMware Open Virtualization Format Tool (ovftool). Untuk mengunduh ovftool, pilih rilis dari halaman Dokumentasi Alat OVF. Ini adalah contoh menggunakan alat baris perintah ovftool untuk menginstal file OVA alat penemuan.

```
ovftool --acceptAllEulas --name='discovery tool' --datastore=datastore1 -
dm=thin AWS-Transform-discovery-tool.ova 'vi://username:password@vcenterurl/
Datacenter/host/esxi/'
```

Deskripsi nilai yang dapat diganti dalam contoh:

- Nama adalah nama yang ingin Anda gunakan untuk alat penemuan Anda VM.
- Datastore adalah nama datastore di vCenter Anda.
- Nama file OVA adalah nama file OVA alat penemuan yang diunduh.
- Itu username/password adalah kredensyal vCenter Anda.
- Vcenterurl adalah URL vCenter Anda.
- Jalur vi adalah jalur ke host VMware ESXi Anda.

4. Temukan alat penemuan yang digunakan di vCenter Anda. Right-click VM, dan kemudian pilih Power, Power On.
5. Setelah beberapa menit, alamat IP alat penemuan ditampilkan di vCenter. Anda menggunakan alamat IP ini untuk terhubung ke alat penemuan.

Menyebarkan pada Hyper-V

Hyper-V spesifikasi mesin virtual

- Sistem Operasi - Amazon Linux 2023
- RAM - Kami merekomendasikan mengalokasikan setidaknya 16 GB
- CPU - Kami merekomendasikan mengalokasikan setidaknya 4 core
- Disk — 35 GB (termasuk dalam VHD)
- Hyper-V persyaratan - Lihat persyaratan Hyper-V host untuk menjalankan AL2023 di Hyper-V

Menyebarkan VHD Hyper-V

1. Unduh file VHD dari URL ini: https://s3.us-east-1.amazonaws.com/atx.discovery.collector.bundle/releases/latest/AWS-Transform-discovery-tool.vhd
2. Salin file VHD ke mesin Windows Server yang memiliki Hyper-V peran diaktifkan.
3. Buka Hyper-V Manajer.
4. Pilih Baru, lalu pilih Virtual Machine.
5. Selesaikan wizard pengaturan. Pada halaman Tentukan Generasi, pilih Generasi 1. Mesin virtual generasi 2 tidak mendukung format VHD. Pada halaman Tetapkan Memori, alokasikan setidaknya 16384 MB. Pada halaman Connect Virtual Hard Disk, pilih Gunakan hard disk virtual yang ada dan pilih file VHD yang Anda salin.
6. Mulai VM. Setelah beberapa menit, periksa tab Jaringan VM di Hyper-V Manajer untuk menemukan alamat IP, atau sambungkan ke konsol VM dan jalankan. `ip addr` Anda menggunakan alamat IP ini untuk terhubung ke alat penemuan.

Menyebarkan di Linux

Alat penemuan tersedia sebagai installer Linux mandiri untuk penyebaran pada distribusi Linux yang didukung. Gunakan opsi ini ketika Anda ingin menginstal alat penemuan langsung di server Linux yang ada alih-alih menggunakan mesin virtual.

Spesifikasi server Linux

- RAM - Kami merekomendasikan mengalokasikan setidaknya 16 GB
- CPU - Kami merekomendasikan mengalokasikan setidaknya 4 core
- Disk - 35 GB

Distribusi yang didukung

- Amazon Linux - Amazon Linux 2, Amazon Linux 2023
- Red Hat Enterprise Linux dan turunannya - RHEL 8-9, Rocky Linux 8-9, 9 AlmaLinux
- Ubuntu - 20.04, 22.04, 24.04
- Debian — 10, 11, 12
- SUSE — SLES 15 SP5

Instal alat penemuan

Unduh skrip penginstal dari URL ini: https://s3.us-east-1.amazonaws.com/atx.discovery.collector.bundle/releases/latest/AWS-Transform-discovery-tool.sh

Untuk menginstal alat penemuan di Linux

1. Unduh skrip penginstal ke server Linux Anda.

```
curl -O https://s3.us-east-1.amazonaws.com/atx.discovery.collector.bundle/releases/latest/AWS-Transform-discovery-tool.sh
chmod +x AWS-Transform-discovery-tool.sh
```

2. (Opsional) Jalankan pemeriksaan kompatibilitas untuk memverifikasi bahwa sistem Anda memenuhi persyaratan.

```
sudo ./AWS-Transform-discovery-tool.sh check
```

3. Instal alat penemuan. Pilih salah satu opsi berikut:

   Opsi 1: Instal dengan instalasi ketergantungan otomatis (disarankan)

   Opsi ini secara otomatis menginstal dependensi sistem yang hilang sebelum menginstal layanan.

```
sudo ./AWS-Transform-discovery-tool.sh install --install-deps
```

   Opsi 2: Instal tanpa instalasi ketergantungan otomatis

   Opsi ini memeriksa dependensi yang hilang dan menampilkan kesalahan dengan instruksi pemasangan manual jika dependensi kritis tidak ada. Gunakan opsi ini jika organisasi Anda membatasi penginstalan paket otomatis dan Anda lebih memilih untuk mengelola paket sistem secara manual.

```
sudo ./AWS-Transform-discovery-tool.sh install
```

4. Mulai layanan alat penemuan.

```
sudo ./AWS-Transform-discovery-tool.sh start
```

5. Akses alat penemuan di`https://`*`ip_address`*`:5000`.

Untuk menghapus instalasi alat penemuan, jalankan`sudo ./AWS-Transform-discovery-tool.sh uninstall`.

Anda juga dapat menggunakan perintah berikut untuk mengelola layanan:

- `sudo ./AWS-Transform-discovery-tool.sh status`— Periksa apakah layanan sedang berjalan.
- `sudo ./AWS-Transform-discovery-tool.sh stop`— Hentikan layanan.
- `sudo ./AWS-Transform-discovery-tool.sh logs`— Lihat log layanan.

Ketergantungan sistem

Pemasang memeriksa dependensi sistem berikut:

Dependensi	Kepelikan	Tujuan
OpenSSL	Kritis	Menghasilkan kunci enkripsi basis data
libcrypt	Kritis	Diperlukan oleh runtime Python
Perpustakaan Kerberos	Peringatan	Diperlukan untuk otentikasi Kerberos ke server Windows. Tanpa ini, NTLM dan otentikasi dasar masih berfungsi.

Dependensi kritis diperlukan untuk memulai alat penemuan. Jika hilang dan Anda memilih Opsi 2 (tanpa`--install-deps`), penginstal menampilkan kesalahan dengan instruksi untuk menginstalnya secara manual. Warning-level dependensi bersifat opsional - penginstal memperingatkan Anda tetapi melanjutkan dengan instalasi.

> **Note**
>
> Pemasang memeriksa kompatibilitas sistem termasuk versi glibc dan ketersediaan systemd. Pada sistem dengan systemd 250 atau yang lebih baru (Amazon Linux 2023, RHEL 9, Rocky 9, 9, Ubuntu 24.04+, AlmaLinux Debian 12+), kunci enkripsi database dienkripsi saat istirahat menggunakan systemd-creds. Pada sistem yang lebih lama, kunci enkripsi disimpan sebagai file yang dilindungi izin.

Konfigurasikan alat penemuan

Akses konsol alat penemuan

1. Dalam akses browser web:`https://`*`ip_address`*`:5000`, di *`ip_address`* mana alamat IP alat penemuan dari Deploy Discovery Tool. Alat penemuan menggunakan sertifikat yang ditandatangani sendiri untuk koneksi HTTPS yang menghasilkan peringatan keamanan. Pilih Terima risiko dan lanjutkan ke konsol alat penemuan.
2. Jika Anda mengakses konsol alat penemuan untuk pertama kalinya, buat kata sandi login alat penemuan. Buat kata sandi, yang akan Anda gunakan untuk login di masa mendatang.

> **Important**
>
> Ingat kata sandi ini - tidak ada mekanisme pemulihan kata sandi.

Mengakses alat penemuan VM

- Alat penemuan VM datang secara default dengan nama pengguna dan kata sandi (“penemuan”, “kata sandi”). Untuk keamanan yang kuat, kami sarankan Anda memperbarui kata sandi dengan menggunakan `sudo passwd discovery` setelah masuk ke VM melalui konsol hypervisor Anda (misalnya, VSphere Client for VMware atau Manager for). Hyper-V Hyper-V
- Akses SSH dinonaktifkan secara default. Pengguna dapat menggunakan pra-konfigurasi `enablessh` dan `disablessh` alias untuk akses enable/disable SSH ke alat penemuan VM. Pengguna dapat SSH ke VM melalui `ssh discovery@<VM-IP>` setelah mengaktifkan akses SSH. Pengguna didorong untuk menjaga akses SSH dinonaktifkan sebagian besar waktu dan mengaktifkannya hanya saat diperlukan secara aktif. Perubahan kata sandi diberlakukan saat menjalankan`enablessh`.
- Untuk mengakses direktori data alat penemuan di`/home/ec2-user/.local/share/DiscoveryTool`, kami sarankan beralih ke `ec2-user` dengan menjalankan`sudo su ec2-user`.

mengonfigurasi autentikasi Kerberos

Otentikasi Kerberos adalah metode yang disarankan untuk menghubungkan ke server Windows dari alat penemuan. Alat penemuan VM menggunakan pustaka Amazon Linux 2023 Kerberos asli untuk mengautentikasi terhadap domain Direktori Aktif Anda.

Berikut ini adalah poin-poin penting tentang otentikasi Kerberos pada alat penemuan VM:

- Gunakan `kinit` perintah untuk mendapatkan tiket Kerberos dan `klist` untuk memverifikasi tiket.
- File konfigurasi Kerberos terletak di. `/etc/krb5.conf`
- Sebelum Anda mengonfigurasi alat penemuan, verifikasi bahwa `kinit` berhasil dari CLI pada alat penemuan VM.

Prasyarat Kerberos

Sebelum Anda mengonfigurasi otentikasi Kerberos, verifikasi bahwa Anda memiliki informasi dan konektivitas jaringan berikut.

1. Dapatkan informasi berikut dari administrator Direktori Aktif Anda:
   - Nama ranah Kerberos (biasanya nama domain Anda dalam huruf besar, misalnya,). `EXAMPLE.COM`
   - Nama host atau alamat IP dari Key Distribution Center (KDC), yang biasanya merupakan pengontrol domain (misalnya,`dc01.example.com`).
   - Akun layanan dengan izin untuk mengautentikasi terhadap server Windows target.
2. Verifikasi bahwa alat penemuan VM memiliki konektivitas jaringan ke yang berikut:
   - KDC pada port 88 (TCP dan UDP) untuk otentikasi Kerberos.
   - Target server Windows pada port WinRM (5985 untuk HTTP, 5986 untuk HTTPS).

Konfigurasikan Kerberos

Selesaikan langkah-langkah berikut untuk mengonfigurasi otentikasi Kerberos pada alat penemuan VM.

1. SSH ke alat penemuan VM.

   ```
   ssh discovery@<discovery-tool-vm-ip>
   ```

2. Edit file konfigurasi Kerberos di. `/etc/krb5.conf`

   ```
   sudo nano /etc/krb5.conf
   ```

   Tambahkan konfigurasi berikut, ganti nilai placeholder dengan detail lingkungan Anda.

```
[libdefaults]
    default_realm = EXAMPLE.COM
    dns_lookup_realm = false
    dns_lookup_kdc = true

[realms]
    EXAMPLE.COM = {
        kdc = dc01.example.com
    }

[domain_realm]
    .example.com = EXAMPLE.COM
    example.com = EXAMPLE.COM
```

> **Important**
>
> Kerberos peka huruf besar/kecil. Nama ranah harus dalam huruf besar (misalnya,`EXAMPLE.COM`, tidak`example.com`). Nama domain di `[domain_realm]` bagian harus dalam huruf kecil.

Beberapa domain Direktori Aktif

Alat penemuan mendukung beberapa kredensyal Kerberos untuk domain Active Directory yang berbeda. Setiap kredensi mengautentikasi secara independen, sehingga Anda dapat mengonfigurasi beberapa kredensyal secara normal dan isolasi otomatis.

Jika Anda memiliki server di beberapa domain, tambahkan entri untuk setiap ranah dalam file Anda`/etc/krb5.conf`:

```
[libdefaults]
    default_realm = DEV.COMPANY.COM
    dns_lookup_realm = false
    dns_lookup_kdc = true

[realms]
    DEV.COMPANY.COM = {
        kdc = dc01.dev.company.com
    }
    PROD.COMPANY.COM = {
```

```
        kdc = dc01.prod.company.com
    }

[domain_realm]
    .dev.company.com = DEV.COMPANY.COM
    dev.company.com = DEV.COMPANY.COM
    .prod.company.com = PROD.COMPANY.COM
    prod.company.com = PROD.COMPANY.COM
```

3. Verifikasi bahwa Anda dapat memperoleh tiket Kerberos dengan menjalankan perintah. `kinit`

```
kinit username@REALM.COM
```

   Masukkan kata sandi saat diminta. Jika perintah selesai tanpa kesalahan, otentikasi berhasil.

4. Verifikasi tiket dengan menjalankan `klist` perintah.

```
klist
```

   Output yang diharapkan mirip dengan yang berikut ini.

```
Ticket cache: FILE:/tmp/krb5cc_1000
Default principal: username@REALM.COM

Valid starting       Expires              Service principal
01/01/2025 12:00:00  01/01/2025 22:00:00  krbtgt/REALM.COM@REALM.COM
```

5. Konfigurasikan alat penemuan dengan prinsip case-sensitive yang sama dengan yang Anda gunakan `kinit` (misalnya,`username@REALM.COM`).

`krb5.conf`Konfigurasi eksplisit mungkin tidak diperlukan jika lingkungan Anda memiliki catatan DNS SRV yang dikonfigurasi untuk penemuan layanan Kerberos. Untuk informasi lebih lanjut tentang opsi konfigurasi Kerberos, lihat dokumentasi MIT Kerberos krb5.conf dan contoh file krb5.conf.

Temukan konfigurasi Kerberos dari mesin yang bergabung dengan domain

Jika Anda tidak memiliki detail konfigurasi Kerberos, Anda dapat mengambilnya dari mesin Windows yang bergabung ke domain. Jalankan perintah berikut dari prompt perintah pada mesin yang bergabung dengan domain.

Untuk menemukan nama domain, jalankan perintah berikut.

```
echo %USERDNSDOMAIN%
```

Contoh output:

```
EXAMPLE.COM
```

Untuk menemukan nama host pengontrol domain, jalankan perintah berikut.

```
nltest /dsgetdc:EXAMPLE.COM
```

Contoh output:

```
           DC: \\dc01.example.com
      Address: \\10.0.1.100
     Dom Guid: xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx
     Dom Name: EXAMPLE.COM
  Forest Name: example.com
 Dc Site Name: Default-First-Site-Name
Our Site Name: Default-First-Site-Name
        Flags: 0xe00033fd
The command completed successfully
```

Petakan output ke `krb5.conf` konfigurasi Anda sebagai berikut:

- Realm — Gunakan nilai dari `%USERDNSDOMAIN%` huruf besar (misalnya,`EXAMPLE.COM`).
- KDC — Gunakan nama host DC dari `nltest` output (misalnya,`dc01.example.com`).

Konfigurasikan akses vCenter

1. Pada halaman alat Discovery, di bawah Langkah 1. Konfigurasikan sumber penemuan, pilih Konfigurasikan sumber.
2. Pada halaman Konfigurasi sumber penemuan, berikan nama Ramah, vCenter, Nama Pengguna FQDN/IP, dan Kata Sandi.
3. Pilih Simpan konfigurasi.

   Alat penemuan mulai mengumpulkan informasi vCenter, seperti yang dijelaskan dalam Inventaris Ditemukan.

Setelah konfigurasi awal pilih Edit akses vCenter di bingkai status alat Discovery untuk mengubah pengaturan akses vCenter Anda.

Alat penemuan mengumpulkan dari semua server vCenter yang dikonfigurasi secara paralel. Jika satu server vCenter tidak dapat dijangkau selama pengumpulan, alat melaporkan keberhasilan sebagian dan terus mengumpulkan dari server vCenter yang tersisa.

Jika VM muncul di beberapa server vCenter (misalnya, karena host ESXi bersama atau cross-vCenter vMotion), alat penemuan secara otomatis menghapus duplikasi VM. Setiap VM unik hanya muncul sekali dalam inventaris.

Konfigurasikan Hyper-V akses

1. Pada halaman alat Discovery, di bawah Langkah 1. Konfigurasikan sumber penemuan, pilih Konfigurasikan sumber.
2. Pada halaman Konfigurasi sumber penemuan, berikan nama yang ramah, FQDN host atau alamat IP, jenis otentikasi (NTLM atau Kerberos), nama pengguna WinRM, dan kata sandi WinRM.
3. Pilih Simpan konfigurasi.

   Alat penemuan mulai mengumpulkan Hyper-V informasi, seperti yang dijelaskan dalam Inventaris yang ditemukan.

Koleksi dimulai secara otomatis setelah Anda menyimpan kredensialnya.

Untuk cluster Hyper-V failover, Anda dapat menambahkan beberapa host di cluster yang sama. Alat ini secara otomatis menghapus duplikasi VM yang muncul di lebih dari satu host.

Impor server

1. Arahkan ke halaman Impor server dari beranda alat Discovery.
2. Siapkan file CSV dengan kolom berikut: `hostname_or_ip` (wajib), `os_credential_name` (opsional), dan `oracle_credential_name` (opsional).

   - `hostname_or_ip`Nilai harus berupa alamat IPv4 yang valid atau nama domain yang sepenuhnya memenuhi syarat (FQDN).
   - `os_credential_name`Nilai, jika disediakan, harus sesuai dengan nama ramah kredensi OS yang sudah Anda konfigurasikan (SSH, WinRM, atau SNMP). Biarkan kosong untuk server di mana Anda belum mengkonfigurasi kredensi OS.

- `oracle_credential_name`Nilai, jika disediakan, harus sesuai dengan nama ramah kredensi Oracle yang sudah Anda konfigurasikan.

3. Unggah file CSV. Alat ini memvalidasi semua baris dan menolak file jika ada baris yang tidak valid.

Setelah impor berhasil, alat secara otomatis memulai pengumpulan metrik database, jaringan, dan OS untuk server yang diimpor, jika kredensyal OS dikonfigurasi. Jika Anda mengunggah file CSV lain, catatan yang ada diperbarui tanpa membuat duplikat dan catatan baru digabungkan ke dalam inventaris.

Impor otoritas sertifikat yang ditandatangani sendiri ke alat penemuan (Opsional)

Ini diperlukan saat Anda menggunakan WinRM melalui HTTPS dan server target menggunakan sertifikat WinRM HTTPS yang ditandatangani oleh Otoritas Sertifikat (CA) yang ditandatangani sendiri, dan Anda ingin mengaktifkan “Validasi sertifikat SSL server” pada alat penemuan.

Prasyarat

1. Self-signed Sertifikat CA yang digunakan untuk menandatangani sertifikat WinRM HTTPS pada server target
2. Sertifikat dalam format PEM (ekstensi.pem atau .crt)

Untuk mengimpor otoritas sertifikat yang ditandatangani sendiri pada alat penemuan VM:

1. Alat Ssh ke Discovery VM
2. Tempatkan sertifikat CA yang menandatangani sertifikat WinRM server target Anda ke direktori trust store `/etc/pki/ca-trust/source/anchors/` pada alat penemuan VM. Sebagai contoh: `sudo cp winrm-ca.pem /etc/pki/ca-trust/source/anchors/winrm-ca.pem`. Catatan: Jika server target Anda menggunakan sertifikat yang ditandatangani oleh CA yang berbeda, salin semua sertifikat CA yang relevan ke direktori ini.
3. Perbarui toko kepercayaan sertifikat: `sudo update-ca-trust`
4. Reboot VM
5. (Opsional) Untuk memverifikasi bahwa sertifikat telah berhasil diimpor, Anda dapat menjalankan perintah berikut. `sudo trust list –filter=ca-anchors | grep -A 5 "<certificate_name>"`

Lihat Instalasi dan konfigurasi untuk Windows Remote Management

Konfigurasikan alat penemuan untuk akses OS

Konfigurasikan akses OS sehingga alat penemuan dapat:

- Temukan database untuk melakukan penilaian basis data dan untuk membantu migrasi VM,
- Lacak koneksi jaringan antar server dalam inventaris Anda, termasuk proses yang terkait dengan setiap koneksi, untuk membantu dalam pemetaan ketergantungan aplikasi dan perencanaan gelombang. Hanya koneksi di mana kedua titik akhir berada dalam inventaris alat penemuan yang disertakan.

Aktifkan alat penemuan OS Access

1. Arahkan ke halaman akses Siapkan OS untuk menyediakan kredensyal Windows dan Linux.
2. Pilih protokol yang ingin Anda tambahkan kredensialnya.
3. Berikan kredensyal yang diperlukan untuk protokol yang dipilih.
4. Pilih Auto-connectuntuk mengaktifkan alat penemuan untuk mencoba semua kredensyal yang disediakan di server yang ditemukan hingga kredensyal yang cocok ditemukan untuk setiap server.

   Lihat the section called “Menggunakan Auto-Connect Fitur Dengan Hati-hati” rekomendasi keamanan penting terkait fitur sambungkan otomatis.
5. Pilih Siapkan dan sambungkan.

Ketika proses pencocokan OS selesai, Anda melihat pesan bahwa pengumpulan data sedang berlangsung, dan kesalahan terkait server yang kredensialnya cocok tidak ditemukan.

Pengaturan protokol yang didukung

Anda harus mengatur protokol WinRM, SSH, dan SNMP pada server target agar alat penemuan dapat berkomunikasi dengan mereka.

Mengatur WinRM dan WMI

WinRM diinstal secara otomatis dengan semua versi sistem operasi Windows yang didukung saat ini.

Untuk memverifikasi atau mengedit konfigurasi WinRM, gunakan alat baris `winrm` perintah:

- Verifikasi pendengar WinRM yang diinstal: `winrm enumerate winrm/config/listener`
- Verifikasi konfigurasi WinRM: `winrm get winrm/config`
- Contoh perintah untuk mengatur WinRM: `winrm quickconfig -transport:https`

Pelabuhan Pendengar

Port HTTP default adalah 5985; HTTPS adalah 5986. Anda dapat menggunakan port lain sesuai kebutuhan. Port harus terbuka antara alat penemuan dan server target.

Enkripsi

Alat penemuan menggunakan komunikasi WinRM terenkripsi. Kami menyarankan agar pendengar WinRM di server target juga menggunakan enkripsi: `winrm set winrm/config/service '@{AllowUnencrypted="false"}'`

NTLM vs Kerberos

Protokol otentikasi WinRM Kerberos dan NTLM didukung oleh alat penemuan. NTLM hanya dapat digunakan dengan HTTPS dan Kerberos dapat digunakan dengan HTTP atau HTTPS.

Persyaratan WMI

Alat penemuan menanyakan ruang nama WMI berikut. Akun WinRM memerlukan akses baca ke setiap namespace yang relevan dengan modul koleksi Anda:

Ruang nama WMI	Digunakan oleh
`root\cimv2`	Metrik OS, metadata Hyper-V host, koleksi SQL Server
`root\virtualization\v2`	Hyper-V Persediaan VM
`root\StandardCIMV2`	Koleksi jaringan
`root\Microsoft\SqlServer\ComputerManagement*`	Koleksi SQL Server
`root\Microsoft\SqlServer\ReportServer\*`	Koleksi SQL Server (SSRS)

Untuk pengumpulan jaringan, pastikan kondisi ini terpenuhi:

- Izinkan konektivitas jaringan melalui ICMP
- Izinkan konektivitas jaringan melalui port TCP 135 + rentang port TCP sementara (49152 - 65535)
- Nonaktifkan UAC
- Izin DCOM jarak jauh diatur
- Buat akun layanan khusus dengan izin minimal yang diperlukan
- Izin namespace WMI disiapkan untuk akun Windows dengan ruang nama:, kelas `\\root\ \standardcimv2` `MSFT_NetTCPConnection`

Untuk koleksi SQL Server, akun Windows (lokal atau domain) milik Grup Administrator Lokal diperlukan karena persyaratan izin objek WMI yang kompleks.

Menyiapkan SSH

- Port default adalah 22. Port khusus didukung. Port yang dikonfigurasi harus terbuka antara alat penemuan dan server target.
- Agar pengumpulan jaringan SSH berfungsi dengan baik, sediakan pengguna yang dikonfigurasi untuk sudo tanpa kata sandi.
- Pastikan bahwa perintah berikut tersedia di server Linux target (diinstal secara default pada sebagian besar distribusi): `ss` atau `netstat` untuk pengumpulan jaringan, dan,,,`lsblk`,`iostat`,`dmidecode`,,`smartctl`,`top`, `ps` `freeip`, dan `df` untuk pengumpulan metrik OS.

Alat penemuan mendukung dua metode otentikasi untuk SSH:

Opsi 1: Nama pengguna dan kata sandi

Berikan nama pengguna dan kata sandi SSH. Ini adalah metode otentikasi default.

Opsi 2: Kunci pribadi SSH

Berikan nama pengguna SSH dan kunci pribadi dalam format PEM. Untuk menggunakan opsi ini, pilih SSH Key dari dropdown tipe otentikasi saat mengonfigurasi kredensyal SSH. Jika kunci privat dienkripsi dengan frasa sandi, masukkan frasa sandi di bidang Frasa Sandi Kunci opsional.

Format kunci berikut didukung:

- RSA
- ECDSA
- Ed25519
- Format OpenSSH
- Format PKCS #8

Kedua metode otentikasi mendukung koneksi otomatis. Kredensyal disimpan dienkripsi saat istirahat.

Mengatur SNMP

- Port defaultnya adalah 161/UDP. Port khusus didukung. Port yang dikonfigurasi harus terbuka antara alat penemuan dan server target.
- Untuk SNMP v2: Menyediakan string komunitas read-only yang dapat mengakses OID koneksi TCP.
- Untuk SNMP v3: Berikan username/password dan auth/privacy detail dengan izin hanya-baca yang dapat mengakses OID koneksi TCP

Alat penemuan membutuhkan akses ke:

- `"1.3.6.1.2.1.6.13.1.1."` (tcpConnState)
- `"1.3.6.1.2.1.6.19.1.8."` (tcpConnectionProcess)
- `"1.3.6.1.2.1.25.4.2.1.2."` (hrSWRunName)

Konfigurasikan akses database Oracle

Konfigurasikan akses database Oracle untuk mengumpulkan metadata database Oracle terperinci secara langsung melalui koneksi SQL. Metadata yang dikumpulkan mencakup topologi CDB dan PDB, penggunaan fitur, dan opsi yang diinstal. Data ini membantu Anda merencanakan migrasi database Oracle dengan lebih akurat. Anda dapat mengumpulkan Oracle Database 12c Release 1 (12.1) dan kemudian melalui koneksi SQL langsung. OS-level deteksi fallback berfungsi dengan semua versi Oracle.

Konfigurasikan kredenal Oracle di alat penemuan

1. Pada halaman alat Discovery, di sidebar, pilih akses Database.
2. Pilih Tambahkan kredensi Oracle.

3. Saat diminta, berikan informasi berikut:
   - Nama ramah — Nama deskriptif untuk kredensi ini (misalnya,`Oracle Production`).
   - Port - Port pendengar Oracle (default 1521).
   - Nama layanan — Nama layanan Oracle untuk database target.
   - Nama pengguna - Nama pengguna akun layanan Oracle.
   - Kata sandi — Kata sandi akun layanan Oracle.
   - Auto-connect— Aktifkan opsi ini untuk mencoba kredensyal terhadap semua server di inventaris Anda. Matikan opsi ini untuk menetapkan kredensi secara manual ke server tertentu.
4. Untuk menambahkan lebih banyak kredensyal (misalnya, untuk lingkungan Oracle yang berbeda), pilih Tambah kredensi Oracle lagi.
5. Pilih Simpan.

Mode kredensyal

Saat Anda mengonfigurasi kredensyal Oracle, Anda dapat memilih di antara dua mode:

- Manual — Sematkan kredensi ke server tertentu. Alat penemuan menggunakan kredensi itu secara eksklusif untuk server itu. Jika koneksi gagal, tidak ada fallback yang terjadi. Perbaiki konfigurasi kredensyal untuk menyelesaikan masalah.
- Auto-connect— Alat penemuan mencoba setiap kredensi koneksi otomatis terhadap setiap server di inventaris Anda. Ketika kredensi berhasil untuk server, alat penemuan menggunakan kredensi itu untuk semua putaran pengumpulan berikutnya.

Aliran deteksi

Ketika Anda mengkonfigurasi kredensyal Oracle, alat penemuan pertama mencoba koneksi SQL langsung. Jika semua kredensyal database gagal, alat akan kembali ke OS-level deteksi melalui SSH atau WinRM, sehingga Anda masih dapat menemukan instalasi Oracle tanpa akses database.

Memperbarui alat penemuan

Alat penemuan tidak memiliki fitur pembaruan otomatis namun Anda akan menerima pemberitahuan pengingat setelah 30 hari instalasi untuk memperbarui. Disarankan untuk menjaga aplikasi tetap up-to-date untuk menerima fitur terbaru dan patch keamanan.

Untuk memperbarui alat secara manual

1. Unduh file gambar alat penemuan terbaru (OVA untuk VMware atau VHD untuk Hyper-V) dari tautan yang disediakan.
2. (Opsional) Kami menyarankan Anda menghapus file gambar alat penemuan sebelumnya sebelum Anda menerapkan yang terbaru.
3. Ikuti langkah-langkah di bagian Deploy the discovery tool untuk menerapkan versi yang diperbarui.

Mencabut akses

Anda dapat mencabut akses untuk setiap sumber penemuan secara independen. Ketika Anda mencabut akses untuk satu sumber, data dari sumber lain tidak terpengaruh.

- Mencabut akses vCenter - Menghapus kredensyal dan data vCenter. VMware-collected Tidak menghapus Hyper-V data, data server yang diimpor, atau kredensyal OS.
- Mencabut Hyper-V akses — Menghapus Hyper-V kredensyal dan data saja. Hyper-V-collected
- Menghapus server yang diimpor - Menghapus server yang diimpor dari inventaris. Pengumpulan data hilir (jaringan, database) yang dikumpulkan dari server tersebut dipertahankan.

Izin yang diperlukan untuk alat penemuan

Alat penemuan terhubung ke infrastruktur Anda menggunakan beberapa protokol. Setiap protokol dan modul pengumpulan memerlukan izin akun tertentu. Bagian ini menjelaskan izin minimum yang diperlukan untuk pengumpulan data lengkap, dan data apa yang hilang jika izin tertentu tidak tersedia.

VMware vCenter

Alat penemuan terhubung ke VMware vCenter Server pada port 443 (HTTPS) dan melakukan operasi read-only. Tidak ada perubahan yang dilakukan pada lingkungan vCenter Anda.

Peran minimum yang diperlukan: Read-Only, ditetapkan pada tingkat root vCenter.

Read-Only Peran ini menyediakan semua akses yang dibutuhkan alat penemuan untuk mengumpulkan inventaris VM (nama, UUID, CPU, memori, disk, jaringan, status daya, OS tamu) dan metrik kinerja (pemanfaatan CPU, pemanfaatan memori, IOPS disk, dan throughput).

Rekomendasi: Buat pengguna vCenter khusus dengan Read-Only peran bawaan di tingkat pusat data root.

Hyper-V tuan rumah (Windows)

Alat penemuan terhubung ke Hyper-V host melalui WinRM (port 5985 untuk HTTP, port 5986 untuk HTTPS) dan menjalankan PowerShell perintah untuk mengumpulkan inventaris VM, metadata host, dan kinerja penyimpanan.

Keanggotaan grup minimum yang diperlukan pada setiap Hyper-V host:

Grup Windows	Mengapa itu dibutuhkan
Pengguna Manajemen Jarak Jauh	Akses jarak jauh WinRM Baseline
Hyper-V Administrator	Koleksi inventaris VM (daftar VM, disk, jaringan, memori, OS tamu)
Pengguna Monitor Kinerja	Penghitung I/O kinerja penyimpanan (read/write IOPS dan throughput per VM)

Akun ini juga membutuhkan akses baca WMI ke `root\cimv2` namespace untuk versi OS host dan UUID perangkat keras. Akses ini diberikan secara default untuk administrator lokal tetapi harus dikonfigurasi secara eksplisit untuk akun non-administrator.

Rekomendasi: Buat akun layanan domain khusus dan tambahkan ke grup Pengguna Manajemen Jarak Jauh, Hyper-V Administrator, dan Pengguna Monitor Kinerja di setiap Hyper-V host. Hindari menggunakan akun administrator domain.

Server Linux (SSH)

Alat penemuan terhubung ke server Linux melalui SSH (port 22) untuk mengumpulkan metrik OS, koneksi jaringan, dan inventaris server. Sebagian besar pengumpulan data berjalan sebagai pengguna biasa. Sejumlah kecil perintah mencoba `sudo` dan secara otomatis mundur jika sudo tidak tersedia.

Akses minimum yang diperlukan: Akun pengguna SSH yang dapat masuk ke server target.

Akses yang disarankan: Pengguna SSH yang sama dikonfigurasi dengan sudo tanpa kata sandi untuk pengumpulan data lengkap.

Apa yang membutuhkan sudo dan apa yang terjadi tanpanya:

Data yang dikumpulkan	Sudo dibutuhkan?	Apa yang Anda kehilangan tanpa sudo
Nama server, OS, CPU, memori, IP, jumlah disk	Tidak	Tidak ada - dikumpulkan sebagai pengguna biasa
CPU, memori, dan pemanfaatan jaringan	Tidak	Tidak ada - dikumpulkan sebagai pengguna biasa
Disk IOPS, throughput, dan ruang	Tidak	Tidak ada - dikumpulkan sebagai pengguna biasa
Konfigurasi antarmuka jaringan	Tidak	Tidak ada - dikumpulkan sebagai pengguna biasa
Menjalankan proses (nama, PID, perintah)	Tidak	Tidak ada - dikumpulkan sebagai pengguna biasa
Server UUID dan SMBIOS UUID	Ya	Bidang UUID kosong di ekspor
Produsen perangkat keras (deteksi fisik vs. virtual)	Ya	Deteksi tipe sumber daya kurang akurat pada distribusi yang lebih lama
Deteksi volume logis LVM	Ya	Volume LVM tidak terdeteksi; jenis volume mungkin ditampilkan sebagai “Tidak Diketahui”
Koneksi jaringan dengan nama proses dan PID	Ya	Koneksi masih dikumpulkan, tetapi tanpa atribusi proses (PID dan kolom nama proses kosong)
OS-level Deteksi Oracle Database (mundur saat kredensyal Oracle tidak dikonfigurasi)	Ya	Deteksi kehadiran Oracle dasar (oratab, monitor proses) bekerja tanpa sudo. Koleksi terperinci (opatch, lsnrctl, ASM, Grid Infrastructure) memerlukan akses tanpa kata sandi ke binari rumah Oracle. `sudo -u oracle`

> **Note**
>
> Alat penemuan tidak pernah gagal sepenuhnya karena kurangnya sudo. Ini mengumpulkan apa yang bisa dan melaporkan sebagian hasil. Namun, untuk data yang paling lengkap — terutama pemetaan ketergantungan jaringan dengan nama proses — kami merekomendasikan sudo tanpa kata sandi.

Utilitas yang diperlukan pada server target: Perintah berikut harus tersedia (diinstal secara default pada sebagian besar distribusi Linux): `ss` atau`netstat`,`lsblk`,`top`,`ps`,`free`,`ip`,`df`,`hostname`, `catnproc`, dan`grep`. Opsional untuk data tambahan: `iostat` (I/O detail disk), `dmidecode` (UUID perangkat keras), `smartctl` (tipe antarmuka disk), `lvdisplay` (deteksi LVM).

Server Windows (WinRM) - Metrik OS

Alat penemuan terhubung ke server Windows melalui WinRM untuk mengumpulkan metrik OS (inventaris server, kinerja, penyimpanan, antarmuka jaringan, dan proses yang berjalan).

Akses minimum yang diperlukan: Akun WinRM-enabled pengguna dengan akses jarak jauh ke server target.

Akses yang disarankan: Pengguna dalam grup Administrator lokal untuk pengumpulan data lengkap.

Tingkat izin	Apa yang memungkinkan	Apa yang Anda kehilangan tanpanya
Pengguna Manajemen Jarak Jauh (akses WinRM)	Semua pengumpulan data Windows	Tidak ada data yang dikumpulkan dari server
Akses baca WMI ke `root\cimv2`	Nama server, versi OS, memori, UUID, serial BIOS, ruang disk	Bidang inventaris server kosong
Pengguna Monitor Kinerja	Pemanfaatan CPU, throughput jaringan, IOPS disk dan throughput	Metrik kinerja tidak dikumpulkan
Administrator Lokal	Menjalankan nama pemilik proses	Daftar proses dikumpulkan tetapi user/owner kolom kosong

> **Note**
>
> Keanggotaan Administrator Lokal secara implisit memberikan akses Pengguna Monitor Kinerja dan akses baca WMI, sehingga memenuhi semua persyaratan di atas.

> **Note**
>
> Deteksi OS-level fallback Oracle Database pada Windows menggunakan kueri registri dan `Get-Service` dan `Get-Process` perintah ketika kredensyal Oracle tidak dikonfigurasi. Perintah ini hanya memerlukan akses Pengguna Manajemen Jarak Jauh standar. Tidak diperlukan izin tambahan di luar persyaratan metrik OS.

Server Windows (WinRM) - koleksi SQL Server

Koleksi SQL Server menemukan instance SQL Server, Layanan Pelaporan (SSRS), dan Layanan Integrasi (SSIS) di server Windows.

Akses yang disarankan: Pengguna dalam grup Administrator lokal di setiap server Windows target.

Administrator Lokal direkomendasikan karena penemuan SQL Server menanyakan beberapa ruang nama WMI dan memerlukan akses yang lebih tinggi untuk beberapa operasi:

Apa yang ditemukan	Izin diperlukan	Apa yang Anda kehilangan tanpanya
Contoh SQL Server Database Engine (versi, edisi, status)	Akses baca WMI ke ruang nama `root\Microsoft\SqlServer\ComputerManagement*`	Instance SQL Server tidak ditemukan
Layanan Pelaporan SQL Server (SSRS)	Akses baca WMI ke ruang nama `root\Microsoft\SqlServer\ReportServer`	Komponen SSRS tidak ditemukan

Apa yang ditemukan	Izin diperlukan	Apa yang Anda kehilangan tanpanya
URL SSRS dan konfigurasi port	Hak istimewa yang ditinggikan (Admin Lokal)	Rincian reservasi URL SSRS hilang
SQL Server Integration Services (SSIS)	Akses baca registri (HKLM)	Versi dan edisi SSIS tidak ada
Port-to-service asosiasi	Akses ke enumerasi pendengar TCP	Tidak dapat mengaitkan port mendengarkan dengan layanan SQL Server

> **Note**
>
> Koleksi SQL Server adalah Windows-only. Alat penemuan melewatkan server Linux untuk penemuan SQL Server.

Database Oracle (SQL)

Koleksi Oracle Database terhubung langsung ke instance Oracle pada port 1521 (atau port khusus) untuk mengumpulkan metadata database. Alat penemuan tidak memerlukan hak istimewa DBA atau SYSDBA.

Akses minimum yang diperlukan: Akun layanan Oracle read-only dengan hibah SELECT_CATALOG_ROLE.

Alat penemuan tidak mengakses tampilan Paket Diagnostik atau Paket Tuning, jadi tidak ada lisensi Oracle tambahan yang diperlukan untuk pengumpulan data.

SQL untuk membuat akun:

```
CREATE USER discovery_user IDENTIFIED BY <password>;
GRANT CREATE SESSION TO discovery_user;
GRANT SELECT_CATALOG_ROLE TO discovery_user;
```

OS-level fallback: Jika kredensyal database tidak dikonfigurasi atau koneksi gagal, alat penemuan menggunakan kredensyal SSH atau WinRM OS yang ada untuk mendeteksi instalasi Oracle. Tidak diperlukan izin Oracle-specific OS tambahan di luar apa yang dibutuhkan modul metrik OS.

Koleksi jaringan

Pengumpulan jaringan menggunakan protokol yang berbeda tergantung pada sistem operasi server:

Server OS	Protokol	Izin diperlukan
Linux	SSH	Pengguna reguler untuk data koneksi. Sudo tanpa kata sandi direkomendasikan untuk detail tingkat proses (PID dan nama proses).
Linux	SNMPv2	String komunitas read-only dengan akses ke TCP MIB (tcpConnState, tcpConnectionProcess) dan Host Resources MIB (hr). SWRunName
Linux	SNMPv3	Pengguna USM dengan akses baca ke MIB yang sama dengan SNMPv2. Mendukung tingkat AuthNoPriv keamanan no, authNoPriv, dan AuthPriv.
Windows	WinRM	WMI membaca akses ke `root\StandardCIMV2` namespace (kelas MSFT_NettcpConnection).

Referensi cepat: Izin minimum berdasarkan kasus penggunaan

Kasus penggunaan	Jenis akun	Izin minimum
Penemuan VMware VM	pengguna vCenter	Read-Only peran di tingkat pusat data root
Hyper-V Penemuan VM	Domain Windows atau akun lokal	Pengguna Manajemen Jarak Jauh + Hyper-V Administrator+Pengguna Monitor Kinerja di setiap host
Metrik OS Linux — data lengkap	Pengguna SSH	Sudo tanpa kata sandi

Kasus penggunaan	Jenis akun	Izin minimum
Metrik OS Linux — sebagian data	Pengguna SSH	Pengguna biasa (tidak ada sudo). Info proses UUID, pabrikan, LVM, dan jaringan akan hilang.
Metrik OS Windows — data lengkap	Pengguna WinRM	Administrator Lokal
Metrik OS Windows — data dasar	Pengguna WinRM	Pengguna Manajemen Jarak Jauh + Akses baca WMI `root\cimv2`
Penemuan database SQL Server	Pengguna WinRM	Administrator Lokal
Penemuan database Oracle (SQL)	Akun layanan Oracle	SELECT_CATALOG_ROLE (hanya-baca, tanpa DBA atau SYSDBA)
Koleksi jaringan - Linux (SSH)	Pengguna SSH	Sudo tanpa kata sandi untuk data tingkat proses; pengguna biasa hanya untuk data koneksi
Koleksi jaringan - Linux (SNMP)	String komunitas SNMP atau pengguna USM	Baca akses ke TCP dan Host Resources MIB
Koleksi jaringan — Windows	Pengguna WinRM	Akses baca WMI ke `root\StandardCIMV2`

Pengumpulan data

Jadwal pengumpulan alat penemuan

Setelah pengumpulan penemuan awal Anda, alat penemuan terus berjalan pada jadwal yang terhuyung-huyung untuk menghindari pertengkaran sumber daya:

- Penemuan VMware - setiap jam (pukul 00 UTC)
- Hyper-V penemuan — setiap jam (pada:20 UTC)

Alat penemuan juga mengumpulkan metrik OS melalui modul independen berikut, masing-masing dengan jadwal terhuyung-huyung sendiri:

- Metrik jaringan — setiap 15 detik, mungkin lebih jarang untuk lingkungan yang luas
- Metrik kinerja server - setiap 10 menit (pada:03, :13, :23, :33, :43, :53 UTC)
- Metrik kinerja penyimpanan - setiap 10 menit (pada:07, :17, :27, :37, :47, :57 UTC)
- Menjalankan proses - setiap jam (pada:40 UTC)
- Data penyediaan server — setiap hari (pukul 00:05 UTC)
- Data penyediaan penyimpanan — setiap hari (pukul 00:35 UTC)
- Antarmuka jaringan - setiap hari (pukul 01:05 UTC)
- Penemuan SQL Server - setiap hari (pukul 03:05 UTC)
- Penemuan database Oracle — setiap hari (pukul 05:05 UTC)

Anda dapat memulai, menghentikan, atau memicu setiap modul metrik OS secara mandiri dengan menggunakan Kumpulkan data sekarang.

Untuk menjalankan koleksi secara manual, dari menu Tindakan pilih:

- Mulai - Mengaktifkan modul penemuan.
- Berhenti - Menonaktifkan modul penemuan.
- Kumpulkan data sekarang — Mulai penemuan segera. Gunakan opsi ini, misalnya, setelah Anda membuat perubahan di jaringan Anda.

Tindakan ini berlaku per modul. Anda dapat mengontrol modul metrik OS satu per satu.

Upaya pengumpulan data OS

Ketika server baru ditemukan, alat penemuan mencoba setiap kredensi yang dikonfigurasi untuk setiap alamat IP dan nama host. Setelah alat penemuan menemukan kredensi yang valid, ia terus menggunakan kredensi itu kecuali Anda menambahkan kredensi baru.

Setelah kegagalan pengumpulan, alat penemuan mencoba mengumpulkan data jaringan untuk server setelah 3 menit, 30 menit, 2 jam, dan kemudian 6 jam. Setelah 4 upaya gagal, alat penemuan terus mencoba semua kredensyal yang dikonfigurasi setiap 6 jam sekali.

Inventaris yang ditemukan

Setelah Anda mengonfigurasi sumber penemuan, nilai Jumlah server yang ditemukan dalam bingkai status alat Discovery mulai meningkat. Status penemuan untuk sumber yang dikonfigurasi berubah menjadi Diaktifkan dalam bingkai modul Collection. Halaman inventaris menunjukkan server dari semua sumber yang dikonfigurasi: VMware VM, VM, dan server yang Hyper-V diimpor. Setiap server menunjukkan sumber dan status pengumpulannya per modul.

Arahkan ke halaman inventaris Ditemukan untuk melihat server yang ditemukan alat penemuan. Dari halaman ini, pilih Unduh inventaris untuk mengunduh file ZIP (`discovery_tool_export.zip`) yang berisi hingga 30 hari data yang dikumpulkan, termasuk file MPA untuk semua sumber yang dikonfigurasi, data pemanfaatan kinerja, informasi basis data, dan informasi komunikasi server-ke-server.

Anda dapat mengunduh file ZIP sementara alat penemuan terus bekerja, dan mendapatkan hasil sebagian. Unggah file ini ke Penilaian migrasi untuk mendapatkan kasus bisnis untuk migrasi.

Opsi ekspor

Saat mengekspor data, Anda dapat menyesuaikan ekspor dengan opsi berikut:

Rentang tanggal

Pilih tanggal mulai dan tanggal akhir untuk mengekspor hanya data yang dikumpulkan dalam periode waktu tersebut. Kedua tanggal itu inklusif. Rentang tanggal maksimum adalah 30 hari.

> **Note**
>
> Alat penemuan menyimpan hingga 30 hari data yang dikumpulkan. Jika Anda membutuhkan data yang mencakup lebih dari 30 hari, jalankan ekspor tambahan setiap 30 hari untuk menangkap semua data.

Pemilihan modul

Pilih modul data mana yang akan disertakan dalam ekspor. Anda dapat mengekspor semua modul atau memilih yang spesifik:

Modul	Deskripsi
Data VMware	Inventaris mesin virtual dari server vCenter
Hyper-V data	Inventaris mesin virtual dari Hyper-V host
Data jaringan	Koneksi jaringan antar server
Inventaris server	Perangkat keras server dan informasi OS
Metrik kinerja server	CPU, memori, dan pemanfaatan jaringan
Kinerja penyimpanan server	Disk IOPS dan throughput
Konfigurasi penyimpanan	Konfigurasi disk dan volume
Antarmuka jaringan	Detail adaptor jaringan
Metrik proses	Proses berjalan
Data SQL Server	Inventaris basis data SQL Server
Data Database Oracle	Inventaris database Oracle - CDB, PDB, fitur, opsi, dan komponen

Jika Anda tidak memilih modul apa pun, alat penemuan mengekspor semua data yang tersedia.

Poin data dikumpulkan

Alat penemuan mengumpulkan data komprehensif di seluruh VMware, metrik OS Hyper-V, database, dan komponen jaringan. Bagian berikut merinci titik data spesifik yang dikumpulkan untuk setiap komponen.

Pengumpulan data VMware

Tabel ini menjelaskan informasi mesin virtual VMware yang dikumpulkan oleh alat penemuan:

Nama	Tipe	Kategori	Nilai sampel
vm_nama	String	Info VM	“w2k22-snmpd-v2-en-us-mssql-2022-testcase4-1"
vm_id	String	Info VM	“vm-30920"
vm_uuid	String	Info VM	“4201ecf8-cc44-ee7e-01da-34dfb2acf6c0"
powerstate	String	Info VM	“PowerEdon”
host	String	Info VM	“esxi-70-node1.testlab.local”
primer_ip_address	String	Info VM	“192.168.0.52"
CPU	Bilangan Bulat	Info VM	2
memori	Bilangan Bulat	Info VM	4096
total_disk_capacity_mib	Bilangan Bulat	Info VM	32768
os_according_to_the_configuration_file	String	Info VM	“Microsoft Windows Server 2016 atau yang lebih baru (64-bit)”
max_cpu_usage_pct_dec	Desimal	Kinerja VM	79.33
avg_cpu_usage_pct_dec	Desimal	Kinerja VM	45.06
max_ram_usage_pct_dec	Desimal	Kinerja VM	63,99
avg_ram_utl_pct_dec	Desimal	Kinerja VM	29.27

Hyper-V pengumpulan data

Tabel ini menjelaskan informasi mesin Hyper-V virtual yang dikumpulkan oleh alat penemuan:

Nama	Tipe	Kategori	Nilai sampel
vm_nama	String	Info VM	“win2022-hyperv-test-01"
vm_id	String	Info VM	“a1b2c3d4-e5f6-7890-abcd-ef1234567890"
powerstate	String	Info VM	“Berlari”
CPU	Bilangan Bulat	Info VM	4
memory_mb	Bilangan Bulat	Info VM	8192
disk_paths	String	Disk	“C:\\ VMs\\ disk1.vhdx”
disk_size_gb	Desimal	Disk	127.0
network_adapters	String	Jaringan	“00:15:5D:01:02:03”
ip_alamat	String	Jaringan	“10.0.1.50"
host_name	String	Host	“hyperv-host-01.example.com”
host_os_version	String	Host	“Pusat Data Windows Server 2022”
cluster_name	String	Host	“FailoverCluster01"
hypervisor	String	Info VM	"Hyper-V"

Data server yang diimpor

Server yang diimpor tidak ditemukan secara otomatis. Mereka diimpor melalui file CSV. Alat penemuan tidak mengumpulkan data tingkat hypervisor untuk server yang diimpor. Sebaliknya, ia mengumpulkan database, jaringan, dan data metrik OS dengan menggunakan kredensyal OS yang terkait dengan setiap server selama impor.

OS-related Data alat penemuan

Alat penemuan mengumpulkan inventaris server, kinerja, penyimpanan, antarmuka jaringan, dan data proses melalui SSH (Linux) dan WinRM (Windows). Tabel berikut menjelaskan titik data yang dikumpulkan.

Inventaris server (server_inventory.csv)

Menggabungkan penyediaan server (konfigurasi perangkat keras dan OS) dengan kinerja penyimpanan agregat. Dikumpulkan setiap 24 jam.

Nama	Tipe	Kategori	Nilai sampel
server_id	String	Info Server	“vm-web-server-01"
server_name	String	Info Server	“web-server-01"
resource_type	String	Info Server	“virtual_mesin”
power_state	String	Info Server	“Berlari”
jenis os_	String	Info Server	“Linux”
os_nama	String	Info Server	“Amazon Linux”
os_versi	String	Info Server	“2023"
primary_hostname	String	Info Server	“web-server-01.example.com”
primer_ip_address	String	Info Server	“10.0.2.101"
netmask	String	Info Server	“255.255.255.0"
total_num_network_cards	Bilangan Bulat	Info Server	2
total_num_disk	Bilangan Bulat	Info Server	1
cpu_count	Bilangan Bulat	Info Server	4

Nama	Tipe	Kategori	Nilai sampel
total_memory_gb	Desimal	Info Server	15.88
server_uuid	String	Info Server	“4201ecf8-cc44-ee7e-01da-34 dfb2acf6c0"
smbios_uuid	String	Info Server	“4201ecf8-cc44-ee7e-01da-34 dfb2acf6c0"
cluster_name	String	Info Server	“kluster-produksi-01"
hypervisor_object_id	String	Info Server	“vm-30920"
hypervisor_type	String	Info Server	“VMware”
hypervisor_version	String	Info Server	“8.0.0"
hypervisor_hostname	String	Info Server	“esxi-node1.example.com”
hypervisor_host_id	String	Info Server	“host-1234"
hypervisor_id	String	Info Server	“4201ecf8-cc44-ee7e-01da-34 dfb2acf6c0"
disk_read_iops_avg	Desimal	Kinerja Penyimpanan	12,5
disk_read_iops_peak	Desimal	Kinerja Penyimpanan	245.0
disk_write_iops_avg	Desimal	Kinerja Penyimpanan	8.3
disk_write_iops_peak	Desimal	Kinerja Penyimpanan	180.0
disk_total_iops_avg	Desimal	Kinerja Penyimpanan	20.8

Nama	Tipe	Kategori	Nilai sampel
disk_total_iops_peak	Desimal	Kinerja Penyimpanan	425.0
disk_read_throughput_avg_mbps	Desimal	Kinerja Penyimpanan	1.2
disk_read_throughput_peak_mbps	Desimal	Kinerja Penyimpanan	24,5
disk_write_throughput_avg_mbps	Desimal	Kinerja Penyimpanan	0,8
disk_write_throughput_peak_mbps	Desimal	Kinerja Penyimpanan	18.0
disk_total_throughput_avg_mbps	Desimal	Kinerja Penyimpanan	2.0
disk_total_throughput_peak_mbps	Desimal	Kinerja Penyimpanan	42.5

Metrik kinerja server (server_performance_metrics.csv)

CPU, memori, dan pemanfaatan throughput jaringan. Sampel setiap 10 menit, dikumpulkan selama 30 hari.

Nama	Tipe	Kategori	Nilai sampel
server_id	String	Info Server	“vm-web-server-01"
data_sumber	String	Info Server	“OS”
cpu_utilization_avg_pct	Desimal	CPU	45.06
cpu_utilization_peak_pct	Desimal	CPU	79.33

Nama	Tipe	Kategori	Nilai sampel
cpu_count	Bilangan Bulat	CPU	4
memory_total_gb	Desimal	Memori	15.88
memory_utilization_avg_pct	Desimal	Memori	29.27
memory_utilization_peak_pct	Desimal	Memori	63,99
network_in_avg_mbps	Desimal	Jaringan	0,52
network_in_peak_mbps	Desimal	Jaringan	12.3
network_out_avg_mbps	Desimal	Jaringan	0,31
network_out_peak_mbps	Desimal	Jaringan	8.7
network_total_avg_mbps	Desimal	Jaringan	0,83
network_total_peak_mbps	Desimal	Jaringan	21.0

Kinerja penyimpanan (server_storage_performance.csv)

Per-volume pemanfaatan disk I/O dan ruang. Sampel setiap 10 menit, dikumpulkan selama 30 hari.

Nama	Tipe	Kategori	Nilai sampel
server_id	String	Info Server	“vm-web-server-01"
data_sumber	String	Info Server	“OS”
disk_volume_id	String	Info Volume	"/dev/nvme0n1p1"
disk_mount_point	String	Info Volume	"/"
file_sistem	String	Info Volume	“xfs”
disk_total_gb	Desimal	Ruang Disk	30.0

Nama	Tipe	Kategori	Nilai sampel
disk_used_gb	Desimal	Ruang Disk	12,5
disk_free_gb	Desimal	Ruang Disk	17.5
disk_read_iops_avg	Desimal	Disk I/O	12,5
disk_read_iops_peak	Desimal	Disk I/O	245.0
disk_write_iops_avg	Desimal	Disk I/O	8.3
disk_write_iops_peak	Desimal	Disk I/O	180.0
disk_total_iops_avg	Desimal	Disk I/O	20.8
disk_total_iops_peak	Desimal	Disk I/O	425.0
disk_read_throughput_avg_mbps	Desimal	Throughput Disk	1.2
disk_read_throughput_peak_mbps	Desimal	Throughput Disk	24,5
disk_write_throughput_avg_mbps	Desimal	Throughput Disk	0,8
disk_write_throughput_peak_mbps	Desimal	Throughput Disk	18.0
disk_total_throughput_avg_mbps	Desimal	Throughput Disk	2.0
disk_total_throughput_peak_mbps	Desimal	Throughput Disk	42.5

Konfigurasi penyimpanan (storage_config.csv)

Detail perangkat keras disk fisik. Dikumpulkan setiap 24 jam.

Nama	Tipe	Kategori	Nilai sampel
server_id	String	Info Server	“vm-web-server-01"
disk_controller_id	String	Info Disk	"/dev/sda"
vmdk_vhd_file_name	String	Info Disk	“web-server-01.vmdk”
disk_volume_type	String	Info Disk	“Virtual”
disk_provisioned_gb	Desimal	Info Disk	30.0
disk_device_type	String	Info Disk	“SCSI HDD”
disk_interface_type	String	Info Disk	“SCSI”
disk_protocol	String	Info Disk	“LSI Logika SAS”

Antarmuka jaringan (network_interfaces.csv)

Konfigurasi adaptor jaringan. Dikumpulkan setiap 24 jam.

Nama	Tipe	Kategori	Nilai sampel
server_id	String	Info Server	“vm-web-server-01"
antarmuka_nama	String	Info Antarmuka	“et0"
antarmuka_index	Bilangan Bulat	Info Antarmuka	2
alamat mac_	String	Info Antarmuka	“0A:1B:2C:3D:4E:5F”
adapter_type	String	Info Antarmuka	“vmxnet3"

Nama	Tipe	Kategori	Nilai sampel
virtual_network_name	String	Info Antarmuka	“Jaringan VM”
virtual_network_id	String	Info Antarmuka	“dvportgroup-1234"
virtual_switch	String	Info Antarmuka	“VSwitch0"
ipv4_alamat	String	Konfigurasi IP	“10.0.2.101"
ipv4_subnet_mask	String	Konfigurasi IP	“255.255.255.0"
pintu gerbang ipv4_	String	Konfigurasi IP	“10.0.2.1"
ipv6_alamat	String	Konfigurasi IP	“fe80: :a1b: 2cff:fe3d: 4e5f”
ipv6_prefix_length	Bilangan Bulat	Konfigurasi IP	64
pintu gerbang ipv6_	String	Konfigurasi IP	“fe80: :1"
dns_server	String	Konfigurasi IP	“10.0.0.2"
dhcp_enabled	Boolean	Konfigurasi IP	false
antarmuka_status	String	Info Antarmuka	“Naik”
vlan_id	Bilangan Bulat	Info Antarmuka	100

Nama	Tipe	Kategori	Nilai sampel
adalah_primer	Boolean	Info Antarmuka	true

Menjalankan proses (process_metrics.csv)

Cuplikan proses yang sedang berjalan. Dikumpulkan setiap jam, dideduplikasi selama 30 hari.

Nama	Tipe	Kategori	Nilai sampel
server_id	String	Info Server	“vm-web-server-01"
process_name	String	Info Proses	“sshd”
process_id	Bilangan Bulat	Info Proses	1234
process_command_line	String	Info Proses	“/usr/sbin/SSHD -D”
process_user	String	Info Proses	“akar”

Koleksi jaringan

Modul pengumpulan Jaringan membantu Anda menemukan dependensi antar server di pusat data lokal Anda. Data jaringan ini mempercepat perencanaan migrasi Anda dengan memberikan visibilitas tentang bagaimana aplikasi berkomunikasi di seluruh server.

Modul ini mengumpulkan data jaringan untuk server dari semua sumber yang dikonfigurasi, termasuk VMware, Hyper-V, dan server yang diimpor. Ini menggunakan WinRM untuk mengumpulkan data dari server Windows dan menggunakan SSH, SNMPv2, dan SNMPv3 untuk mengumpulkan data dari server Linux.

Pengumpulan data jaringan

Modul koleksi Jaringan menangkap koneksi TCP IPv4 dalam status DITETAPKAN atau TIME_WAIT antar server dalam inventaris yang Anda temukan. Koneksi muncul di output hanya ketika alamat IP sumber dan target milik server yang ditemukan alat penemuan atau yang telah Anda impor. Koneksi

ke atau dari alamat IP di luar inventaris Anda — seperti layanan eksternal, titik akhir cloud, atau server yang belum ditambahkan ke alat penemuan — tidak disertakan.

Desain ini memfokuskan data jaringan pada dependensi server-ke-server dalam lingkungan Anda, yang merupakan informasi yang diperlukan untuk pemetaan ketergantungan aplikasi dan perencanaan gelombang migrasi.

Titik data ini dikumpulkan untuk setiap koneksi:

Nama	Tipe	Kategori	Nilai sampel
IP sumber	String	Koneksi	“192.168.1.10"
Port sumber	Bilangan Bulat	Koneksi	49152
ID proses sumber	Bilangan Bulat	Proses	1234
Nama proses sumber	String	Proses	“Jawa”
Target IP	String	Koneksi	“192.168.1.20"
Pelabuhan target	Bilangan Bulat	Koneksi	5432
ID proses target	Bilangan Bulat	Proses	5678
Nama proses target	String	Proses	“postgres”
Status	String	Koneksi	“DIDIRIKAN”
Protokol transportasi	String	Koneksi	“TCP”
Versi IP	String	Koneksi	“IPv4"
Hitungan	Bilangan Bulat	Koneksi	42

> **Tip**
>
> Untuk memaksimalkan kelengkapan peta ketergantungan jaringan Anda, konfigurasikan semua sumber penemuan (VMware Hyper-V, dan impor CSV server) dan tambahkan kredensi OS sebelum meninjau data jaringan. Semakin banyak server dalam inventaris Anda, semakin banyak koneksi yang dapat ditangkap modul jaringan.

Pengumpulan jaringan alamat pribadi

Secara default, modul koleksi Jaringan hanya menangkap koneksi di mana kedua titik akhir adalah server dalam inventaris yang Anda temukan. Anda dapat mengaktifkan pengumpulan alamat pribadi untuk juga menangkap koneksi ke dan dari alamat IP pribadi RFC 1918 (10.0.0. 0/8, 172.16.0. 0/12, 192.168.0. 0/16) yang tidak ada dalam inventaris Anda.

Untuk memulai pengumpulan alamat pribadi

1. Pada halaman konfigurasi Kolektor, cari bagian Modul koleksi.
2. Temukan penemuan koneksi jaringan di bawah Penemuan aplikasi.
3. Buka dropdown Actions.
4. Pilih Mulai pengumpulan alamat pribadi.
5. Jika status modul Diaktifkan, Anda dapat melihatnya berubah menjadi Diaktifkan · Alamat Pribadi aktif.

Untuk menghentikan pengumpulan alamat pribadi, buka dropdown Tindakan dan pilih Hentikan pengumpulan alamat pribadi. Data alamat pribadi yang dikumpulkan sebelumnya disimpan bahkan setelah berhenti.

Koneksi alamat pribadi hanya muncul di ekspor CSV penuh (di dalam file ZIP), bukan di file MPA CSV. Jika alamat IP milik server yang sudah ada di inventaris Anda, itu selalu diidentifikasi oleh ID server yang ditemukan terlepas dari pengaturan ini.

Pengaturan ini tetap ada di seluruh restart. Anda dapat memulai atau menghentikan pengumpulan alamat pribadi kapan saja. Data alamat pribadi yang dikumpulkan sebelumnya diekspor terlepas dari pengaturan saat ini.

Koleksi SQL Server

Modul pengumpulan SQL Server mengumpulkan informasi SQL Server dari server Windows di semua sumber yang dikonfigurasi, termasuk VMware,, dan server yang diimpor. Hyper-V Modul ini menggunakan WinRM untuk terhubung dari jarak jauh ke setiap server Windows dan menjalankan kueri. PowerShell Ini mengumpulkan informasi tentang semua layanan SQL Server yang diinstal (komponen) dengan menggunakan ruang nama WMI, registri, dan properti file.

Komponen SQL Server adalah layanan atau contoh fitur tertentu yang diinstal sebagai bagian dari penyebaran SQL Server pada server Windows. Alat penemuan mengumpulkan Mesin Basis Data, Layanan Analisis, Layanan Pelaporan, dan Layanan Integrasi.

Pengumpulan data SQL Server

Modul koleksi SQL Server mengumpulkan informasi komponen SQL Server. Tabel ini menjelaskan poin data utama yang dikumpulkan:

Nama	Tipe	Kategori	Nilai sampel
Tipe Mesin	String	Komponen	sql_server
Adalah Komponen Mesin	Boolean	Komponen	Y
Status	String	Layanan	Berlari, Berhenti, StartPending
Versi	String	Layanan	2015.131.5026.0
Edisi	String	Layanan	Edisi Pengembang (64-bit)
Nama Layanan SQL	String	Layanan	MsDtsServer130, Mssql
Jenis Layanan SQL	String	Layanan	Layanan SQL Server, Layanan Layanan Integrasi
Nama Instance	String	Instans	MSSQLSERVER
Nama Tampilan	String	Layanan	SQL Server (MSSQLSER VER2017)
Mode Mulai	String	Layanan	Otomatis, Manual, Dinonakti fkan

Nama	Tipe	Kategori	Nilai sampel
Nama Akun Layanan	String	Layanan	NT Service/MsDtsServer130
Berkerumun	Boolean	Konfigurasi	T

> **Note**
>
> Format lengkap mencakup semua jenis layanan. Format MPA hanya mencakup komponen mesin database. Tidak semua bidang tersedia tergantung pada jenis dan konfigurasi layanan SQL.

Koleksi Oracle Database

Dengan koleksi Oracle Database, Anda dapat menemukan instance database Oracle di semua sumber yang dikonfigurasi, termasuk VMware, Hyper-V dan server yang diimpor. Modul ini mengumpulkan data berikut:

- Database kontainer (CDB) dan penghitungan database pluggable (PDB)
- Versi dan edisi
- Opsi yang diinstal
- Statistik penggunaan fitur
- Inventaris komponen (DBA_REGISTRY)
- Ukuran Datafile
- Bendera topologi (RAC, Data Guard, dan arsitektur multitenant)

Database-connected (SQL)

Saat Anda mengonfigurasi kredensyal Oracle, alat penemuan terhubung langsung ke database Oracle menggunakan akun layanan read-only. Ini menyediakan SQL-level koleksi lengkap termasuk detail PDB, penggunaan fitur, dan opsi yang diinstal. Semua file CSV berisi data lengkap.

OS-level mundur

Jika Anda belum mengonfigurasi kredensyal database, atau jika koneksi gagal, alat penemuan menggunakan SSH atau WinRM untuk mendeteksi instalasi Oracle. Ini menemukan rumah Oracle,

pendengar, tambalan, versi, dan edisi tanpa akses database. Untuk OS-detected host, CDB CSV yang diekspor berisi nama instance, nama host, versi, dan edisi. PDB, fitur, opsi, dan komponen file CSV tidak mengandung baris.

Koleksi Oracle menghasilkan file CSV berikut di ZIP ekspor. Setiap tabel menjelaskan titik data yang dikumpulkan per file.

Data CDB (oracle_data_cdbs_full.csv)

Satu baris per instance CDB. Tabel ini menjelaskan titik data CDB yang dikumpulkan:

Nama	Tipe	Kategori	Nilai sampel	Sumber
Nama Instance	String	Identitas	“ORCL”	SQL, OS
Nama Host	String	Identitas	“oracledb01.example.com”	SQL, OS
Nama DB	String	Identitas	“ORCL”	SQL saja
Nama Unik DB	String	Identitas	“ORCL_PRIMARY”	SQL saja
DBID	Bilangan Bulat	Identitas	1234567890	SQL saja
Versi	String	Versi	“19.0.0.0"	SQL, OS
Versi penuh	String	Versi	“19.21.0.0.0"	SQL saja
Edisi	String	Versi	“Edisi Perusahaan”	SQL, OS
Jenis Database	String	Konfigurasi	“TUNGGAL”	SQL saja
Peran Database	String	Konfigurasi	“UTAMA”	SQL saja
Mode Terbuka	String	Konfigurasi	“BACA TULIS”	SQL saja
Mode Log	String	Konfigurasi	“ARCHIVELOG”	SQL saja
Platform	String	Konfigurasi	“Linux x86 64-bit”	SQL saja
Bendera CDB	String	Konfigurasi	“YA”	SQL saja

Nama	Tipe	Kategori	Nilai sampel	Sumber
Adalah RAC	String	Topologi	“YA” atau “TIDAK”	SQL saja
Apakah Data Guard Siaga	String	Topologi	“YA” atau “TIDAK”	SQL saja
Mode Perlindungan	String	Topologi	“KINERJA MAKSIMUM”	SQL saja
Pialang Penjaga Data	String	Topologi	“DIAKTIFKAN”	SQL saja
Karakter NLS	String	Konfigurasi	“AL32UTF8"	SQL saja
Hitungan PDB	Bilangan Bulat	Penyesuaian ukuran	3	SQL saja
Kilas Balik Aktif	String	Konfigurasi	“YA”	SQL saja
Jalur Deteksi	String	Metadata	“phase2" atau “phase1b”	SQL, OS

Data PDB (oracle_data_pdbs_full.csv)

Satu baris per PDB (database pluggable). Tabel ini menjelaskan titik data PDB yang dikumpulkan:

> **Note**
>
> Data PDB membutuhkan kredensi database Oracle (koneksi SQL). OS-level fallback tidak mengisi CSV ini.

Nama	Tipe	Kategori	Nilai sampel	Sumber
Nama Instans CDB	String	Identitas	“ORCL”	SQL saja
Nama PDB	String	Identitas	“APPPDB1"	SQL saja
Mode Terbuka	String	Status	“BACA TULIS”	SQL saja

Nama	Tipe	Kategori	Nilai sampel	Sumber
Status Siklus Hidup	String	Status	“NORMAL”	SQL saja
Hitungan Tablespace	Bilangan Bulat	Penyesuaian ukuran	5	SQL saja
Hitungan Datafile	Bilangan Bulat	Penyesuaian ukuran	12	SQL saja
Total Ukuran Byte	Bilangan Bulat	Penyesuaian ukuran	5368709120	SQL saja
Hitungan Skema Pengguna	Bilangan Bulat	Penyesuaian ukuran	8	SQL saja
Jumlah Tautan DB	Bilangan Bulat	Konektivitas	2	SQL saja
Komponen Terpasang	String	Konfigurasi	“PUNCAK; JVM; XML'	SQL saja
Hitungan Tablespace Terenkripsi	Bilangan Bulat	Keamanan	1	SQL saja

Data penggunaan fitur (oracle_data_features_full.csv)

Satu baris per entri penggunaan fitur dari DBA_FEATURE_USAGE_STATISTICS. Tabel ini menjelaskan titik data penggunaan fitur yang dikumpulkan:

> **Note**
>
> Data penggunaan fitur memerlukan kredensyal database Oracle (koneksi SQL). OS-level fallback tidak mengisi CSV ini.

Nama	Tipe	Kategori	Nilai sampel	Sumber
Nama Instans CDB	String	Identitas	“ORCL”	SQL saja
Nama	String	Fitur	“Partisi (pengguna)”	SQL saja
Penggunaan Terdeteksi	Bilangan Bulat	Penggunaan	42	SQL saja
Saat ini Digunakan	String	Penggunaan	“BENAR”	SQL saja
Tanggal Penggunaan Pertama	DateTime	Penggunaan	“2024-01-15T00:00:00”	SQL saja
Tanggal Penggunaan Terakhir	DateTime	Penggunaan	“2026-06-01T00:00:00”	SQL saja

Data opsi (oracle_data_options_full.csv)

Satu baris per entri V $ OPTION per CDB. Tabel ini menjelaskan titik data opsi terinstal yang dikumpulkan:

> **Note**
>
> Data opsi memerlukan kredensi database Oracle (koneksi SQL). OS-level fallback tidak mengisi CSV ini.

Nama	Tipe	Kategori	Nilai sampel	Sumber
Nama Instans CDB	String	Identitas	“ORCL”	SQL saja
Nama Opsi	String	Opsi	“Analisis Lanjutan”	SQL saja
Dipasang	String	Opsi	“BENAR”	SQL saja
ID Kontainer	Bilangan Bulat	Opsi	0	SQL saja

Data komponen (oracle_data_components_full.csv)

Satu baris per komponen DBA_REGISTRY. Tabel ini menjelaskan titik data komponen yang dikumpulkan:

> **Note**
>
> Data komponen memerlukan kredensyal database Oracle (koneksi SQL). OS-level fallback tidak mengisi CSV ini.

Nama	Tipe	Kategori	Nilai sampel	Sumber
Nama Instans CDB	String	Identitas	“ORCL”	SQL saja
Nama PDB	String	Identitas	“APPPDB1"	SQL saja
ID Komponen	String	Komponen	“PUNCAK”	SQL saja
Nama Komponen	String	Komponen	“Aplikasi Oracle Ekspres”	SQL saja
Versi	String	Komponen	“22.1.0.15.0"	SQL saja
Status	String	Komponen	“VALID”	SQL saja
Skema	String	Komponen	“APEX_220100"	SQL saja

Pertimbangan keamanan

Mengamankan alat penemuan VM

Alat penemuan Keamanan VM sangat penting karena alat penemuan menyimpan semua kredensyal, log, dan data pelanggan pada alat penemuan VM.

Alat penemuan VM memiliki akses ssh dinonaktifkan secara default dan hanya dapat diakses dari klien. `vCenter UI -> "Launch Web Console"`

Alat penemuan VM dilengkapi dengan kata sandi login default “kata sandi” untuk “penemuan” pengguna.

- Kami menyarankan Anda memperbarui kata sandi ini segera setelah penerapan.
- Kami meminta Anda untuk memperbarui kata sandi jika Anda ingin mengaktifkan ssh menggunakan perintah `enablessh` setelah masuk menggunakan vCenter`Launch Web Console`. Harap dicatat, setiap kali perintah ini dipanggil, Anda perlu mengatur ulang.

Mengamankan Kredensyal

Praktik terbaik umum untuk Manajemen Kredensi

- Simpan kredensyal dengan aman
- Putar semua kredensyal secara teratur
- Gunakan pengelola kata sandi atau brankas aman
- Pantau penggunaan kredensi
- Ikuti prinsip hak istimewa terkecil dan hanya berikan izin minimum yang diperlukan

Kredensyal SNMP v2

- Gunakan string komunitas yang kompleks dan non-default
- Hindari string umum seperti “publik” atau “pribadi”
- Perlakukan string komunitas seperti kata sandi

Kredensyal SNMP v3

- Aktifkan otentikasi dan privasi
- Gunakan protokol otentikasi yang kuat (SHA lebih disukai daripada MD5)
- Gunakan protokol enkripsi yang kuat (AES lebih disukai daripada DES)
- Gunakan kata sandi yang kompleks untuk autentikasi dan privasi
- Gunakan nama pengguna yang unik (hindari nama umum)

Kredensyal WinRM

- Hindari menonaktifkan pemeriksaan sertifikat WinRM.
- Kami menyarankan Anda membuat akun layanan khusus dengan izin minimal yang diperlukan.

- Hindari menggunakan administrator domain atau akun administrator lokal kecuali Anda memerlukan koleksi SQL Server. Koleksi SQL Server memerlukan akses Administrator Lokal karena menanyakan beberapa ruang nama WMI dan menggunakan perintah yang ditinggikan. Untuk metrik OS tanpa penemuan SQL Server, akun non-administrator dengan Pengguna Manajemen Jarak Jauh, Pengguna Monitor Kinerja, dan akses baca WMI sudah cukup. `root\cimv2` Lihat the section called “Izin yang diperlukan” untuk detail per modul.

Hyper-V credentials

- Gunakan akun layanan khusus dengan izin Hyper-V manajemen minimal.
- Hindari menggunakan akun administrator domain.
- Hyper-V kredensyal mendukung NTLM (hanya HTTPS) dan otentikasi Kerberos.
- Alat penemuan menyimpan kredensyal yang dienkripsi saat istirahat dengan menggunakan SqlCipher.

Kredensyal Oracle

- Gunakan akun layanan hanya-baca khusus dengan hanya SELECT_CATALOG_ROLE. Jangan gunakan hak istimewa DBA, SYSDBA, atau SYSOPER.
- Alat penemuan hanya melakukan operasi baca dan tidak pernah menulis ke database Oracle.
- Alat penemuan tidak mengakses tampilan Paket Diagnostik atau Paket Tuning, jadi tidak diperlukan lisensi Oracle tambahan.
- Putar kata sandi akun layanan Oracle secara teratur dan perbarui kredensialnya di alat penemuan.

Penyimpanan kredensi

Alat penemuan mengenkripsi kredensyal yang disimpan saat istirahat menggunakan kunci enkripsi basis data. Pada sistem dengan systemd 250 atau yang lebih baru, kunci ini dienkripsi menggunakan systemd-creds. Pada sistem yang lebih lama, kunci disimpan sebagai file yang dilindungi izin. Dalam kedua kasus tersebut, penyerang dengan akses root ke host alat penemuan dapat mengakses kunci enkripsi dan mendekripsi kredensyal yang disimpan. Batasi akses ke host alat penemuan dan perlakukan sebagai sistem istimewa di lingkungan Anda.

Menggunakan Auto-Connect Fitur Dengan Hati-hati

Alat penemuan menggunakan dua mekanisme untuk menetapkan kredensyal ke server selama OS-level pengumpulan: koneksi otomatis dan manual. OS-level koleksi termasuk Jaringan, SQL Server, Oracle Database, dan modul metrik OS. Modul-modul ini terhubung ke server individual dari semua sumber, termasuk VM VMware, VM, dan server yang Hyper-V diimpor.

Manual: server dapat dikaitkan secara manual dengan kredensi tertentu. Dalam hal ini, alat penemuan hanya menggunakan kredensi itu, terlepas dari keberhasilan atau kegagalan. Anda harus memantau status pengumpulan server secara manual dan melakukan penyesuaian.

Auto-connect: jika tidak ada kredensi yang dikaitkan secara manual dengan server, alat penemuan menggunakan mekanisme koneksi otomatis untuk server itu. Itu berarti:

- Pada awal setiap putaran pengumpulan, alat penemuan mendapatkan daftar kredensyal yang tersedia untuk server tersebut (berdasarkan jenis OS) dan juga dikonfigurasi menjadi “dapat dihubungkan secara otomatis”.
- Alat penemuan kemudian menguji semua kredensyal terhadap server dalam satu lingkaran.
  - Jika kredensi kerja ditemukan, putaran pengumpulan untuk server berhasil. Alat penemuan mengingatnya dan mencobanya pertama kali.
  - Jika tidak ada kredensi kerja yang ditemukan, putaran pengumpulan untuk server gagal.
    - Modul jaringan: server menggunakan jadwal backoff, memulai putaran pengumpulan berikutnya dalam 3 menit, 30 menit, 2 jam, dan 6 jam setelah setiap kegagalan (mirip dengan backoff eksponensial).
    - Koleksi SQL Server: Alat penemuan tidak mencoba lagi. Itu membuat satu upaya per server setiap hari.

Impacts/Risks:

Hanya gunakan koneksi otomatis ketika Anda yakin bahwa risiko dikurangi di sistem Anda:

Risiko 1: Dengan koneksi otomatis yang dikonfigurasi pada beberapa kredensyal yang salah, mencobanya secara otomatis terhadap server dapat memicu penguncian akun untuk lingkungan produksi tempat kebijakan penguncian dikonfigurasi. Misalnya, pusat data dapat mengatur VM-nya untuk dikunci setelah 3 upaya login SSH gagal. Dalam hal ini, jika koneksi otomatis dikonfigurasi untuk 3 kredensyal SSH yang salah, penguncian akun yang sah akan terjadi. Jika penguncian terjadi di beberapa sistem, proses bisnis penting dapat terpengaruh, berpotensi menyebabkan kegagalan

berjenjang dalam sistem dependen. Selain itu, Pusat Operasi Keamanan dapat mengalami badai peringatan dari peristiwa kegagalan otentikasi massal, menciptakan insiden keamanan positif palsu yang menguras sumber daya dan dapat menutupi serangan nyata.

Risiko 2: Aktor dengan akses ke alat penemuan (mengetahui kata sandi alat penemuan) dapat memaksa kredensyal OS di semua server, dengan mengonfigurasi sejumlah besar kredensyal pengujian dan menggunakan koneksi otomatis untuk menemukan yang berhasil.

Mitigasi:

Ikuti pedoman berikut:

- Pastikan kata sandi alat penemuan diamankan dengan benar dan hanya diketahui oleh aktor yang berwenang
- Pastikan kredensyal yang tepat dimasukkan untuk lingkungan jika kebijakan penguncian diberlakukan. Kami merekomendasikan hanya mengonfigurasi kredensyal yang diketahui dan berfungsi meskipun kebijakan penguncian akun tidak ada untuk memastikan beban operasional minimal pada VM individu.
- “Auto-connect” adalah fitur opt-in. Jangan memilihnya dan gunakan penetapan kredenal manual jika penguncian akun menjadi perhatian lingkungan.

Keamanan Impor CSV

Saat Anda mengimpor server menggunakan file CSV, pertimbangkan implikasi keamanan berikut:

- File CSV mungkin berisi nama host atau alamat IP server internal. Perlakukan itu sebagai data sensitif.
- Referensi `os_credential_name` dan `oracle_credential_name` kolom kredensyal yang telah dikonfigurasi sebelumnya dengan nama ramah. CSV tidak mengandung rahasia.
- All-or-nothing validasi: jika ada baris di CSV yang tidak valid, seluruh unggahan ditolak. Ini mencegah impor sebagian yang dapat membuat status yang membingungkan.
- Server yang diimpor segera terlihat di inventaris dan memenuhi syarat untuk dikumpulkan. Pastikan kredensyal OS dan Oracle tercakup dengan benar sebelum Anda mengimpor.

Mencabut Pertimbangan Akses

Saat Anda mencabut akses, penghapusan dicakup ke sumber tertentu:

- Mencabut akses vCenter hanya menghapus data vCenter. Itu tidak mempengaruhi Hyper-V atau mengimpor data server.
- Mencabut Hyper-V akses hanya menghapus data. Hyper-V Itu tidak mempengaruhi VMware atau data server yang diimpor.
- Menghapus server yang diimpor menghapusnya dari inventaris, tetapi data pengumpulan hilir (jaringan, database, metrik OS) dipertahankan.
- Untuk menghapus semua data inventaris khusus sumber, Anda harus mencabut atau menghapus setiap sumber secara independen. Data pengumpulan hilir (jaringan, database, metrik OS) dipertahankan bahkan setelah semua sumber dicabut.

Pemecahan masalah

Memverifikasi konektivitas alat penemuan ke vCenter

Saat Anda mengalami kesalahan konfigurasi modul VMware, ikuti langkah-langkah berikut untuk memverifikasi konektivitas:

Akses alat penemuan VM

- Log-in ke alat penemuan VM, buka Remote Console di vCenter
  - Nama Pengguna: discovery
  - Kata sandi: kata sandi

Uji Konektivitas vCenter

1. Uji Akses API vCenter:

```
curl -v --insecure -u <username>:<password> https://<vcenter-ip-or-hostname>:443/
mob
```

2. Output Sukses yang Diharapkan:

```
[ec2-user@discoverytool ~]$ curl -v --insecure -u <user>:<password> https://vcsa/
mob > tmp.txt
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
```

```
  0     0    0     0    0     0      0      0 --:--:-- --:--:-- --:--:--     0*
 Trying 192.168.2.125:443...
* Connected to vcsa (192.168.2.125) port 443 (#0)
...
</xml>
* Connection #0 to host vcsa left intact
```

Uji Sertifikat SSL

1. Jalankan perintah ini:

```
openssl s_client -showcerts -servername <hostname> -connect <hostname>:443
```

2. Output Sukses yang Diharapkan:

   - Harus menunjukkan rincian sertifikat vSphere
   - Memverifikasi SSL/TLS konektivitas pada port 443

```
[ec2-user@discoverytool ~]$ openssl s_client -showcerts -servername vcsa -connect
 vcsa:443
CONNECTED(00000003)
depth=0 CN = vcsa.onpremsim.env, C = US
verify error:num=20:unable to get local issuer certificate
verify return:1
depth=0 CN = vcsa.onpremsim.env, C = US
verify error:num=21:unable to verify the first certificate
verify return:1
---
Certificate chain
 0 s:/CN=vcsa.onpremsim.env/C=US
   i:/CN=CA/DC=vsphere/DC=local/C=US/ST=California/O=vcsa.onpremsim.env/OU=VMware
 Engineering
-----BEGIN CERTIFICATE-----
...
-----END CERTIFICATE-----
---
Server certificate
subject=/CN=vcsa.onpremsim.env/C=US
issuer=/CN=CA/DC=vsphere/DC=local/C=US/ST=California/O=vcsa.onpremsim.env/OU=VMware
 Engineering
```

```
---
```

Pemecahan Masalah WinRM

Jika Anda mengalami masalah konektivitas dengan WinRM, ikuti langkah-langkah berikut untuk menguji koneksi. Langkah-langkah ini juga berlaku untuk masalah Hyper-V konektivitas, karena alat penemuan menggunakan WinRM untuk berkomunikasi dengan Hyper-V host.

Uji konektivitas WinRM dasar menggunakan port 5985 (HTTP) dan 5986 (HTTPS). Kita perlu memastikan bahwa konektivitas berfungsi pada port 5986 (HTTPS)

```
# Check WinRM listener configuration
winrm enumerate winrm/config/listener

# Note: Replace <HOST> with the target computer's hostname or IP address. Adjust the
 username and password as needed.
# Test WinRM connection on port 5985 (HTTP)
$cred = Get-Credential
Test-WSMan -Computer <HOST> -Authentication Negotiate -Credential $cred -Port 5985

# Test WinRM connection on port 5986 (HTTPS)
Test-WSMan -Computer <HOST> -Authentication Negotiate -Credential $cred -Port 5986
```

Jika tes di atas gagal, coba buat PowerShell sesi dengan validasi sertifikat dinonaktifkan:

```
$cred = Get-Credential
$so = New-PsSessionOption -SkipCACheck -SkipCNCheck -SkipRevocationCheck
Enter-PSSession -ComputerName <HOST> -Credential $cred -Port 5985 -SessionOption $so
```

Pemecahan masalah Kerberos

Jika Anda mengalami kegagalan otentikasi Kerberos saat mengumpulkan data dari server Windows, gunakan bagian berikut untuk mendiagnosis dan menyelesaikan masalah umum.

Verifikasi persyaratan jaringan

Sebelum Anda memecahkan masalah otentikasi Kerberos, verifikasi bahwa alat penemuan dapat mencapai titik akhir jaringan yang diperlukan.

Untuk memverifikasi persyaratan jaringan untuk Kerberos

1. Verifikasi resolusi DNS ke pengontrol domain. Jalankan perintah berikut dari alat penemuan VM:

   ```
   nslookup dc01.example.com
   ```

   Atau, Anda dapat menggunakan`dig`:

   ```
   dig dc01.example.com
   ```

   Jika resolusi DNS gagal, verifikasi bahwa alat penemuan VM dikonfigurasi untuk menggunakan server DNS yang dapat menyelesaikan domain Direktori Aktif Anda. Periksa `/etc/resolv.conf` dan konfirmasikan bahwa entri nameserver mengarah ke server DNS domain Anda.
2. Verifikasi konektivitas ke Pusat Distribusi Kunci (KDC) di port 88. Jalankan perintah berikut:

   ```
   nc -zv dc01.example.com 88
   ```

   Keluaran yang diharapkan

   ```
   Connection to dc01.example.com 88 port [tcp/kerberos] succeeded!
   ```

   Jika koneksi gagal, verifikasi bahwa tidak ada aturan firewall yang memblokir lalu lintas dari alat penemuan VM ke pengontrol domain pada port 88.
3. Verifikasi konektivitas ke server Windows target pada port WinRM. Jalankan perintah berikut:

   ```
   nc -zv <windows-server> 5985
   nc -zv <windows-server> 5986
   ```

   Jika koneksi gagal, verifikasi bahwa WinRM diaktifkan di server target dan aturan firewall memungkinkan lalu lintas masuk pada port 5985 dan 5986.

Masalah umum Kerberos

Berikut ini adalah masalah umum Kerberos dan solusinya.

Kesalahan sensitivitas kasus

Gejala: Anda menerima kesalahan “Server tidak ditemukan di database Kerberos” selama otentikasi.

Kesalahan ini biasanya terjadi ketika nama ranah Kerberos tidak dalam huruf besar. Alam Kerberos harus ditentukan dalam huruf besar di file Anda. `krb5.conf` Misalnya, gunakan `EXAMPLE.COM` sebagai ganti dari `example.com`.

Pencegahan penguncian akun

Alat penemuan menggunakan mekanisme backoff untuk mencegah penguncian akun dari upaya otentikasi yang gagal berulang. Jika akun layanan Anda terkunci, Anda dapat mengatur ulang proses pengumpulan dengan menghentikan dan kemudian memulai modul pengumpulan melalui UI web tool penemuan di`https://<discovery-tool-vm-ip>:5000`.

kinit gagal dari CLI

Tabel berikut mencantumkan `kinit` kesalahan umum dan solusinya.

Kesalahan	Penyebab	Solusi
Tidak dapat menemukan KDC untuk ranah	Nama host KDC atau alamat IP tidak dapat dijangkau, atau ranah tidak dikonfigurasi. `krb5.conf`	Verifikasi bahwa `krb5.conf` file tersebut berisi nama host dan ranah KDC yang benar. Konfirmasikan resolusi DNS dan konektivitas jaringan ke KDC pada port 88.
Pra-otentikasi gagal	Kata sandi untuk akun layanan tidak benar.	Verifikasi kata sandi dan coba lagi. Jika akun terkunci, buka kunci di Active Directory sebelum Anda mencoba lagi.
Klien tidak ditemukan di database Kerberos	Nama utama tidak cocok dengan akun apa pun di Active Directory.	Verifikasi bahwa nama utama sama persis dengan nama akun, termasuk kasus. Gunakan format `username@REALM` dengan ranah dalam huruf besar.
Tidak dapat menyelesaikan alamat jaringan untuk KDC	DNS tidak dapat menyelesaikan nama host KDC.	Verifikasi konfigurasi DNS di`/etc/resolv.conf` . Konfirmasikan bahwa server DNS dapat menyelesaikan

Kesalahan	Penyebab	Solusi
		nama host KDC. Uji dengan `nslookup` atau`dig`.

Koleksi gagal meskipun kinit berhasil

Jika `kinit` berhasil tetapi pengumpulan data masih gagal, periksa hal berikut:

1. Verifikasi bahwa nama utama yang digunakan untuk koleksi cocok dengan kasus yang digunakan selama `kinit` persis.
2. Verifikasi bahwa akun layanan memiliki izin yang diperlukan pada server target.
3. Verifikasi bahwa WinRM diaktifkan di server target.
4. Verifikasi bahwa nama host yang digunakan untuk koleksi cocok dengan nama host yang terdaftar di Active Directory.

Kerberos berfungsi untuk beberapa server tetapi tidak yang lain

Jika otentikasi Kerberos berhasil untuk beberapa server tetapi gagal untuk yang lain, selidiki area berikut:

Jika server Anda menjangkau beberapa domain Active Directory, konfigurasikan kredensi Kerberos terpisah untuk setiap domain. Pastikan `/etc/krb5.conf` file Anda menyertakan entri untuk semua ranah. Setiap domain membutuhkan kredensialnya sendiri dengan `username@REALM` prinsip yang benar.

Bandingkan konfigurasi WinRM di server yang berfungsi dengan server yang gagal. Jalankan perintah berikut di setiap server:

```
winrm get winrm/config
```

Uji konektivitas dengan Remote Desktop untuk mengisolasi masalah. Alat penemuan menggunakan format`username@DOMAIN`, sedangkan Remote Desktop menggunakan format`DOMAIN\username`.

Verifikasi bahwa akun layanan adalah anggota grup Administrator lokal di server yang gagal. Jalankan perintah berikut di server target:

```
net localgroup Administrators
```

WMI memerlukan hak administrator lokal untuk mengakses informasi sistem operasi. Koleksi SQL Server juga mengharuskan akun layanan memiliki akses administrator lokal pada server target.

Daftar periksa konfigurasi Kerberos

Gunakan daftar periksa berikut untuk memverifikasi konfigurasi Kerberos Anda sebelum Anda memulai pengumpulan data.

- `krb5.conf`File ada di alat penemuan VM.
- Nama ranah dalam huruf besar di. `krb5.conf`
- Berjalan `kinit` dengan akun layanan berhasil tanpa kesalahan.
- Running `klist` menunjukkan tiket yang valid dan tidak kedaluwarsa.
- Nama utama sama persis dengan nama akun Active Directory.
- Resolusi DNS berfungsi untuk nama host KDC.
- Konektivitas jaringan ke KDC pada port 88 dikonfirmasi.
- Konektivitas jaringan untuk menargetkan server Windows pada port 5985 dan 5986 dikonfirmasi.
- (Multi-domain) Setiap domain Active Directory memiliki kredensialnya sendiri yang dikonfigurasi dalam alat penemuan, `[realms]` dan `krb5.conf` berisi serta `[domain_realm]` entri untuk semua domain.

Pemecahan masalah Oracle Database

Untuk mendiagnosis masalah pengumpulan Database Oracle, seperti data atau kesalahan yang hilang, periksa yang berikut ini.

Koneksi ditolak atau batas waktu

Gejala: Status koleksi Oracle menunjukkan kesalahan koneksi untuk server.

Untuk memecahkan masalah ini, periksa hal berikut:

- Verifikasi bahwa pendengar Oracle berjalan pada host target: `lsnrctl status`
- Verifikasi konektivitas jaringan dari alat penemuan ke host Oracle pada port 1521 (atau port khusus Anda): `nc -zv <oracle-host> 1521`
- Verifikasi bahwa aturan firewall memungkinkan koneksi masuk pada port pendengar Oracle.
- Verifikasi nama layanan dengan berjalan `lsnrctl services` di host Oracle. Jika nama layanan salah, pendengar Oracle menolak koneksi.

Kegagalan otentikasi () ORA-01017

Gejala: Pengumpulan gagal dengan kesalahan nama pengguna atau kata sandi yang tidak valid.

Untuk memecahkan masalah ini, periksa hal berikut:

- Verifikasi bahwa akun layanan Oracle ada dan tidak terkunci: `SELECT account_status FROM dba_users WHERE username = 'DISCOVERY_USER';`
- Verifikasi kata sandi sudah benar dengan menghubungkan secara manual: `sqlplus discovery_user/<password>@<host>:1521/<service_name>`
- Jika akun terkunci, buka kunci: `ALTER USER discovery_user ACCOUNT UNLOCK;`

Hak istimewa yang tidak memadai () ORA-01031

Gejala: Koneksi berhasil tetapi pengumpulan mengembalikan data yang tidak lengkap.

Untuk memecahkan masalah ini, periksa hal berikut:

- Verifikasi bahwa SELECT_CATALOG_ROLE diberikan: `SELECT * FROM dba_role_privs WHERE grantee = 'DISCOVERY_USER';`
- Berikan peran yang diperlukan jika tidak ada: `GRANT SELECT_CATALOG_ROLE TO discovery_user;`

Kredensi manual menunjukkan kesalahan tetapi koneksi otomatis berfungsi

Saat Anda menyematkan kredensi secara manual ke server, alat penemuan tidak akan mundur jika koneksi gagal. Verifikasi bahwa port dan nama layanan yang Anda konfigurasikan pada kredensi cocok dengan pendengar Oracle di server tertentu. Jika server memiliki port atau nama layanan non-standar, perbarui konfigurasi kredensialnya.

OS-level fallback tidak mendeteksi Oracle

Jika tidak ada kredensyal database yang dikonfigurasi dan OS-level fallback tidak mendeteksi Oracle:

- Verifikasi bahwa kredensyal SSH atau WinRM OS dikonfigurasi dan berfungsi untuk server (periksa status pengumpulan metrik OS).
- Untuk host Linux, verifikasi yang `/etc/oratab` ada atau proses Oracle Process Monitor (`pmon`) sedang berjalan.

- Untuk host Windows, verifikasi bahwa entri registri Oracle ada di bawah `HKLM\SOFTWARE\Oracle` atau bahwa `oracle.exe` proses sedang berjalan.

Pemecahan Masalah SNMP

Akses alat penemuan VM

- Log-in ke alat penemuan VM, buka Remote Console di vCenter
  - Nama Pengguna: discovery
  - Kata sandi: kata sandi

Instal SNMP Tools (jika diperlukan)

- `sudo yum install net-snmp-utils -y`

Uji Koneksi SNMP ke Server Linux

1. `snmptable -v 2c -c <COMMUNITY_STRING> <REMOTE_SERVER_IP> .1.3.6.1.2.1.6.13.1`
2. Contoh:

```
#SNMPv2c:
snmptable -v 2c -c public 192.168.1.100 .1.3.6.1.2.1.6.13.1

#SNMPv3 (with authentication):
snmptable -v 3 -u <username> -a MD5 -A <auth_password>
 192.168.1.100 .1.3.6.1.2.1.6.13.1

#SNMPv3 (with privacy):
snmptable -v 3 -u <username> -a MD5 -A <auth_password> -x DES -X <priv_password>
 192.168.1.100 .1.3.6.1.2.1.6.13.1
```

Kesalahan pengumpulan jaringan

terminal diperlukan untuk membaca kata sandi

Kesalahan:

```
ss command failed on <host>: sudo: a terminal is required to read the password; either
 use the -S option to read from standard input or configure an askpass helper sudo: a
 password is required
```

Perintah ss meminta kata sandi pengguna. Pengguna ssh yang dikonfigurasi harus berada dalam grup sudoers dan dikonfigurasi dengan sudo tanpa kata sandi untuk perintah tersebut. ss/netstat Untuk mengonfigurasi sudo tanpa kata sandi:

1. Buat file sudoers baru:

   ```
   sudo vi -f /etc/sudoers.d/<username>
   ```

2. Tambahkan baris:

   ```
   <username> ALL=(ALL) NOPASSWD: /usr/sbin/ss, /usr/bin/netstat
   ```

3. Setelah perubahan ini, jalankan `sudo ss -tnap` dan `sudo netstat -tnap` harus dijalankan tanpa meminta kata sandi

Koleksi jaringan berjalan tanpa sudo

Jika Anda melihat peringatan berikut di halaman inventaris Ditemukan:

```
Network collection ran without sudo. Process-level connection data may be missing.
```

Peringatan ini menunjukkan bahwa akun pengguna SSH tidak memiliki akses sudo pada server target. Tanpa sudo, alat penemuan masih dapat mengumpulkan data koneksi jaringan, tetapi tidak dapat menentukan proses mana yang memiliki setiap koneksi. Untuk mengumpulkan data koneksi tingkat proses yang lengkap, pastikan pengguna SSH memiliki akses sudo di server target.

Kesalahan pengumpulan metrik OS

Server UUID tidak ada untuk server Linux

Jika alat penemuan tidak dapat mengumpulkan UUID server (ditampilkan sebagai kosong atau hilang) untuk server Linux, verifikasi bahwa kredensyal SSH yang dikonfigurasi untuk server tersebut memiliki hak istimewa sudo. Alat ini digunakan `dmidecode` untuk membaca UUID server. Jika `dmidecode` tidak diinstal, alat akan kembali membaca`/sys/class/dmi/id/product_uuid`, yang juga membutuhkan akses sudo. Tanpa sudo, tidak ada metode yang dapat mengambil UUID.

Resolusi: Pastikan akun pengguna SSH yang disediakan untuk alat penemuan memiliki akses sudo pada server Linux target.

Masalah akses di Inventaris yang ditemukan

Jika Anda melihat pesan dalam status pengumpulan Server seperti Kredensyal tidak ada, atau Akses ditolak:

1. Pilih server pada tabel server yang ditemukan.
2. Pilih Kelola kredensi akses Anda dapat memilih untuk:
    a. Pilih kredensyal alternatif dari menu tarik-turun Pilih kredensyal.
    b. Pilih Gunakan kredensyal baru dan berikan kredensyal baru.
3. Simpan.

Alat penemuan mencoba ulang koneksi setelah Anda menyimpan perubahan.

Pemecahan masalah otentikasi kunci SSH

Menguji konektivitas kunci SSH dari alat penemuan

Jika otentikasi kunci SSH gagal, verifikasi konektivitas dari alat penemuan ke server target:

1. Masuk ke alat penemuan (melalui konsol vSphere atau SSH ke host Linux).
2. Uji konektivitas SSH menggunakan kunci pribadi Anda:

```
ssh -i /path/to/private_key -o StrictHostKeyChecking=no <username>@<target_ip>
```

3. Jika koneksi berhasil, masalahnya adalah bagaimana kunci diunggah ke alat penemuan. Re-upload kunci dan verifikasi nama pengguna cocok.
4. Jika koneksi gagal, periksa pesan kesalahan di tabel berikut.

Pesan kesalahan	Penyebab	Resolusi
`Permission denied (publickey)`	Kunci publik tidak ada di `authorized_keys` file server	Tambahkan kunci publik ke `~/.ssh/authorized_keys` server target untuk pengguna yang benar. Verifikasi izin file:`chmod`

Pesan kesalahan	Penyebab	Resolusi
	target, atau nama pengguna salah.	`700 ~/.ssh && chmod 600 ~/.ssh/authorized_keys .`
`Connection timed out after 20s`	Port 22 tidak dapat dijangkau dari alat penemuan.	Verifikasi bahwa port 22 terbuka antara alat penemuan dan server target. Periksa firewall dan grup keamanan.
`Connection refused`	Layanan SSH tidak berjalan di server target.	Mulai layanan SSH:`sudo systemctl start sshd`.
`Invalid SSH key for credential 'name'`	Format kunci tidak didukung, data kunci rusak, atau frasa sandi hilang atau salah.	Verifikasi format kunci adalah RSA, ECDSA, atau Ed25519 dalam format PEM, OpenSSH, atau PKCS #8. Jika kunci dienkripsi, pastikan frasa sandi sudah benar.

Pemecahan masalah penginstal Linux

Port 5000 sudah digunakan

Gejala: Layanan alat penemuan gagal dimulai setelah instalasi.

Resolusi: Identifikasi dan hentikan proses menggunakan port 5000:

```
sudo ss -tlnp | grep :5000
```

Hentikan proses yang bertentangan, lalu mulai ulang alat penemuan:

```
sudo ./AWS-Transform-discovery-tool.sh start
```

Pesan kesalahan umum

Tabel ini menjelaskan pesan kesalahan umum dan penjelasannya:

Pesan	Lokasi	Penjelasan
Kata sandi telah dibuat	Buat halaman kata sandi	Kondisi balapan saat dua pengguna membuat kata sandi secara bersamaan ; segarkan
Ekspor gagal	Halaman inventaris	Coba lagi atau kirim log
Koleksi sesuai permintaa n sudah berlangsung	Halaman inventaris	Kondisi balapan ketika dua pengguna memulai koleksi manual secara bersamaan; coba lagi setelah pengumpulan manual saat ini selesai
`Command timed out after 60s`	Status pengumpulan server	Perintah pada server target tidak selesai dalam waktu 60 detik. Ini dapat terjadi pada server yang dimuat dengan berat. Coba lagi koleksi atau selidiki beban server target.
Satu atau lebih kredensil berisi UUID yang tidak dikenal	Halaman akses OS	Kondisi balapan ketika dua pengguna mengedit kredensyal OS secara bersamaan; coba lagi
Kata sandi tidak valid	Sign-in halaman	Kata sandi yang salah untuk masuk; hubungi admin atau hubungi
Sesi Anda telah kedaluwarsa. Silakan masuk lagi.	Sign-in halaman	Sesi telah habis, perlu login lagi
Terjadi kesalahan internal	Berbagai halaman	Coba lagi atau kirim log

Catatan rilis

Bagian ini menjelaskan fitur, peningkatan, dan perbaikan bug untuk alat AWS Transform penemuan, yang diatur berdasarkan tanggal rilis.

Date	Versi	Catatan rilis
30 Juni 2026	1.0.1646	Fitur baru • Menambahkan penemuan Oracle Database. Alat ini mengumpulkan topologi CDB dan PDB, versi, edisi, opsi yang diinstal, penggunaan fitur, dan inventaris komponen. Alat penemuan mendukung koneksi SQL langsung (menggunakan SELECT_CATALOG_ROLE) dan deteksi fallback melalui SSH atau WinRM. OS-level • Menambahkan dukungan untuk beberapa kredensya l Kerberos di berbagai domain Active Directory. Setiap kredensi mengautentikasi secara independen, memungkinkan pengumpulan dari server Windows di beberapa domain tanpa konflik otentikasi. Perbaikan • Peningkatan koleksi metrik OS untuk host lama. Menambahk an pengumpulan data fallback untuk sistem operasi yang lebih lama (RHEL 5, SLES 11, Windows Server 2008 R2) di mana perintah modern tidak tersedia. Penyimpanan, kinerja, dan data antarmuka jaringan sekarang dikumpulkan menggunakan antarmuka sistem alternatif. Perbaikan bug • Memperbaiki kegagalan otentikasi Kerberos untuk server yang ditemukan melalui. Hyper-V Alat penemuan sekarang dengan benar menyelesaikan nama host server (FQDN) dari metadata Hyper-V tamu, yang diperlukan untuk pencocokan SPN Kerberos.
14 Mei 2026	1.0.1363	Fitur baru • Menambahkan penginstal Linux untuk menerapkan alat penemuan pada distribusi Linux yang didukung (Amazon

Date	Versi	Catatan rilis
		Linux 2, Amazon Linux 2023, RHEL 8—9, Rocky Linux 8—9, 9, Ubuntu 20.04—24.04, Debian 10—12, SLES AlmaLinux 15 SP5). • Menambahkan dukungan untuk mengkonfigurasi beberapa server VMware vCenter dengan deduplikasi VM otomatis di seluruh vCenters. • Menambahkan otentikasi kunci pribadi SSH sebagai alternati f untuk nama pengguna dan kata sandi untuk akses OS Linux. Format kunci yang didukung termasuk RSA, ECDSA, Ed25519, OpenSSH, dan PKCS #8. • Menambahkan koleksi jaringan alamat pribadi untuk menangkap koneksi ke dan dari alamat IP pribadi RFC 1918 yang tidak ada dalam inventaris yang Anda temukan. • Menambahkan opsi ekspor untuk memilih rentang tanggal kustom dan memilih modul data mana yang akan disertakan dalam ekspor. Perbaikan • Peningkatan keandalan koleksi terjadwal dengan jadwal kolektor yang mengejutkan. • Peningkatan kinerja pengumpulan metrik OS melalui batching perintah SSH. • Perubahan kredenal sekarang segera coba lagi server yang sebelumnya gagal. • Koleksi metrik OS sekarang melaporkan status sebagian ketika beberapa perintah mengembalikan hasil kosong. • Peningkatan visibilitas kesalahan untuk masalah koneksi vCenter di dasbor.

Date	Versi	Catatan rilis
April 7, 2026	1.0.1074	Fitur baru • Menambahkan dukungan untuk Hyper-V penemuan Microsoft , termasuk penemuan otomatis VM dari Hyper-V host dengan deduplikasi cluster failover. • Menambahkan dukungan untuk impor server melalui unggahan file CSV. • Menambahkan koleksi metrik OS dengan enam modul independen: kinerja server, kinerja penyimpanan, penyediaa n server, penyediaan penyimpanan, antarmuka jaringan, dan proses yang berjalan. • Menambahkan opsi Hyper-V penyebaran menggunakan file gambar VHD. • Peningkatan pengerasan keamanan lokal dengan menghapus EC2-specific layanan dan memblokir akses ke titik akhir metadata instans.
Maret 26, 2026	1.0.1014	Perbaikan bug dan perbaikan • Memindahkan validasi kapitalisasi domain Kerberos dari backend ke frontend. • Beralih dari Python-kerberos ke Python-pykerberos dan menjadikan cache berbasis file sebagai default. • Memperbaiki validasi ID server edit massal yang tidak peka huruf besar/kecil. • Nama file ZIP ekspor tetap (sekarang`discovery _tool_export.zip`). • Layanan tetap menjalankan pemeriksaan dan menambahkan log debug untuk kredensyal sistem. • Ditambahkan skrip konfigurasi jaringan untuk membangun artefak.

Date	Versi	Catatan rilis
Desember 16, 2025	1.0.718	Fitur dan peningkatan baru • Menambahkan dukungan SMBIOS UUID untuk mencegah duplikasi VM di layanan hilir. • Memperkenalkan metrik Min/Max/Avg pelacakan dan penyimpanan yang diperluas untuk pemantauan sumber daya yang lebih baik. • Menambahkan perbaikan validasi API. • Ditambahkan fungsi panel bantuan dengan panduan kontekstual. • Terminologi yang diperbarui dari “kolektor” ke “alat” di seluruh antarmuka.
Oktober 31, 2025	1.0.0	Rilis awal Peluncuran awal alat AWS Transform penemuan. Rilis ini meliputi: • Koleksi offline (tidak ada ketergantungan pada AWS akun atau Application Discovery Service). • Penemuan dan pemantauan VMware. • Penemuan database SQL Server. • Server-to-server pengumpulan komunikasi melalui SSH, WMI, dan SNMP.

AWS Transform Konektor

AWS Konektor Transform memungkinkan Anda mengakses sumber daya dengan aman di seluruh batas akun untuk alur kerja migrasi dan modernisasi. Ini memungkinkan Anda mengakses dan mengelola sumber daya di seluruh batas akun sambil mempertahankan kontrol dan izin keamanan. Konektor mewakili koneksi antara akun AWS Transform-enabled sumber Anda dan sumber daya eksternal di akun AWS target atau di sistem pihak ketiga.

Konektor memerlukan permintaan akses dari konektor sumber, dan persetujuan dari pemilik akun target. Anda membuat permintaan ini ketika Anda menyiapkan konektor sumber. Demikian pula, Anda mungkin menerima permintaan untuk menyetujui konektor tujuan untuk mengizinkan akses akun lain ke akun Anda.

- Pembuatan konektor: Buat konektor di akun AWS Transform-enabled sumber Anda, tentukan akun target dan izin yang diperlukan
- Pengaturan AWS Izin: Transform memerlukan peran dan izin IAM tertentu di akun tujuan untuk melakukan tindakan migrasi, termasuk:
  - Mengelola data migrasi dengan AWS KMS
  - Membuat panggilan API lintas wilayah
  - Menginstal agen replikasi di server VMware
  - Membuat infrastruktur jaringan (VPC, subnet, routing)
  - Meluncurkan instans Amazon EC2 dan menerapkan tumpukan CloudFormation
  - Menjalankan alur kerja migrasi melalui Migration Hub
- Proses Persetujuan: Anda harus menyetujui permintaan konektor tujuan sebelum AWS Transform dapat mengakses sumber daya Anda.
- Koneksi Aktif: Setelah Anda menyetujui konektor, ini memungkinkan AWS Transform jobs untuk mengakses dan mengelola sumber daya dengan aman di akun target.

Keterlacakan pengguna untuk operasi agen

Secara default, AWS CloudTrail merekam tindakan agen pada AWS sumber daya Anda di bawah identitas layanan AWS Transform. Keterlacakan pengguna mengaitkan tindakan tersebut kepada pengguna Pusat AWS Identitas IAM tertentu yang memulai atau berinteraksi dengan pekerjaan, memungkinkan audit keamanan dan investigasi insiden di tingkat individu-pengguna.

> **Note**
>
> Keterlacakan pengguna hanya didukung untuk pengguna IAM Identity Center (IDC).

Untuk mengaktifkan keterlacakan pengguna, aktifkan sesi Latar Belakang di pengaturan profil AWS Transform Anda. Saat diaktifkan, AWS Transform secara otomatis membuat sesi latar belakang Pusat Identitas IAM saat Anda berinteraksi dengan pekerjaan dan menggunakannya untuk menghasilkan sesi peran IAM yang disempurnakan identitas untuk operasi konektor. Sesi ini menyematkan identitas pengguna Anda dalam panggilan dan AWS CloudTrail acara AWS layanan hilir. Saat sesi latar belakang dinonaktifkan, AWS Transform akan kembali ke sesi peran yang diasumsikan standar tanpa konteks identitas pengguna.

Konektor baru secara otomatis dikonfigurasi untuk mendukung keterlacakan pengguna. Untuk konektor yang ada, Anda harus menambahkan secara manual `sts:SetContext` ke kebijakan kepercayaan peran konektor:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": { "Service": ["transform.amazonaws.com"] },
      "Action": ["sts:AssumeRole", "sts:SetContext"],
      "Condition": {
        "StringEquals": { "aws:SourceAccount": "your-account-id" }
      }
    }
  ]
}
```

Anda dapat melihat dan mencabut sesi latar belakang aktif Anda dari tab Sesi latar belakang pada halaman pekerjaan. Mencabut sesi segera membatalkannya dan sesi konektor apa pun yang berasal darinya.

Untuk informasi selengkapnya, lihat ikhtisar propagasi identitas tepercaya di Panduan Pengguna Pusat AWS Identitas IAM.

Mengelola konektor

Pada halaman Konektor Anda dapat melihat:

- Tab Konektor Sumber: Daftar semua konektor yang telah Anda buat untuk terhubung ke akun lain, menampilkan akun target dan status konektor
- Tab Konektor Tujuan: Menampilkan permintaan konektor masuk dari akun lain yang ingin mengakses sumber daya Anda, dan memerlukan persetujuan atau penolakan Anda

AWS Transform memerlukan konfirmasi untuk tindakan penting seperti menyetujui, menolak, atau menghapus konektor, untuk menghindari perubahan tidak disengaja yang dapat mengganggu alur kerja migrasi aktif Anda.

Untuk membuat konektor

1. Arahkan ke tab Source Connectors.
2. Tentukan akun target dan jenis sumber daya.
3. Konfigurasikan izin dan ruang lingkup akses yang diperlukan.

Untuk mengatur izin

1. Pilih untuk membuat peran IAM baru dengan izin yang diperlukan, atau
2. Pilih peran yang sudah ada yang sebelumnya Anda buat untuk konektor.
3. Pastikan peran memiliki semua izin yang diperlukan untuk operasi AWS Transform.

Untuk mengelola permintaan masuk

1. Tinjau permintaan konektor di tab Konektor Tujuan.
2. Menyetujui permintaan yang sah untuk mengaktifkan akses lintas akun.
3. Tolak upaya koneksi yang tidak sah atau tidak perlu.

Untuk mengelola konektor Anda

1. Pantau konektor aktif Anda untuk keamanan dan kepatuhan.
2. Hapus konektor saat Anda tidak lagi membutuhkannya. Perhatikan bahwa ini akan menyebabkan pekerjaan yang berjalan menggunakan konektor gagal.

3. Perbarui izin saat persyaratan Anda berubah.

Migrasi (termasuk VMware)

AWS Transform dapat membantu Anda memigrasikan lingkungan server virtual dan bare metal ke Amazon EC2 dengan menggunakan AI generatif, termasuk Hyper-V VMware,, dan sumber lainnya. Bab ini memberikan gambaran umum tentang kemampuan migrasi AWS Transform dan alur kerja proses migrasi.

Kemampuan dan fitur utama

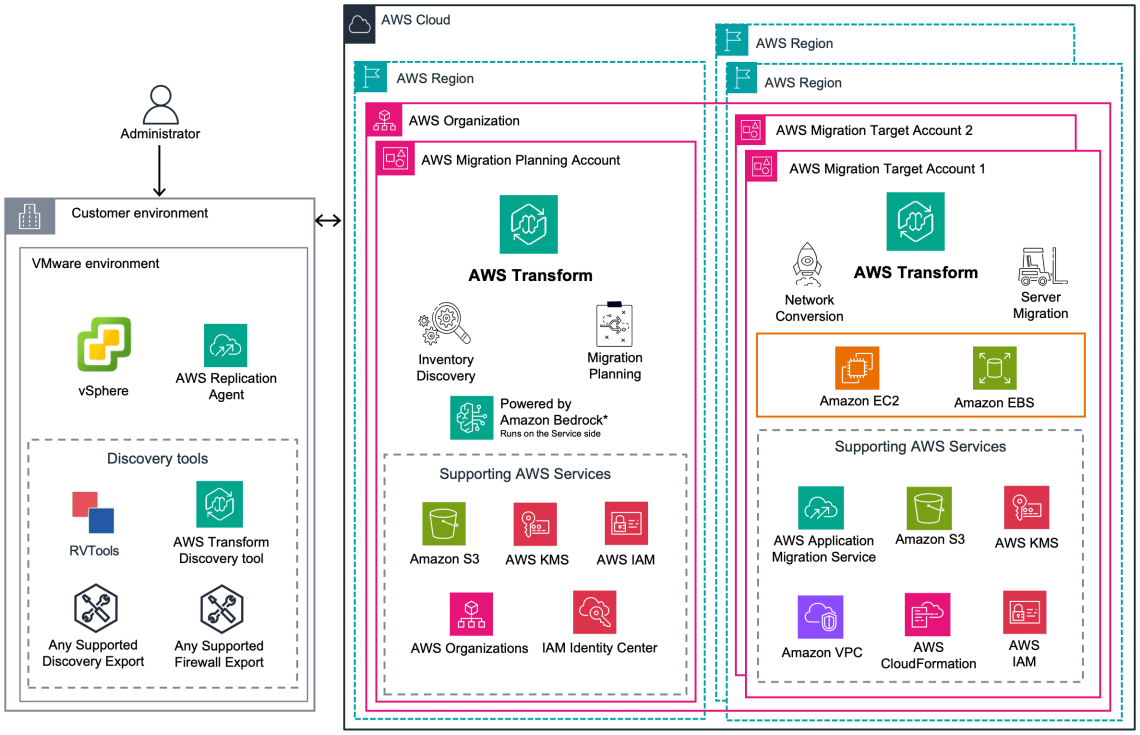
AWS Transform menawarkan kemampuan dan fitur utama berikut untuk memigrasikan lingkungan server Anda ke AWS.

- Beberapa opsi penemuan:
  - Membantu penemuan lingkungan VMware Anda dengan menggunakan kolektor dari. AWS Application Discovery Service
  - Gunakan alat Export for vCenter open-source (lingkungan VMware).
  - Mengimpor data penemuan yang dikumpulkan secara independen (lingkungan sumber apa pun).
- AI-driven konversi konfigurasi jaringan sumber Anda ke arsitektur jaringan Amazon VPC.
- AI-driven pembuatan rencana migrasi, termasuk pengelompokan aplikasi dan gelombang migrasi yang disarankan.
- Rehosting server Anda agar berjalan secara native di Amazon EC2.
- Antarmuka aplikasi web yang dilokalkan, memungkinkan Anda menggunakan AWS Transform untuk migrasi dalam bahasa pilihan Anda. Untuk informasi selengkapnya, lihat Pengaturan bahasa.

AWS Transform mendukung migrasi server Windows dan Linux dari sistem operasi yang didukung. Untuk daftar lengkap sistem operasi yang didukung, lihat Sistem operasi yang didukung di Panduan AWS Transform MGN Pengguna.

AWS Mengubah arsitektur migrasi

Diagram ini menampilkan ikhtisar arsitektur migrasi AWS Transform.



Lowongan kerja migrasi

Untuk menggunakan AWS Transform for migrations, pertama-tama Anda memerlukan ruang kerja, yang merupakan wadah logis tempat Anda dapat membuat satu atau beberapa pekerjaan transformasi. Bagian dalam topik ini menjelaskan cara mendapatkan ruang kerja dan cara membuat dan memulai pekerjaan migrasi di dalamnya.

Mendapatkan ruang kerja

Untuk informasi tentang mendapatkan ruang kerja, lihat Memulai.

Ruang kerja yang Anda gunakan menentukan di Wilayah AWS mana Anda dapat membuat pekerjaan transformasi. Wilayah AWS Di situlah pekerjaan Anda akan berada. Data penemuan Anda dan rekomendasi AWS Transform juga akan berada dalam hal ini Wilayah AWS. Untuk membuat ruang kerja dan pekerjaan di tempat lain Wilayah AWS, mintalah administrator Anda untuk membuat ruang kerja yang berbeda untuk Anda. Untuk informasi tentang dukungan Wilayah AWS, lihatWilayah yang Didukung.

Meskipun Wilayah AWS tempat Anda dapat membuat pekerjaan dan menyimpan data penemuan dan rekomendasi ditentukan oleh administrator AWS Transform, Anda dapat menentukan target

migrasi yang berbeda Wilayah AWS . Dengan kata lain, Anda dapat menjalankan penemuan dan menerima rekomendasi AWS Transform dalam satu Wilayah AWS, tetapi kemudian membuat lingkungan target Anda di lingkungan yang berbeda Wilayah AWS. Jika Anda melakukannya, Anda akan mentransfer data Anda. Wilayah AWS Untuk informasi selengkapnya, lihat the section called “Connect target Akun AWS s dan wilayah”.

Jenis Job

AWS Transform menawarkan jenis pekerjaan migrasi berikut yang dapat Anda pilih tergantung pada kebutuhan migrasi Anda. Selain opsi prasetel ini, Anda dapat secara dinamis menambahkan atau menghapus langkah apa pun dari pekerjaan Anda kapan saja untuk menyesuaikan alur kerja migrasi Anda.

End-to-end migrasi

1. Lakukan penemuan
2. Membangun rencana migrasi
3. Connect akun target
4. Bangun landing zone
5. Migrasikan jaringan
6. Migrasi server

Penemuan dan perencanaan migrasi

1. Lakukan penemuan
2. Membangun rencana migrasi

Migrasi jaringan

1. Connect akun target
2. Migrasikan jaringan

Zona pendaratan

1. Connect akun target
2. Bangun landing zone

Zona pendaratan, jaringan, dan migrasi server

1. Connect akun target
2. Bangun landing zone
3. Migrasikan jaringan
4. Migrasi server

Perencanaan migrasi dan migrasi server

Termasuk penemuan, rencana gelombang, dan rehost.

1. Lakukan penemuan
2. Membangun rencana migrasi
3. Connect akun target
4. Migrasi server

Membuat dan memulai pekerjaan

Langkah pertama dari proyek migrasi adalah membuat pekerjaan AWS Transform. Untuk proyek migrasi, Anda dapat memilih jenis pekerjaan yang berbeda, tergantung pada tujuan Anda. Prosedur berikut menjelaskan cara membuat dan memulai pekerjaan migrasi baru dari jenis apa pun. Untuk informasi tentang berbagai jenis pekerjaan, lihatthe section called “Jenis Job”.

Untuk membuat dan memulai pekerjaan migrasi baru

1. Di halaman landing ruang kerja Anda, pilih Buat pekerjaan.
2. Pilih opsi migrasi, lalu tentukan jenis pekerjaan migrasi yang ingin Anda buat. Untuk informasi tentang langkah-langkah yang disertakan dalam setiap jenis pekerjaan migrasi, lihatthe section called “Jenis Job”.
3. Setelah Anda menjawab semua pertanyaan obrolan, pilih Buat pekerjaan.

Mengunduh laporan ringkasan ruang kerja

Anda dapat membuat laporan ringkasan ruang kerja sebagai PDF yang dapat diunduh kapan saja selama migrasi. Laporan ini menyediakan tampilan konsolidasi di semua pekerjaan migrasi di ruang

kerja Anda, termasuk status pekerjaan dan langkah alur kerja saat ini, tindakan dan persetujuan pengguna, detail perencanaan gelombang, topologi migrasi jaringan, konfigurasi landing zone, kemajuan rehost, keputusan kontainerisasi, dan artefak utama yang dihasilkan. Untuk membuat laporan, tanyakan kepada agen dalam obrolan untuk ringkasan ruang kerja — misalnya, “Beri saya ringkasan ruang kerja tentang kemajuan migrasi saya di semua pekerjaan.” Agen mengkompilasi data dari semua pekerjaan dan mengirimkan PDF langsung dalam obrolan sebagai file yang dapat diunduh. Anda dapat menggunakan laporan ini untuk melacak kemajuan migrasi secara keseluruhan atau berbagi status dengan pemangku kepentingan.

Batasan

AWS Transform memiliki batasan sebagai berikut:

- Jika Anda menghentikan pekerjaan migrasi yang sedang berjalan, dan kemudian meminta agen untuk memulai ulang, pekerjaan akan dimulai lagi dari awal dan Anda akan kehilangan kemajuan yang telah Anda buat dalam pekerjaan itu. Namun, artefak yang dibuat dalam pekerjaan sebelum memulai ulang itu masih akan tersedia.
- Anda dapat menentukan satu target Wilayah AWS per pekerjaan migrasi. Untuk memigrasikan aplikasi ke Wilayah target yang berbeda, buat beberapa pekerjaan migrasi.
- Multi-account migrasi — Hanya wilayah tunggal — Anda dapat bermigrasi ke beberapa akun dalam satu Wilayah AWS akun. Untuk migrasi multi-wilayah, Anda harus membuat proyek terpisah untuk setiap wilayah target.
- Multi-account migrasi - Satu akun per gelombang - Setiap gelombang migrasi hanya dapat menargetkan satu akun. Aplikasi yang membutuhkan akun target yang berbeda harus ditempatkan dalam gelombang terpisah.
- Multi-account migrasi — AWS Organizations required — Semua akun target harus menjadi bagian dari AWS Organisasi.

Temukan sumber data

Untuk menemukan dan mengumpulkan data lokal, Anda dapat menggunakan alat penemuan AWS Transform, pengumpul Evaluator AWS Migrasi, atau mengunggah ekspor dari RVTools dan ModelizeIt, serta ekspor format Penilaian Portofolio AWS Migrasi (MPA) yang dihasilkan oleh alat seperti Cloudamize. AWS Transform dapat memproses ekspor alat penemuan AWS Transform sebagai file CSV dan JSON dalam format file ZIP, RVTools mengekspor baik sebagai file Excel atau file ZIP yang berisi file CSV, dan ekspor ModelizeIT CSV dalam file ZIP. Untuk setiap sumber data

AWS Transform menerima file server utama satu per satu. Namun, file tambahan seperti koneksi hanya akan diproses ketika mereka disertakan dalam file ZIP dengan file server atau dalam file Excel dengan lembar server.

Kami menyarankan Anda mengunggah data paling detail yang tersedia dan meninjau file ekspor Anda sebelum mengunggah untuk memastikan kelengkapan dan akurasi data. Verifikasi bahwa semua file yang diperlukan disertakan dalam unggahan Anda, dan konfirmasikan bahwa data tersebut mencerminkan status lingkungan Anda saat ini. Tinjauan awal ini membantu meminimalkan kebutuhan untuk mengunggah ulang dan memastikan bahwa AWS Transform dapat menangkap lingkungan lokal Anda dengan lebih akurat dan mendukung perencanaan migrasi dengan lebih baik, termasuk pengelompokan aplikasi dan perencanaan gelombang. Anda juga dapat menambahkan data secara bertahap saat Anda memperoleh data yang lebih baik.

AWS Transform secara otomatis mendeteksi format dan struktur file, mem-parsing dan mengekstrak catatan entitas terstruktur, menghapus duplikat di beberapa file, memvalidasi kualitas data dan melaporkan masalah, dan kemudian menyiapkan ringkasan yang siap untuk perencanaan migrasi hilir.

Setelah membagikan dan mengonfirmasi data lokal, Anda dapat meninjau data penemuan dengan memperluas data lokal Temukan di Rencana Pekerjaan dan memilih ringkasan kesiapan Inventaris. Anda juga dapat mengajukan pertanyaan tentang data yang Anda unggah untuk memverifikasi bahwa AWS Transform menyerap semuanya dengan benar atau untuk mengidentifikasi dan memperbaiki kesalahan apa pun dalam pemrosesan data. Misalnya, Anda dapat bertanya tentang sistem operasi dan versi. Jika Anda perlu memperbaiki kesalahan, Anda dapat mengunggah ulang data inventaris Anda dari alat penemuan yang Anda gunakan, dan AWS Transform akan secara otomatis memproses, menghapus duplikat, dan menggabungkan catatan yang diperbarui. Anda juga dapat menghapus file yang diunggah sebelumnya jika Anda tidak lagi ingin menggunakan data inventaris tersebut.

Membangun rencana migrasi

Langkah pekerjaan perencanaan Migrasi dalam AWS Transform for VMware adalah pengalaman berbasis obrolan kolaboratif untuk merencanakan migrasi besar. AWS Agen Transform menerapkan Panduan AWS Preskriptif untuk memandu pelanggan mulai dari analisis data lokal hingga rencana gelombang migrasi yang diselesaikan.

Setelah tugas Discover data lokal berhasil diselesaikan, AWS Transform menggunakan data penemuan untuk mengelompokkan aplikasi ke dalam gelombang migrasi. AWS Transform memandu

Anda melalui langkah-langkah untuk menganalisis dan cakupan server Anda, mengelompokkannya ke dalam aplikasi, membuat grup pindahan, dan membangun gelombang migrasi. Saat menganalisis lingkungan lokal, Anda dapat mengajukan pertanyaan untuk lebih memahami cara AWS Transform menganalisis perangkat lunak yang diinstal, misalnya, dependensi server dan arsitektur jaringan.

AWS Transform mendukung penyesuaian lingkup dalam kelompok aplikasi dan gelombang. Anda dapat mengunggah ulang data penemuan kapan saja, dan AWS Transform akan secara otomatis memproses, menghapus duplikasi, dan menggabungkan catatan baru dengan data yang ada. Ketika perubahan dideteksi—seperti dependensi yang baru ditemukan atau penambahan infrastruktur, AWS Transform akan menandai grup dependensi yang terkena dampak dan memberikan rekomendasi untuk penyesuaian rencana gelombang. Perencanaan migrasi juga dapat memanfaatkan data teks yang tidak terstruktur untuk memperkaya proses perencanaan.

Ada empat tahap perencanaan migrasi:

- Dalam Lingkup dan analisis, Anda dapat meninjau data penemuan Anda, mengajukan pertanyaan tentang perangkat lunak dan lingkungan jaringan Anda dan menentukan sumber daya dalam ruang lingkup untuk migrasi.
- Di aplikasi Grup, Anda dapat memberikan kombinasi aturan bisnis dan teknis terkait, misalnya, analisis nama host, dependensi jaringan, dan aturan bisnis, sehingga perencanaan migrasi dapat mengelompokkan infrastruktur Anda ke dalam aplikasi. Jika Anda sudah memiliki inventaris aplikasi Anda, perencanaan migrasi dapat menggunakannya sebagai gantinya.
- Di Hasilkan grup pindah, Anda dapat memberikan perencanaan migrasi persyaratan teknis dan bisnis Anda sehingga dapat menentukan aplikasi mana yang harus dipindahkan bersama. Dependensi teknis mencakup database, antrian pesan, atau sumber daya lain yang dibagikan di antara beberapa aplikasi. Dependensi bisnis dan operasional meliputi kekritisan bisnis, RPO dan RTO, lokasi pusat data, dan pemilik aplikasi.
- Terakhir, di gelombang Build, Anda dapat memberikan konteks perencanaan migrasi tentang jadwal dan prioritas Anda sehingga dapat membuat rencana gelombang yang dapat Anda migrasikan. Anda dapat memilih grup pindah untuk dimasukkan dalam gelombang berdasarkan faktor-faktor seperti skor prioritas, ukuran grup pemindahan, jumlah pengguna, dan kompleksitas aplikasi.

Terminologi Perencanaan Migrasi:

- Gelombang migrasi adalah kelompok logis yang bermigrasi bersama. Gelombang migrasi terdiri dari satu atau lebih kelompok bergerak.

- Grup bergerak adalah seperangkat aplikasi co-dependen yang harus dipindahkan bersama. Mereka mungkin memiliki dependensi teknis seperti database bersama atau mereka memiliki dependensi bisnis seperti mendukung fungsi bisnis bersama.
- Ketergantungan adalah hubungan antar sistem. Ada beberapa jenis dependensi termasuk:
  - Dependensi kritis atau keras, di mana sistem tidak dapat beroperasi tanpa ketergantungan. Contoh umum dari ini adalah aplikasi yang bergantung pada database, aplikasi lain, atau layanan.
  - Dependensi lunak, yang tidak penting untuk pengoperasian sistem. Contoh umum dari hal ini termasuk dependensi tidak sensitif latensi yang dapat dimigrasikan secara independen.
  - Non-technical dependensi termasuk dependensi bisnis, organisasi, operasional dan kepatuhan. Ini adalah dependensi yang terkait dengan organisasi Anda dan prioritasnya. Contohnya termasuk fungsi bisnis bersama dan kepemilikan organisasi.

Alur kerja

Perencanaan migrasi adalah alur kerja interaktif dan berulang. Anda dapat kembali dan membuat perubahan pada langkah sebelumnya kapan saja. Alur kerja perencanaan migrasi yang khas adalah:

1. Perencanaan migrasi dimulai dengan meringkas data penemuan yang tersedia. Tinjau data yang tersedia dan kembali ke langkah penemuan kapan saja untuk memberikan data tambahan.
2. Dalam langkah pelingkupan dan analisis, Anda dapat mengajukan pertanyaan tentang lingkungan lokal untuk memvalidasi data yang telah Anda kumpulkan. Contoh pertanyaan meliputi:
   a. Daftar server saya berdasarkan sistem operasi
   b. Meringkas topologi jaringan lokal saya
   c. Buat daftar teknologi paling umum yang berjalan di lingkungan saya.
3. Saat menganalisis lingkungan Anda, jika Anda mengidentifikasi server yang seharusnya tidak berada dalam cakupan migrasi, Anda dapat memberi tahu AWS Transform untuk mengecualikan sumber daya tersebut. Contohnya adalah:
   a. Hapus semua server yang memiliki warisan di nama host mereka
   b. Hapus semua server di 10.0.2. 0/24 subnet
   c. Hapus semua server yang menjalankan versi Windows yang lebih lama dari 2022
4. Setelah Anda cukup menjelajahi lingkungan dan menentukan cakupan migrasi, Anda dapat memberi tahu AWS Transform untuk pindah ke langkah perencanaan migrasi berikutnya.

5. Langkah selanjutnya adalah pengelompokan aplikasi. Jika server sudah dipetakan ke aplikasi, Anda dapat memberi tahu AWS Transform untuk menggunakan pemetaan itu dan lewati langkah ini. Jika Anda tidak memiliki aplikasi yang telah ditentukan sebelumnya, Anda dapat memberikan logika teknis dan bisnis yang mendefinisikan aplikasi Anda. AWS Transform akan memandu Anda melalui proses pengelompokan aplikasi dan menyarankan titik data yang dapat Anda berikan untuk mengelompokkan server Anda secara efektif ke dalam aplikasi. Semakin banyak informasi yang dapat Anda berikan tentang aplikasi lokal, semakin efektif AWS Transform dapat mengelompokkan server Anda ke dalam aplikasi. Setelah Anda memberikan informasi yang cukup, Anda kemudian dapat menginstruksikan AWS Transform untuk melakukan pengelompokan aplikasi.
6. Setelah pengelompokan aplikasi dilakukan, tinjau grup aplikasi. Anda dapat menginstruksikan AWS Transform untuk membuat perubahan yang diperlukan, misalnya:
   a. Pindahkan server contoh-server ke aplikasi-5
   b. Ganti nama aplikasi-5 “Lingkungan Uji Aplikasi SDM”
   c. Hapus semua server Linux dari IIS Dev Farm
7. Setelah aplikasi Anda dikelompokkan, instruksikan AWS Transform untuk pindah ke langkah berikutnya
8. Langkah selanjutnya adalah pengelompokan bergerak. Pada langkah pengelompokan langkah Anda mengidentifikasi aplikasi yang harus dipindahkan bersama. Berikan konteks seputar dependensi teknis dan non-teknis Anda. AWS Transform akan memandu Anda melalui proses dan menyarankan titik data yang dapat Anda berikan untuk mengelompokkan aplikasi Anda bersama-sama. Ada beberapa pertimbangan yang harus dibuat pada tahap ini termasuk:
   a. Apa yang harus menjadi ukuran target kelompok bergerak?
   b. Apakah Anda ingin menggabungkan lingkungan, misalnya dev, test, dan prod, untuk setiap aplikasi, atau membaginya?
   c. Bagaimana Anda ingin mempertimbangkan dependensi jaringan? Apakah semua dependensi penting atau dapatkah beberapa dependensi dianggap dependensi lunak dan dibagi menjadi grup bergerak?
9. Setelah Anda memberikan aturan untuk pengelompokan bergerak, instruksikan AWS Transform untuk menjalankan strategi pengelompokan langkah Anda. Anda kemudian dapat meninjau dan memodifikasi grup pindah Anda. Setelah meninjau grup pemindahan, Anda dapat menginstruksikan AWS Transform untuk pindah ke langkah perencanaan migrasi terakhir.
10. Perencanaan gelombang adalah langkah terakhir dalam perencanaan migrasi. Pada langkah ini, Anda mengelompokkan grup pemindahan Anda ke dalam gelombang migrasi dan

memprioritaskan gelombang tersebut. Dalam langkah perencanaan gelombang, AWS Transform akan memandu Anda melalui penyediaan prioritas bisnis yang diperlukan untuk mengelompokkan kelompok bergerak Anda ke dalam gelombang dan kemudian memprioritaskan gelombang tersebut. Pertimbangan dalam perencanaan gelombang meliputi:

a. Kekritisan bisnis dari masing-masing kelompok bergerak Anda

b. Garis waktu migrasi dan garis waktu untuk setiap grup pemindahan

c. Risiko yang terkait dengan setiap kelompok bergerak

d. Jumlah server yang akan bermigrasi per gelombang

11. Setelah Anda memberikan panduan yang cukup tentang cara mengelompokkan ke dalam gelombang, instruksikan AWS Transform untuk melaksanakan perencanaan gelombang. Anda kemudian dapat meninjau gelombang Anda dan memodifikasinya.

12. Setelah Anda menyelesaikan rencana gelombang Anda, Anda dapat menyelesaikan perencanaan migrasi dan pindah ke eksekusi. Anda dapat kembali ke perencanaan migrasi kapan saja untuk memperbaiki dan mengulangi rencana Anda.

13. Untuk setiap gelombang, Anda dapat menetapkan strategi migrasi: rehost (migrasi server ke Amazon EC2) dan containerize (containerize source code dan deploy ke Amazon Elastic Container Service atau Amazon Elastic Kubernetes Service). Saat Anda menetapkan gelombang strategi containerize, AWS Transform menjalankan alur kerja containerisasi kode sumber untuk gelombang tersebut selama eksekusi migrasi. Untuk informasi selengkapnya, lihat Kontainerisasi kode sumber. Untuk mendapatkan strategi AWS yang direkomendasikan di seluruh kerangka kerja 7R sebelum Anda menetapkannya, lihat. the section called “Rekomendasi 7Rs”

Rekomendasi strategi migrasi (7Rs)

Selama perencanaan migrasi, AWS Transform dapat menganalisis inventaris yang Anda temukan dan merekomendasikan strategi migrasi — salah satu 7R standar industri — untuk setiap server dan aplikasi. Setiap rekomendasi mencakup layanan AWS target yang disarankan, skor kepercayaan diri, dan alasan di baliknya. Ini memberi Anda titik awal yang dapat dipertahankan untuk rencana gelombang Anda dan memunculkan peluang modernisasi yang mungkin Anda lewatkan.

Tujuh strategi tersebut adalah rehost, replatform, refactor, pembelian kembali, pensiun, mempertahankan, dan relokasi. AWS Ubah sinyal pembacaan dalam data Anda — seperti sistem operasi, pemanfaatan, dan jenis beban kerja — untuk membuat rekomendasi yang dapat dijelaskan. Misalnya, dapat merekomendasikan untuk menghentikan server idle atau end-of-life, memindahkan SQL Server atau database sumber terbuka ke Amazon RDS, mengalihkan beban kerja email dan

kolaborasi ke Microsoft 365, mempertahankan sistem warisan khusus, dan menghosting ulang beban kerja yang tersisa di Amazon EC2. Rekomendasi bergulir dari server individual ke aplikasi dan di seluruh portofolio Anda, dengan pandangan kritis berdasarkan strategi dan ringkasan tingkat gelombang.

Untuk menghasilkan laporan strategi migrasi (7Rs):

1. Selesaikan langkah Temukan data lokal dan kelompokkan server Anda ke dalam aplikasi. Keduanya diperlukan agar AWS Transform dapat memberikan rekomendasi per server dan per-aplikasi.
2. Dalam obrolan perencanaan migrasi, minta laporan strategi migrasi (7R) AWS Transform—misalnya, Buat laporan. R-strategy Setelah pengelompokan aplikasi selesai, AWS Transform juga menawarkan untuk menghasilkan laporan untuk Anda.
3. AWS Transform menganalisis inventaris Anda, menerapkan kerangka kerja 7Rs, dan menghasilkan laporan. Anda dapat memberikan detail seperti nama organisasi Anda dan filter apa pun untuk cakupan analisis.
4. Tinjau rekomendasi. Untuk menyesuaikan rekomendasi, beri tahu AWS Transform dalam obrolan—ubah satu server atau aplikasi, atau terapkan perubahan secara massal (misalnya, berdasarkan sistem operasi, lingkungan, gelombang, atau prioritas). AWS Transform update laporan dengan perubahan Anda.

Secara default, AWS Transform menghasilkan dasbor HTML interaktif yang dapat Anda filter dan jelajahi. Berdasarkan permintaan, Anda juga dapat membuat dokumen PDF untuk dibagikan dengan pemangku kepentingan, atau dek geser Microsoft PowerPoint (PPTX) untuk disajikan; PDF dan PowerPoint output adalah snapshot point-in-time dari analisis yang sama. Laporan ini mencakup strategi dan layanan AWS target yang direkomendasikan untuk setiap server dan aplikasi, skor kepercayaan dan penalaran, peluang modernisasi yang disorot, pandangan kritis berdasarkan strategi, ringkasan tingkat gelombang, dan tanda risiko.

Strategi yang direkomendasikan dalam laporan ini melengkapi strategi per gelombang yang Anda tetapkan selama perencanaan gelombang. Untuk kontainerisasi kode sumber, lihat. Kontainerisasi kode sumber

Diagram dan pelaporan

Selama perencanaan migrasi, AWS Transform mengubah inventaris yang Anda temukan dan hasil perencanaan menjadi diagram visual dan laporan analitis. Anda membuatnya secara

percakapan dalam obrolan perencanaan—misalnya, “Tunjukkan diagram topologi jaringan” atau “Hasilkan laporan risiko.” AWS Transform membangunnya dari data yang telah Anda berikan, tanpa memerlukan pengaturan.

Diagram bersifat interaktif dan memungkinkan Anda menjelajahi lingkungan Anda secara visual. Diagram yang tersedia meliputi:

- Topologi jaringan — Peta interaktif server Anda dan hubungan jaringannya, yang dirancang untuk skala ke lingkungan besar ribuan node.
- Ketergantungan aplikasi — Hubungan aplikasi-ke-aplikasi terarah yang berasal dari koneksi jaringan yang Anda amati.
- Bagan Wave Gantt - Tampilan garis waktu gelombang migrasi Anda yang menunjukkan jadwal, dependensi, tonggak sejarah, dan kemajuan (tersedia setelah perencanaan gelombang).
- Bagan umum — Diagram lingkaran, diagram batang, histogram, peta pohon, dan tabel ringkasan di atas dimensi data yang Anda pilih.

Laporan memberikan analisis naratif dan rekomendasi atas data migrasi Anda. Laporan yang tersedia meliputi:

- Penilaian risiko — Menilai setiap aplikasi di seluruh kompleksitas teknis, risiko ketergantungan, kesiapan migrasi, dan risiko operasional, dengan ringkasan per aplikasi, per gelombang, dan portofolio.
- Strategi migrasi (7R) — Per-server dan rekomendasi strategi per aplikasi. Untuk informasi selengkapnya, lihat the section called “Rekomendasi 7Rs”.
- Laporan umum dan kustom — Laporan analitis untuk permintaan seperti ringkasan kesiapan eksekutif, pengorbanan biaya lawan risiko, dan penilaian kandidat kontainerisasi.

Beberapa output tergantung pada seberapa jauh Anda dalam perencanaan. Misalnya, diagram ketergantungan aplikasi menggunakan data koneksi jaringan Anda. Grafik gelombang Gantt dan ringkasan laporan tingkat gelombang tersedia setelah perencanaan gelombang selesai. Jika prasyarat hilang, AWS Transform memberi tahu Anda apa yang dibutuhkan dan dapat menjalankan langkah yang diperlukan terlebih dahulu.

AWS Transform memberikan diagram sebagai HTML interaktif yang dapat Anda filter dan jelajahi. Secara default, AWS Transform memberikan laporan sebagai file HTML mandiri. Berdasarkan permintaan, AWS Transform juga dapat menghasilkan dokumen PDF untuk dibagikan dengan

pemangku kepentingan atau dek geser Microsoft PowerPoint (PPTX) untuk disajikan. PDF dan PowerPoint output adalah snapshot point-in-time dari analisis yang sama.

Connect target Akun AWS s dan wilayah

Konfigurasikan Akun AWS konektor target Anda untuk migrasi jaringan, pembuatan landing zone, dan migrasi server. Ini melibatkan tiga langkah: pilih jenis migrasi Anda, berikan detail perjanjian MAP Anda (jika ada), dan atur konektornya. Pengaturan ini berlaku di semua tahap migrasi — migrasi jaringan, landing zone, dan server rehost.

Langkah 1: Pemilihan tipe migrasi

Pilih apakah Anda melakukan migrasi satu akun atau multi-akun:

- Single-account migrasi - Semua beban kerja bermigrasi ke satu target. Akun AWS Akun target konektor dan akun target adalah sama.
- Multi-account migrasi — Beban kerja bermigrasi ke akun target yang berbeda. Konektor harus terhubung ke akun manajemen organisasi atau akun Administrator Delegasi (DA) yang terdaftar untuk keduanya AWS Transform MGN dan CloudFormation StackSets.

Langkah 2: Perjanjian MAP

Jika migrasi Anda merupakan bagian dari AWS Migration Acceleration Program (MAP 2.0), berikan ID Migration Portfolio Experience (MPE) Anda — kode 10 karakter menggunakan huruf besar dan digit (misalnya, ABCDE12345). Saat Anda memberikan ID MPE Anda, tag MAP diterapkan ke semua sumber daya yang dibuat di seluruh tahap migrasi jaringan, landing zone, dan server rehost. Format tag adalah:

- Kunci: **`map-migrated`** Nilai: `mig`*`MPE_ID`*

Anda harus menerapkan tag MAP untuk menerima kredit MAP. Untuk informasi selengkapnya tentang MAP, lihat AWS Migration Acceleration Program.

Langkah 3: Konfigurasi konektor

Anda menggunakan konektor akun target untuk menghubungkan pekerjaan migrasi Anda ke AWS lingkungan tempat beban kerja Anda akan berada setelah migrasi. Sebelum memulai,

verifikasi bahwa target Anda Akun AWS memiliki izin, kuota, dan konfigurasi yang diperlukan untuk mendukung infrastruktur migrasi Anda.

Saat menyetujui permintaan konektor, Anda memberikan izin AWS Transform ke:

- Kelola operasi bucket Amazon S3 (read/write) untuk migrasi VMware, bersama dengan akses ke Migration Hub dan AWS Application AWS Migration Service (MGN). Ini termasuk izin untuk item berikut, semuanya dibatasi untuk sumber daya dalam akun target yang ditandai dengan `CreatedBy:AWSTransform` atau: `CreatedFor:AWSTransform`
  - Kelola gelombang migrasi.
  - Kelola konfigurasi jaringan (Amazon EC2, VPC, Transit Gateway, Direct Connect, Load Balancers, Network Firewall).
  - Kelola penerapan CloudFormation tumpukan.
  - Lakukan instalasi agen otomatis melalui Systems Manager.
- Migrasikan beban kerja lokal Anda ke target Akun AWS dan Wilayah dengan menggunakan informasi yang disimpan di Wilayah penemuan.
- Menyediakan dan mengelola infrastruktur landing zone di target Akun AWS dan Wilayah. Ini termasuk izin untuk item berikut, dibatasi untuk sumber daya yang ditandai dengan `CreatedBy:AWSTransform` jika berlaku:
  - Lakukan operasi bucket Amazon S3 (buat, baca, tulis, hapus) untuk bucket yang dimulai. `transform-vmware-landing-zone-`
  - Kelola penyebaran CloudFormation tumpukan dan ubah set untuk tumpukan landing zone.
  - Lakukan operasi AWS Control Tower. Anda dapat mengelola zona pendaratan, mengaktifkan garis dasar, dan mengaktifkan kontrol.
  - Kelola AWS Organizations. Anda dapat membuat dan mengelola unit organisasi, membuat akun, dan memindahkan akun.
  - Kelola kebijakan kontrol layanan (SCP) melalui AWS Control Tower.
  - Kelola artefak penyediaan AWS Service Catalog.

> **Note**
>
> Jenis konektor mungkin diperbarui ketika fitur baru memerlukan perubahan izin. Versi saat ini untuk jenis konektor akun target adalah 2.0. Saat Anda membuat konektor baru, ia menggunakan versi terbaru.

Sebelum menyiapkan konektor, pahami peran akun yang terlibat dalam migrasi:

Akun	Deskripsi
AWS Transformasi akun	Setiap akun anggota di AWS Organisasi tempat Anda mengatur AWS Transform. Di sinilah ruang kerja AWS Transform Anda berjalan. Tidak perlu menjadi akun manajemen.
Akun target konektor	Akun konektor AWS Transform Anda dikonfigurasi. Ini tergantung pada jenis migrasi Anda: • Single-accountmigrasi — Sambungkan ke akun tempat Anda memigrasikan beban kerja. Akun target konektor dan akun target adalah sama. • Multi-accountmigrasi — Connect ke akun manajemen organisasi atau akun Delegated Administrator (DA). Akun DA harus terdaftar sebagai administrator yang didelegasikan untuk MGN dan CloudFormation StackSets di Organisasi Anda AWS . AWS Transform memeriksa apakah akun yang terhubung adalah akun manajemen atau akun DA dan menyesuaikan perilakunya.
Akun target	Akun AWS Ke mana beban kerja Anda dimigrasikan. Dalam migrasi satu akun, ini sama dengan akun target konektor. Dalam migrasi multi-aku n, ini adalah akun anggota individu yang menerima beban kerja yang dimigrasi.

Menggunakan akun administrator yang didelegasikan

Untuk migrasi multi-akun, AWS disarankan agar Anda menggunakan akun Administrator Delegasi (DA) daripada akun manajemen organisasi secara langsung. Akun DA mengikuti prinsip hak istimewa paling sedikit dengan membatasi ruang lingkup izin yang diperlukan untuk operasi migrasi. Akun DA harus terdaftar sebagai administrator yang didelegasikan untuk MGN dan CloudFormation StackSets di Organisasi Anda AWS .

Perbedaan utama antara kedua opsi ini adalah:

- Akun manajemen - Dapat mengaktifkan akses tepercaya untuk MGN dan CloudFormation StackSets di seluruh organisasi. AWS Transform panggilan CloudFormation StackSets API dengan`CallAs: SELF`.
- Akun Administrator yang didelegasikan — Tidak dapat mengaktifkan akses tepercaya secara langsung (yang harus dilakukan dari akun manajemen), tetapi dapat mengelola server sumber MGN, meluncurkan instance, dan menyebarkan CloudFormation StackSets di seluruh akun anggota. AWS Transform panggilan CloudFormation StackSets API dengan`CallAs: DELEGATED_ADMIN`.

Untuk informasi selengkapnya, lihat Administrator yang didelegasikan untuk MGN di Panduan Pengguna MGN.

Peran IAM dibuat selama pengaturan konektor

Selama pengaturan konektor, AWS Transform membuat peran IAM berikut di akun target Anda:

- `AWSTransform-Connector-role-`— Dibuat saat Anda mengatur konektor akun target, di akun manajemen, atau akun administrator yang didelegasikan dari AWS Organisasi Anda. Peran ini memungkinkan AWS Transform untuk terhubung ke akun target Anda dan bertindak atas nama Anda untuk menjalankan operasi migrasi.

Pengaturan konektor akun target

> **Important**
>
> Selama pengaturan konektor, bucket Amazon S3 dibuat di target Anda. Akun AWS Bucket ini tidak akan menerapkan HTTPS-only access (`SecureTransport`) secara default. Jika Anda ingin kebijakan bucket menyertakan transportasi yang aman, Anda harus memperbarui sendiri kebijakan tersebut. Untuk informasi selengkapnya, lihat Praktik terbaik keamanan untuk Amazon S3.

Untuk menggunakan konektor akun target yang ada

1. Di panel Job Plan, perluas Pilih akun target, lalu pilih Buat atau pilih konektor.
2. Di tab Kolaborasi, pilih konektor yang ada dan kemudian pilih Gunakan konektor. Jika konektor tidak tersedia, versinya tidak kompatibel dengan jenis pekerjaan yang Anda pilih.

> **Important**
>
> Jika Anda menentukan konektor dengan target Wilayah AWS yang berbeda dari AWS Transform Region, AWS Transform akan mentransfer data Anda Wilayah AWS.

3. Pilih Lanjutkan.

Untuk membuat konektor baru

1. Di panel Job Plan, perluas Connect target account, lalu pilih Buat atau pilih konektor.
2. Tentukan Akun AWS dan Wilayah AWS untuk target Anda, lalu pilih Berikutnya.

> **Important**
>
> Jika target Wilayah AWS berbeda dari penemuan Wilayah AWS, AWS Transform akan mentransfer data Anda Wilayah AWS.

3. Pilih apakah akan menggunakan kunci terkelola Amazon S3 untuk enkripsi. Jika Anda menentukan kunci KMS Anda sendiri, Anda dapat menggunakan kebijakan kunci default atau yang kurang permisif. Untuk informasi tentang membuat kunci KMS, lihat Membuat kunci KMS di Panduan AWS Key Management Service Pengembang.

   AWS Transform menggunakan `kms:DescribeKey` izin untuk memverifikasi kunci yang ada, serta `kms:Decrypt` untuk mengenkripsi `kms:GenerateDataKey` dan mendekripsi data pekerjaan di bucket Amazon S3. Untuk informasi selengkapnya, lihat Mengurangi biaya SSE-KMS dengan Amazon S3 Bucket Keys.
4. Pilih Lanjutkan.
5. Salin tautan verifikasi, bagikan dengan administrator target Akun AWS, dan minta mereka untuk menyetujui permintaan koneksi.
6. Setelah administrator menyetujui permintaan, pilih konektor yang baru dibuat dari daftar dan pilih Gunakan konektor.
7. Pilih Kirim ke AWS Transformasi.

Jika Anda berencana untuk memodifikasi AWS Transform MGN template untuk mengaktifkan tindakan pasca-peluncuran, tambahkan izin berikut ke peran konektor target. Pernyataan kebijakan

JSON ini memberikan `iam:PassRole` izin untuk peran tindakan pasca-peluncuran. Anda dapat menemukan nama peran di tab Kolaborasi setelah konektor dibuat. Untuk informasi tentang menambahkan izin ke peran, lihat Memperbarui izin untuk peran dalam Panduan Pengguna IAM.

```
{
      "Sid": "MGNPostLaunchActions",
      "Effect": "Allow",
      "Action": [
        "iam:PassRole"
      ],
      "Resource": "arn:aws:iam::target-account-ID:role/service-role/
AWSApplicationMigrationLaunchInstanceWithSsmRole"
}
```

Wilayah target yang didukung

Wilayah target migrasi adalah Wilayah AWS tempat sumber daya yang dimigrasi digunakan, termasuk zona pendaratan, infrastruktur jaringan, dan rehosting server. Saat Anda membuat konektor, tentukan target Wilayah AWS. Anda dapat menggunakan salah satu dari berikut ini Wilayah AWS:

- Timur AS (N. Virginia)
- AS Timur (Ohio)
- AS Barat (California Utara)
- AS Barat (Oregon)
- Afrika (Cape Town)
- Asia Pasifik (Hong Kong)
- Asia Pasifik (Taipei)
- Asia Pasifik (Mumbai)
- Asia Pasifik (Hyderabad)
- Asia Pasifik (Tokyo)
- Asia Pasifik (Seoul)
- Asia Pasifik (Osaka)
- Asia Pasifik (Singapura)
- Asia Pasifik (Sydney)

- Asia Pasifik (Jakarta)
- Asia Pacific (Melbourne)
- Asia Pasifik (Malaysia)
- Asia Pasifik (Selandia Baru)
- Asia Pasifik (Thailand)
- Kanada (Pusat)
- Kanada Barat (Calgary)
- Eropa (Frankfurt)
- Europe (Zurich)
- Eropa (Irlandia)
- Eropa (London)
- Eropa (Paris)
- Eropa (Stockholm)
- Europe (Milan)
- Eropa (Spanyol)
- Israel (Tel Aviv)
- Meksiko (Tengah)
- Amerika Selatan (Sao Paulo)

> **Important**
>
> Jika Anda menentukan target Wilayah AWS yang berbeda dari AWS Transform Wilayah AWS, beberapa data Anda ditransfer Wilayah AWS.
> Perhatikan bahwa data replikasi server Anda langsung dari lingkungan sumber Anda ke akun dan wilayah target Anda.

Bangun landing zone

AWS Transform memandu Anda dalam merancang dan menerapkan AWS landing zone sebagai bagian dari proyek migrasi Anda. Landing zone adalah AWS lingkungan multi-akun yang berfungsi

sebagai dasar untuk beban kerja Anda dengan batasan organisasi, kontrol tata kelola, dan struktur akun sebelum beban kerja tiba. AWS Transform menganalisis inventaris migrasi dan persyaratan bisnis Anda untuk merekomendasikan Unit Organisasi (OU) dan struktur akun, menerapkan Kebijakan Kontrol Layanan (SCP) yang direkomendasikan, dan menghasilkan and/or penerapan infrastruktur sebagai kode (IAc).

Agen landing zone memandu Anda melalui dua fase:

- Pengaturan pondasi - Menetapkan struktur landing zone inti: AWS Control Tower, OU dasar, dan akun inti.
- Desain akun beban kerja — Rancang dan buat OU dan akun beban kerja berdasarkan gelombang migrasi, unit bisnis, dan persyaratan pemisahan lingkungan Anda.

AWS Transform mendukung lingkungan greenfield (tidak ada landing zone yang ada) dan lingkungan brownfield (OU yang ada dan akun sudah digunakan). Dalam skenario brownfield, AWS Transform mendeteksi struktur organisasi yang ada dan merekomendasikan hanya perubahan yang diperlukan untuk mengisi celah terhadap praktik AWS terbaik.

Pengaturan konektor

Agen landing zone memerlukan Akun AWS konektor target untuk menyediakan sumber daya di akun manajemen organisasi Anda. Konektor memiliki izin untuk:

- Siapkan AWS Control Tower
- Buat unit dan akun organisasi
- Konfigurasikan Kebijakan Kontrol Layanan (SCP)

Saat menyetujui permintaan konektor, Anda memberikan izin AWS Transform ke:

- Menyediakan dan mengelola infrastruktur landing zone di target Akun AWS dan Wilayah.
  Ini termasuk izin untuk item berikut, dibatasi untuk sumber daya yang ditandai dengan `CreatedBy:AWSTransform` dan `ATWorkspace:{workspace-id}` jika berlaku:
  - Operasi bucket S3 (buat, baca, tulis, hapus) untuk bucket yang dimulai dengan `transform-vmware-landing-zone-`
  - CloudFormation penyebaran tumpukan dan mengubah manajemen set untuk tumpukan landing zone
  - AWS Operasi Control Tower (mengelola zona pendaratan, mengaktifkan garis dasar dan kontrol)

- AWS Organizations management (membuat dan mengelola unit organisasi, membuat akun, dan memindahkan akun)
- Manajemen kebijakan kontrol layanan (SCP) melalui AWS Control Tower
- AWS Manajemen artefak penyediaan Service Catalog

Saat Anda membuat konektor, Anda menentukan target Wilayah AWS. Wilayah ini harus sama dengan Wilayah Control Tower rumah Anda. Untuk informasi selengkapnya tentang Wilayah Control Tower, lihat Cara Wilayah AWS bekerja dengan AWS Control Tower.

Pada awal penyiapan landing zone, AWS Transform mengambil konfigurasi konektor Anda dan menampilkan ID akun manajemen AWS Organisasi dan Wilayah target untuk konfirmasi. Untuk informasi selengkapnya, lihat Konektor.

> **Important**
>
> Ketergantungan Wilayah Pusat Identitas IAM — AWS Transform memerlukan AWS IAM Identity Center (IAM Identity Center), yang berarti Wilayah konektor Anda harus cocok dengan Wilayah rumah AWS Control Tower dan Wilayah Pusat Identitas IAM Anda. Jika IAM Identity Center sudah dikonfigurasi di organisasi Anda, inisialisasi AWS Control Tower akan gagal jika konektor menargetkan Wilayah yang berbeda. Untuk informasi selengkapnya, lihat Pertimbangan untuk pelanggan IAM Identity Center di Panduan Pengguna AWS Control Tower.

Pengaturan pondasi

Fase pengaturan pondasi menetapkan infrastruktur landing zone inti menggunakan AWS Control Tower. Ketika AWS Control Tower menyiapkan landing zone, Control Tower secara otomatis menyediakan satu set sumber daya terkelola di akun manajemen Anda yang membentuk fondasi tata kelola untuk seluruh AWS Organisasi Anda:

- Root — Induk tingkat atas yang berisi semua OU di landing zone Anda.
- Keamanan OU - Dibuat secara otomatis oleh Control Tower. Berisi dua akun bersama: akun Arsip Log (pencatatan terpusat dan tidak dapat diubah untuk semua aktivitas AWS API dan perubahan sumber daya di seluruh organisasi Anda) dan akun Audit (akses hanya-baca ke semua akun untuk peninjauan keamanan dan kepatuhan). Akun ini tidak dapat diubah namanya atau diganti setelah penyiapan awal.

- Kontrol wajib (pagar pembatas) — Control Tower secara otomatis menerapkan kontrol preventif dan detektif di seluruh organisasi Anda untuk menegakkan kebijakan tata kelola dasar. Ini tidak dapat dinonaktifkan.
- Direktori IAM Identity Center — Control Tower membuat direktori cloud-native dengan grup yang telah dikonfigurasi sebelumnya dan akses masuk tunggal untuk pengguna landing zone Anda. Untuk informasi selengkapnya, lihat Pusat AWS Identitas IAM.

Control Tower digunakan CloudFormation StackSetsuntuk menyebarkan dan mengelola sumber daya ini secara konsisten di semua akun dan Wilayah di organisasi Anda. Anda tidak boleh memodifikasi atau menghapus sumber daya yang dikelola Control Tower di luar metode yang didukung, karena hal itu dapat menyebabkan landing zone Anda memasuki status yang tidak dikenal.

Konvensi email akun

AWS membutuhkan alamat email unik untuk setiap akun. Email ini menerima pemberitahuan penting untuk akun tersebut. AWS Transform menggunakan plus pengalamatan untuk menghasilkan email akun unik dari satu kotak pesan.

Format: `prefix+account-name@domain`

Anda memberikan awalan (misalnya,`aws-admin`) dan domain (misalnya,`acme.com`), dan AWS Transform memperoleh semua email akun secara otomatis. Contoh:

- Akun audit: `aws-admin+audit@acme.com`
- Akun Arsip Log: `aws-admin+log-archive@acme.com`
- Akun Sandbox: `aws-admin+sandbox@acme.com`

Dalam skenario brownfield, AWS Transform memeriksa email akun yang ada untuk menyimpulkan konvensi plus-addressing yang sudah digunakan dan menawarkan untuk melanjutkan dengan pola yang sama.

Struktur pondasi yang direkomendasikan

Berdasarkan praktik AWS terbaik, AWS Transform merekomendasikan struktur OU dasar berikut. Anda dapat menyesuaikannya sebelum pembuatan.

OU	Tujuan	Akun
Keamanan	Pencatatan dan pemantauan audit terpusat. Mengisolasi layanan ini di akun khusus dirancang untuk membantu menjaga jejak audit Anda terpisah dari tim beban kerja.	Audit, Arsip Log
Infrastruktur	Jaringan bersama (Transit Gateway, VPN), DNS, dan layanan umum. Memusatkan ini disarankan untuk membantu mengurangi duplikasi dan memberi tim jaringan Anda satu tempat untuk mengelola konektivitas.	Tidak ada (dibuat kosong)
Kotak pasir	Eksperimen pengembang dengan batas pengeluaran dan akses terbatas. Disaranka n untuk memberi pengembang ruang untuk bereksperimen tanpa mempertaruhkan sumber daya produksi.	Kotak pasir
Beban kerja	Berisi Produksi, Non-Production, dan Sub-OU yang Diatur secara opsional. Akun beban kerja dirancang pada fase berikutnya berdasarkan persyaratan migrasi Anda.	Tidak ada (dibuat kosong)

> **Note**
>
> Security OU dengan akun Audit dan Log Archive dibuat sebagai bagian dari pengaturan dasar Control Tower. Infrastruktur, Kotak Pasir, dan Beban Kerja OU dibuat secara terpisah setelah Anda mengonfirmasi strukturnya.

Dalam skenario brownfield, AWS Transform membandingkan fondasi Anda yang ada dengan struktur yang direkomendasikan ini dan hanya melaporkan kesenjangan. Misalnya: “Yayasan Anda memiliki Keamanan dan Infrastruktur OU tetapi tidak ada Sandbox OU.”

Kebijakan Kontrol Layanan (SCP)

SCP adalah pagar pembatas izin tingkat organisasi yang menetapkan izin maksimum untuk semua akun di Organisasi Anda. AWS Mereka tidak memberikan akses — sebaliknya, mereka menentukan batasan yang tidak dapat dilampaui oleh siapa pun di akun, bahkan administrator akun.

Sebagai bagian dari penyebaran Control Tower, pagar pembatas dasar diterapkan secara otomatis. AWS Transform juga merekomendasikan SCP tambahan yang dirancang untuk membantu memperkuat postur organisasi Anda. Ini didasarkan pada praktik AWS terbaik untuk landing zone minimum yang layak.

SCP dapat diterapkan ke Infrastruktur, Kotak Pasir, dan Beban Kerja OU. Security OU dikelola oleh Control Tower dan tidak dapat ditargetkan oleh SCP melalui alat ini.

> **Important**
>
> Security OU adalah yayasan OU yang dikelola oleh Control Tower. Anda tidak dapat menambahkan akun, SCP, atau sumber daya apa pun ke dalamnya melalui agen landing zone.

Dalam skenario brownfield, AWS Transform memeriksa SCP mana yang sudah diterapkan dan hanya merekomendasikan SCP yang akan mengisi celah.

Penyebaran yayasan

Setelah desain pondasi selesai, Anda memilih cara menyebarkan:

- Terapkan untuk saya — AWS Transform menyebarkan dasar OU, akun, dan SCP ke Organisasi Anda. AWS
- Saya akan menerapkannya sendiri — AWS Transform menghasilkan artefak Infrastructure as Code (IAc) untuk diunduh dalam format pilihan Anda (lihat). the section called “Format IAc”
- Rancang akun beban kerja terlebih dahulu — Lewati penerapan dan lanjutkan ke fase desain akun beban kerja. Anda dapat menyebarkan semuanya bersama-sama nanti.

Inisialisasi Control Tower

Jika AWS Transform mendeteksi bahwa AWS Control Tower belum diinisialisasi di organisasi Anda, Transform akan memberi pengguna tautan ke halaman konsol AWS Transform. Menghasilkan

operasi di tautan akan membuat CloudFormation tumpukan ke bootstrap Control Tower. Prosesnya akan membuat tumpukan ini di CloudFormation konsol untuk Wilayah target Anda. Setelah pembuatan tumpukan selesai, AWS Transform melanjutkan penerapan.

Desain akun beban kerja

Pada fase desain akun beban kerja, AWS Transform mendesain struktur OU dan akun untuk beban kerja aplikasi Anda berdasarkan inventaris migrasi, persyaratan bisnis, dan preferensi pemisahan lingkungan.

Konteks perencanaan migrasi

AWS Transform mengambil data dari fase perencanaan migrasi Anda, termasuk rencana gelombang, pemetaan server-ke-aplikasi, dan konteks bersama. Jika data perencanaan migrasi tersedia, AWS Transform menampilkan ringkasan dan meminta Anda untuk mengonfirmasi atau menyesuaikannya. Jika tidak ada data perencanaan migrasi yang tersedia, AWS Transform akan mengajukan pertanyaan penemuan secara langsung.

Penemuan

AWS Transform mengajukan pertanyaan untuk memahami persyaratan beban kerja Anda. Anda dapat melewatkan pertanyaan apa pun. Topiknya mencakup:

- Jumlah unit bisnis atau tim yang menggunakan AWS
- Industri dan kerangka kerja apa pun yang berlaku (HIPAA,, SOC2 PCI-DSS, FedRAMP)
- Apakah beban kerja menangani data sensitif (PII, PHI, keuangan)
- Preferensi pemisahan lingkungan (dev/test/staging/prod sebagai akun terpisah atau dibagikan)
- Persyaratan isolasi beban kerja
- Aplikasi bisnis dan tujuannya
- Pengelompokan server ke dalam aplikasi
- Pelacakan biaya dan kebutuhan alokasi (berdasarkan unit bisnis, proyek, lingkungan)
- Pertumbuhan yang diharapkan dalam 12-24 bulan ke depan
- Preferensi strategi akun (aplikasi tunggal per akun, dikelompokkan, atau berbasis lingkungan)

Struktur beban kerja yang diusulkan

Berdasarkan jawaban dan data perencanaan migrasi Anda, AWS Transform mengusulkan OU dan struktur akun di bawah Workloads OU. Proposal tersebut mencakup alasan di balik setiap keputusan desain.

AWS Transform mengikuti prinsip-prinsip desain ini:

- Semua server dalam gelombang migrasi masuk ke akun yang sama — gelombang tidak dapat dibagi di seluruh akun. Ini adalah batasan rehost selama eksekusi gelombang.
- Jika Anda meminta lingkungan yang terisolasi, AWS Transform create Workloads/Production dan Workloads/Non-Production sub-OUS.
- Jika kerangka kerja yang berlaku diidentifikasi, AWS Transform create Workloads/Regulated dan Workloads/Standard sub-OUS.
- Jika beberapa unit bisnis memerlukan tata kelola yang berbeda, AWS Transform membuat OU khusus unit bisnis di bawah Beban Kerja.
- Aplikasi data kritis atau sensitif mendapatkan satu aplikasi per akun. Dalam hal ini, Anda mungkin diminta untuk mengulangi rencana gelombang Anda.
- Aplikasi yang digabungkan erat dengan dependensi bersama dikelompokkan dalam satu akun.

Setiap akun yang diusulkan meliputi: nama, tujuan, target OU, dan unit bisnis. AWS Transform menunjukkan konvensi penamaan yang digunakan (misalnya,`<business-unit>-<environment>-<workload>`).

Anda dapat meninjau dan memodifikasi struktur yang diusulkan sebelum AWS Transform menerapkan perubahan. Setelah menerapkan, Anda dapat mengulangi — membuat perubahan tambahan sampai Anda puas.

Konfigurasi SCP beban kerja

Setelah struktur beban kerja dibuat, AWS Transform menyajikan SCP yang tersedia dan menanyakan apakah Anda ingin menerapkannya ke OU beban kerja Anda. Anda memilih SCP mana yang akan diterapkan dan ke OU mana. AWS Transform menerapkan SCP dan menampilkan pohon organisasi yang diperbarui dengan tabel ringkasan SCP.

Penyebaran beban kerja

Setelah desain beban kerja selesai, Anda memilih cara menerapkan:

- Terapkan untuk saya — AWS Transform menyebarkan beban kerja OU, akun, dan SCP ke Organisasi Anda. AWS
- Saya akan menerapkannya sendiri — AWS Transform menghasilkan artefak IAC untuk diunduh dalam format pilihan Anda (lihat). the section called “Format IAc”

Format IAc

Saat Anda memilih penerapan mandiri, AWS Transform menghasilkan Infrastruktur sebagai artefak Kode dalam format berikut:

- AWS Cloud Development Kit (AWS CDK)— TypeScript proyek untuk penyebaran infrastruktur terprogram.
- HashiCorp Terraform - Menghasilkan template HashiCorp Configuration Language (HCL) untuk mengelola sumber daya landing zone.
- Landing Zone Accelerator (LZA) - Konfigurasi file YAMAL berdasarkan LZA Universal Configuration versi 1.1.0. Template siap perusahaan ini bekerja dengan Landing Zone Accelerator AWS untuk membangun lingkungan multi-akun. AWS File yang dihasilkan mencakup pengaturan pra-konfigurasi untuk tata kelola, struktur organisasi, dan jaringan yang selaras dengan AWS praktik terbaik. Untuk mempelajari lebih lanjut, lihat Konfigurasi Universal LZA.

> **Note**
>
> Saat menerapkan melalui pipeline Landing Zone Accelerator (LZA), akun AWS Transform dan instalasi LZA Anda harus berada di Organisasi yang sama. AWS Deployment akan gagal jika ada ketidakcocokan antara Organizations ID yang digunakan dalam AWS Transform dan LZA. Untuk mempelajari cara mengatur instalasi LZA menggunakan Organizations, lihat Instalasi berbasis AWS Organizations.

Setelah Anda memilih format, AWS Transform menghasilkan artefak dan membuatnya tersedia untuk diunduh.

Untuk memverifikasi file yang diunduh tidak rusak atau dirusak, buat dan unduh checksum, lalu bandingkan dengan hash yang dibuat secara lokal menggunakan:

```
openssl dgst -sha256 -binary <file.zip> | base64
```

Proses persetujuan penerapan

Permintaan penyebaran zona pendaratan memerlukan persetujuan eksplisit sebelum eksekusi. Saat Anda mengirimkan permintaan penerapan, permintaan tersebut secara otomatis merutekan ke pemberi persetujuan resmi melalui tab AWS Transform Approvals.

Approvers meninjau CloudFormation template dan konfigurasi landing zone. Hanya pengguna dengan peran Admin di AWS Transform yang dapat menyetujui permintaan penerapan. Setiap pengiriman memicu siklus peninjauan baru, dan penerapan hanya dilanjutkan setelah menerima konfirmasi.

Jika pemberi persetujuan menolak permintaan Anda, hubungi mereka secara langsung untuk membahas modifikasi yang diperlukan. Sistem melacak semua keputusan persetujuan untuk tujuan audit dan mempertahankan riwayat penyebaran.

Tandai sumber daya landing zone

AWS Transform secara otomatis menandai semua sumber daya yang dihasilkan `"CreatedBy": "AWSTransform"` bersama dengan ID definisi dan eksekusi untuk tujuan pelacakan.

Tag otomatis

Semua sumber daya landing zone menerima tag berikut:

- `CreatedBy`— AWSTransForm
- `ATWorkspace`— Pengidentifikasi ruang kerja

> **Note**
>
> Jika migrasi Anda merupakan bagian dari AWS Migration Acceleration Program (MAP 2.0), Anda dapat menyertakan tag MAP yang diperlukan: Key: `map-migrated` Value: `migMPE_ID` (di mana MPE_ID adalah pengenal Evaluasi Portofolio Migrasi Anda). Tag MAP diminta selama fase pengaturan konektor. AWS Transform menerapkan tag ini selama penyebaran landing zone.

Membalikkan perubahan

Hanya elemen yang tidak digunakan yang dapat dihapus. Setelah OU atau akun dikerahkan, itu tidak dapat dihapus melalui agen landing zone.

Saat menghapus elemen, urutan penting - Anda harus menghapus anak-anak sebelum orang tua:

1. Hapus akun terlebih dahulu (melalui email).
2. Hapus SCP dari OU.
3. Hapus OU anak - OU tidak dapat dihapus jika masih memiliki akun atau OU bersarang.

Sumber daya terkait

- the section called “Connect target Akun AWS s dan wilayah”
- the section called “Migrasikan jaringan Anda ke AWS”
- AWS Panduan Pengguna Control Tower
- AWS Panduan Pengguna Organizations
- AWS Panduan Pengguna Pusat Identitas IAM
- CloudFormation Panduan Pengguna
- Akselerator Zona Pendaratan aktif AWS
- AWS Well-Architected Kerangka
- AWS Panduan Preskriptif: Membangun Zona Pendaratan

Migrasikan jaringan Anda ke AWS

Dengan AWS Transform, Anda dapat memigrasikan jaringan Anda ke AWS. AWS Transform menerjemahkan konfigurasi lingkungan sumber Anda menjadi sumber daya jaringan yang AWS setara - VPC, subnet, grup keamanan, gateway NAT, gateway transit, IP elastis, rute, dan tabel rute sesuai kebutuhan. Anda dapat meninjau dan memodifikasi konfigurasi jaringan yang dihasilkan sebelum penerapan. Anda dapat menerapkan konfigurasi dengan AWS Transform dan menganalisis konektivitas jaringan. Atau, Anda dapat memilih self-deployment dan menerima Infrastructure as Code (IAc) dalam format pilihan Anda: AWS Cloud Development Kit (AWS CDK), Landing Zone Accelerator (LZA), atau Terraform. HashiCorp

Untuk memigrasikan jaringan Anda, ikuti langkah-langkah berikut:

1. Unggah file jaringan sumber Anda.
2. Unggah file konfigurasi tambahan (opsional, untuk lingkungan RVTools).
3. Pilih topologi jaringan.
4. Pilih strategi pemetaan grup keamanan.
5. Tinjau dan optimalkan jaringan Anda.
6. Hasilkan diagram jaringan (opsional).
7. Konfigurasikan penandaan sumber daya.
8. Menyebarkan jaringan Anda.

> **Note**
>
> Untuk penerapan multi-akun, Anda harus mengonfigurasi peran IAM lintas akun dan akses tepercaya untuk AWS Organizations sebelum memulai migrasi jaringan. Untuk informasi selengkapnya tentang jenis migrasi, lihatthe section called “Langkah 1: Pemilihan tipe migrasi”.

Langkah 1: Pemetaan jaringan sumber

Proses pemetaan jaringan mengharuskan Anda mengunggah file konfigurasi dari lingkungan sumber Anda. Alat yang Anda pilih tergantung pada jenis jaringan sumber Anda:

- Software Defined Networks (SDN): Import/Export untuk virtualisasi jaringan VMware NSX atau konfigurasi Cisco ACI untuk Cisco Application Centric Infrastructure.
- Jaringan VMware vSphere: RVTools. Saat Anda menggunakan file RVTools, AWS Transform hanya menghasilkan konfigurasi VPC Amazon. Konfigurasi grup keamanan memerlukan input tambahan dari firewall atau file jaringan yang ditentukan perangkat lunak. Untuk informasi selengkapnya tentang pembuatan grup keamanan dari file tambahan, lihat File konfigurasi tambahan.
- Jaringan berdasarkan data konfigurasi firewall: Ekspor file dari Palo Alto Networks Firewall, Fortinet FortiGate Firewall, atau Cisco ACI. Untuk informasi selengkapnya tentang versi dan instruksi ekstraksi yang didukung, lihat Ekstraksi file konfigurasi.
- Jaringan hybrid yang menjalankan beban kerja VMware dan non-VMware:AWS Transform discovery tool atau ModelizeIt.

- Jenis file lainnya: Jika file konfigurasi Anda bukan salah satu format yang didukung yang tercantum di atas, file dikonversi secara otomatis ke format yang didukung. Konversi ini dapat memakan waktu hingga dua jam berdasarkan ukuran dan kompleksitas file.

> **Note**
>
> Ukuran file jaringan sumber maksimum yang didukung adalah 70 MB.

> **Warning**
>
> Unduh RVTools hanya dari situs resmi Dell di. https://www.dell.com/en-us/shop/vmware/sl/rvtools Jangan mengunduh RVTools dari sumber tidak resmi.

Setiap segmen jaringan sumber dipetakan ke VPC yang berbeda. Segmentasi jaringan bervariasi menurut jenis sumber:

- vNetwork: AWS Transform grup VM oleh VSwitch dan virtual LAN (VLAN). VLAN dapat muncul di bawah beberapa VSwitches (kecuali VLAN 0).
- Jaringan NSX: AWS Mengubah segmen jaringan berdasarkan Tier-1 router, mengelompokkan router dan mengumpulkan segmennya.

Langkah 2: File konfigurasi tambahan

Untuk lingkungan sumber RVTools, Anda dapat mengunggah file konfigurasi tambahan secara opsional untuk mengaktifkan pembuatan grup keamanan. Jika Anda tidak mengunggah file konfigurasi tambahan, tidak ada grup keamanan yang dibuat untuk RVTools-based migrasi Anda.

AWS Transform mendukung jenis file konfigurasi tambahan berikut. Anda hanya dapat mengunggah satu file konfigurasi dari satu platform.

- Cisco Application Centric Infrastructure (ACI) menyediakan konfigurasi kebijakan jaringan.
- Palo Alto Networks menyediakan kebijakan keamanan firewall.
- Fortinet FortiGate menyediakan kebijakan keamanan firewall.

Saat Anda mengunggah firewall atau file Cisco ACI, AWS Transform menghasilkan infrastruktur jaringan dan grup keamanan. Saat Anda mengunggah file RVTools sendirian, AWS Transform hanya menghasilkan infrastruktur jaringan.

Untuk informasi selengkapnya tentang versi dan instruksi ekstraksi yang didukung, lihat Ekstraksi file konfigurasi.

Langkah 3: Topologi jaringan

Selama langkah definisi jaringan, Anda memilih topologi jaringan. Anda dapat memilih topologi VPC Terisolasi atau topologi Hub dan Spoke.

VPC Terisolasi

Apa yang dikerahkan

VPC terisolasi adalah lingkungan jaringan independen yang beroperasi sebagai unit terpisah di dalamnya AWS. VPC Anda benar-benar terisolasi, tanpa jalur komunikasi bawaan di antara mereka. Pemisahan ini memberikan tingkat perlindungan batas jaringan tertinggi.

AWS Transform menciptakan sumber daya berikut:

- VPC khusus untuk setiap segmen jaringan sumber yang terdeteksi.
- Subnet pribadi berdasarkan konfigurasi jaringan sumber Anda.
- Grup keamanan (jika Anda menyediakan firewall atau file konfigurasi SDN).

Selesaikan pengaturan Anda

AWS Transform menyebarkan infrastruktur jaringan tetapi meninggalkan akses internet dan konektivitas antar-VPC kepada Anda, sehingga Anda dapat memilih konfigurasi yang memenuhi persyaratan organisasi Anda.

Untuk mengaktifkan akses internet untuk VPC Terisolasi, selesaikan langkah-langkah berikut:

1. Buat gateway internet dan lampirkan ke VPC.
2. Buat subnet publik di setiap Availability Zone di mana Anda memerlukan akses internet. Tambahkan rute untuk `0.0.0.0/0` menunjuk ke gateway internet. Untuk informasi selengkapnya tentang konfigurasi subnet, lihat Subnet untuk VPC Anda.

3. Buat gateway NAT di subnet publik (satu per AZ untuk ketersediaan tinggi). Alokasikan IP Elastis untuk setiap gateway NAT.
4. Perbarui tabel rute subnet pribadi Anda — tambahkan rute untuk `0.0.0.0/0` menunjuk ke gateway NAT di AZ yang sama.
5. Tinjau aturan grup keamanan Anda — pastikan aturan keluar memungkinkan lalu lintas yang dibutuhkan beban kerja Anda (HTTPS, DNS, dll.).

Untuk VPC-to-VPC komunikasi, siapkan VPC peering atau Transit Gateway dan perbarui tabel rute di setiap VPC untuk merutekan lalu lintas ke koneksi peering atau lampiran TGW.

Hub dan Spoke

Dalam model ini, AWS Transit Gateway bertindak sebagai hub pusat yang menghubungkan beberapa VPC beban kerja (jari-jari).

Apa yang dikerahkan

AWS Transform menciptakan sumber daya berikut:

- Spoke VPC: Satu VPC per segmen jaringan sumber yang terdeteksi, dengan subnet pribadi dan lampiran Transit Gateway.
- Inspeksi VPC: Menghosting alat firewall Anda untuk inspeksi lalu lintas. Semua lalu lintas lintas VPC dirutekan melalui VPC ini. Lampiran Transit Gateway menggunakan mode alat, pengaturan yang memastikan arus lalu lintas secara simetris melalui alat yang sama untuk kedua arah koneksi.
- VPC Masuk: Menangani lalu lintas yang memasuki jaringan Anda dari internet publik (inbound utara-selatan). Termasuk gateway internet dan subnet publik di beberapa Availability Zone.
- VPC Keluar: Menangani lalu lintas yang meninggalkan jaringan Anda ke internet publik (outbound utara-selatan). Termasuk gateway internet, gateway NAT dengan alamat IP elastis di setiap Availability Zone untuk ketersediaan tinggi, dan subnet pribadi untuk lampiran Transit Gateway.
- Tabel rute Transit Gateway: Dua tabel rute mengarahkan lalu lintas melalui VPC Inspeksi. Tabel yang Tidak Diinspeksi dikaitkan dengan VPC spoke, VPC Masuk, dan VPC Keluar — ini merutekan semua lalu lintas (0.0.0. 0/0) ke lampiran VPC Inspeksi dan merupakan tabel rute asosiasi default. Tabel yang Diinspeksi dikaitkan dengan VPC Inspeksi — tabel ini berisi rute yang disebarkan dari semua VPC spoke dan merupakan tabel rute propagasi default.

Untuk penerapan multi-akun, Transit Gateway dibagikan di seluruh akun melalui AWS Resource Access Manager (RAM).

Arus lalu lintas

Semua lalu lintas lintas VPC mengikuti jalur ini:

1. Lalu lintas dari VPC spoke dikirim ke Transit Gateway (rute default 0.0.0. 0/0).
2. Tabel rute yang tidak diperiksa merutekan lalu lintas ke VPC Inspeksi.
3. Firewall Anda di VPC Inspeksi memeriksa lalu lintas dan meneruskannya kembali ke Transit Gateway.
4. Tabel rute yang Diinspeksi merutekan lalu lintas ke tujuan menggunakan VPC menggunakan rute yang disebarkan.

Untuk lalu lintas internet keluar, tabel rute yang Diinspeksi merutekan lalu lintas ke VPC Keluar. Gateway NAT menerjemahkan alamat IP pribadi sebelum lalu lintas diteruskan ke gateway internet. Tabel rute publik VPC Outbound mencakup rute spesifik untuk setiap rentang VPC Inter-Domain Classless Routing (CIDR) spoke kembali ke Transit Gateway. Rute ini memungkinkan lalu lintas kembali untuk mencapai VPC spoke yang benar.

Lalu lintas internet masuk masuk melalui gateway internet VPC Inbound dan mengikuti jalur inspeksi yang sama untuk mencapai VPC spoke.

Selesaikan pengaturan Anda

AWS Transform menyebarkan infrastruktur jaringan tetapi meninggalkan konfigurasi firewall dan penyiapan layanan masuk kepada Anda, sehingga Anda dapat memilih peralatan dan kebijakan keamanan yang memenuhi persyaratan organisasi Anda.

> **Note**
>
> Secara default, lalu lintas lintas VPC melewati VPC Inspeksi tanpa inspeksi. Anda harus menggunakan firewall untuk mengaktifkan inspeksi lalu lintas.

Menyebarkan firewall: Buat subnet tambahan di VPC Inspeksi untuk titik akhir firewall. AWS Transform membuat subnet hanya untuk lampiran Transit Gateway. Rutekan lalu lintas dari subnet lampiran TGW ke titik akhir firewall, dan dari subnet firewall kembali ke Transit Gateway. Anda dapat menggunakan AWS Network Firewall atau alat pihak ketiga. Untuk informasi selengkapnya tentang penerapan firewall dengan Transit Gateway, lihat Membuat firewall dengan Transit Gateway.

Verifikasi konektivitas: Setelah Anda menerapkan firewall, uji akses internet keluar dari instance VPC spoke (misalnya,). `curl https://aws.amazon.com` Anda dapat menggunakan Reachability Analyzer untuk memecahkan masalah konektivitas.

Menyiapkan layanan masuk: Untuk meng-host layanan yang menghadap publik, gunakan Application Load Balancer atau Network Load Balancer di subnet publik VPC Inbound. Konfigurasikan grup target yang mengarah ke instance di VPC spoke Anda melalui Transit Gateway, dan verifikasi bahwa tabel rute yang Diinspeksi memiliki rute kembali ke VPC Masuk.

Jika Anda ingin kontrol halus atas komunikasi antara VPC, pilih opsi Isolated VPC dan modifikasi jaringan yang dihasilkan untuk membuat jalur komunikasi spesifik yang Anda butuhkan.

Langkah 4: Pemetaan kelompok keamanan

Pilih cara kebijakan keamanan sumber Anda diterjemahkan ke grup AWS keamanan. AWS Transform membuat grup keamanan berdasarkan konfigurasi lingkungan sumber Anda. Kebijakan keamanan, aturan kebijakan keamanan, kebijakan gateway, dan aturan kebijakan gateway dikonversi ke grup keamanan.

> **Important**
>
> AWS Transform membuat grup keamanan dengan upaya terbaik agar sesuai dengan lingkungan sumber Anda. Tinjau dan modifikasi grup keamanan yang dihasilkan untuk memastikan bahwa mereka memenuhi kebutuhan dan kebijakan keamanan perusahaan Anda.

Referensi kelompok keamanan

Saat grup keamanan dibuat, AWS Transform menggunakan referensi grup keamanan jika didukung. Referensi grup keamanan menetapkan aturan keamanan berdasarkan ID grup keamanan lain daripada rentang alamat IP tertentu (blok CIDR). Pendekatan ini memberikan konfigurasi keamanan yang lebih fleksibel dan dapat dipelihara.

Aturan grup keamanan Anda hanya dapat mereferensikan grup keamanan lain dalam VPC yang sama, atau dalam VPC yang terhubung dalam Wilayah yang sama. Cross-account Referensi juga didukung. Anda tidak dapat mereferensikan grup keamanan di VPC yang tidak terhubung atau di seluruh Wilayah. Untuk VPC yang terhubung, hanya aturan masuk yang mendukung referensi

grup keamanan lintas-VPC — aturan keluar harus menggunakan aturan. CIDR-based Cara AWS Transform membuat aturan grup keamanan bergantung pada topologi jaringan yang Anda pilih:

- Hub dan Spoke: Transit Gateway menyediakan konektivitas jaringan antara VPC. AWS Transform menggunakan referensi untuk aturan dalam-VPC dan lintas masuk. VPC/cross-account Cross-VPC/cross-account aturan keluar (keluar) menggunakan aturan. CIDR-based
- VPC Terisolasi: VPC Anda tidak memiliki konektivitas jaringan di antara mereka. AWS Transform hanya menggunakan referensi untuk aturan dalam-VPC. Semua aturan lintas VPC dan lintas akun menggunakan aturan. CIDR-based

CIDR-based aturan juga digunakan ketika konfigurasi sumber tidak simetris.

Pilih salah satu strategi pemetaan grup keamanan berikut:

- MAP: Menerjemahkan aturan keamanan dari lingkungan sumber Anda ke grup dan aturan AWS keamanan. Gunakan opsi ini untuk migrasi menggunakan pengalamatan IP statis.
- MAP_DHCP (Terjemahkan dengan dukungan DHCP): Menerjemahkan aturan keamanan dari lingkungan sumber Anda dengan kompatibilitas DHCP. DHCP memberikan alamat IP secara dinamis dari rentang CIDR subnet. Akibatnya, aturan keluar lintas-VPC diperluas agar sesuai dengan subnet tujuan penuh CIDR. CIDR yang lebih sempit akan memblokir DHCP-assigned IP yang berada di luar kisaran itu - tinjau aturan ini pasca-migrasi.

  Gunakan opsi ini untuk dukungan DHCP dengan komunikasi Cross-VPC Transit Gateway. Juga bekerja dengan IP statis, tetapi mungkin menghasilkan aturan yang lebih luas daripada MAP.
- SKIP: Tidak menerjemahkan aturan keamanan. Konfigurasikan grup AWS keamanan secara manual pasca-migrasi. Bekerja dengan lingkungan IP statis dan DHCP. Untuk lingkungan sumber RVTools tanpa file konfigurasi tambahan, AWS Transform secara otomatis menggunakan SKIP.

> **Note**
>
> Strategi pemetaan Anda menentukan opsi penetapan IP Anda. MAP hanya mendukung IP statis. MAP_DHCP dan SKIP mendukung statis dan DHCP.

Pendekatan migrasi IP

Anda memiliki dua pilihan konfigurasi jaringan untuk migrasi Anda:

Pemilihan rentang jaringan

- Pertahankan Rentang yang Ada (Retensi Rentang Alamat IP): Pertahankan rentang alamat IP asli selama migrasi. Ideal untuk migrasi tempat Anda memindahkan aplikasi AWS tanpa modifikasi (lift-and-shift), terutama dengan aplikasi lama yang memiliki dependensi IP hard-code atau aturan firewall yang ada.
- Memperbarui ke rentang IP baru (pembaruan CIDR): Anda dapat memodifikasi setiap rentang CIDR VPC selama migrasi, AWS dan Transform secara otomatis menyebarkan perubahan ke subnet, tabel rute, dan grup keamanan.

Penetapan alamat IP

- Alamat IP Tetap (Statis): AWS Transform menetapkan IP statis berdasarkan CIDR. Ini terbaik untuk aplikasi yang memerlukan perilaku jaringan yang dapat diprediksi, manajemen DNS, atau kontrol IP-based akses. IP tetap ada di seluruh instans restart melalui Elastic Network Interfaces (ENI), yang merupakan kartu jaringan virtual yang dilampirkan ke instans Anda.
- Penugasan IP Dinamis (AWS DHCP): Secara otomatis menetapkan IP dari subnet pool saat peluncuran instance. Optimal untuk aplikasi yang dirancang untuk berjalan di cloud dan beban kerja auto-scaling. Mengurangi overhead operasional tetapi membutuhkan aplikasi untuk menggunakan DNS atau penemuan layanan.

Anda dapat menggabungkan salah satu pilihan rentang dengan salah satu metode penetapan IP.

> **Note**
>
> Strategi penetapan alamat IP diatur pada tingkat gelombang. Anda dapat menetapkan strategi yang berbeda untuk server tertentu dengan menyesuaikan file gelombang. Misalnya, jika Anda memilih pendekatan alamat IP statis untuk gelombang tetapi ingin menetapkan pendekatan dinamis ke server tertentu, Anda akan menggunakan `[RESET_VALUE]` seperti yang dijelaskan dalam Mengedit konfigurasi Anda di panduan pengguna MGN.

Langkah 5: Tinjau dan optimalkan jaringan Anda

Setelah AWS Transform menghasilkan konfigurasi jaringan target, Anda dapat meninjau segmen jaringan lokal yang dikonvergenkan ke AWS infrastruktur. Gunakan antarmuka visual untuk meninjau jaringan Anda, dan antarmuka obrolan untuk membuat perubahan dan menerima rekomendasi

terpandu. AWS Transform melakukan analisis dampak cascading dan mengimplementasikan perubahan yang diperlukan untuk menjaga konsistensi jaringan dan kepatuhan terhadap praktik terbaik. Anda juga dapat meminta AWS Transform untuk menganalisis jaringan Anda dan menyarankan pengoptimalan — lihat Rekomendasi terpandu.

VPC yang ada di akun target Anda

Jika akun target Anda sudah berisi VPC — dari fase migrasi sebelumnya atau proyek infrastruktur paralel — AWS Transform secara otomatis mendeteksi dan menampilkannya bersama VPC yang dipetakan selama proses peninjauan. Untuk migrasi multi-akun, AWS Transform mendeteksi VPC yang ada di semua akun di Organisasi Anda. AWS

Visibilitas ini membantu Anda memahami bagaimana jaringan yang direncanakan terkait dengan infrastruktur yang ada, mengidentifikasi potensi konflik CIDR, dan membuat keputusan berdasarkan informasi sebelum penerapan.

> **Note**
>
> AWS Transform mendeteksi VPC yang ada saja (bukan subnet atau sumber daya lainnya). Deteksi hanya baca - AWS Transform tidak memodifikasi VPC Anda yang ada.

Optimalkan jaringan Anda

> **Note**
>
> Operasi ini hanya berlaku untuk VPC beban kerja. Untuk VPC alat di topologi Hub dan Spoke (Inspeksi, Masuk, Keluar), hanya Ubah alamat IP yang didukung.
> Anda tidak dapat membatalkan operasi Delete, Merge, dan Split. Tinjau konfigurasi Anda dengan cermat sebelum Anda menerapkan perubahan ini.

Operasi berikut tersedia untuk VPC:

- Hapus: Hapus VPC secara permanen dari konfigurasi. Gunakan ini untuk segmen jaringan usang yang seharusnya tidak bermigrasi. AWS
- Kecualikan: Hapus sementara VPC dari migrasi untuk strategi migrasi bertahap. VPC yang dikecualikan tidak digunakan tetapi dapat dimasukkan kembali nanti.

- Sertakan: Tambahkan VPC yang sebelumnya dikecualikan kembali ke migrasi.
- Gabung: Gabungkan dua VPC menjadi satu. VPC pertama menyimpan identitasnya dan menyerap semua subnet dari VPC kedua. Grup keamanan dipindahkan ke VPC gabungan dan asosiasi mereka dibangun kembali sesuai dengan itu. CIDR VPC pertama diperluas ke rentang CIDR terkecil yang berisi CIDR asli, dan perutean diperbarui secara otomatis. VPC kedua dihapus dari konfigurasi.

  Persyaratan gabungan:

  - CIDR subnet tidak boleh tumpang tindih antara dua VPC.
  - CIDR yang digabungkan tidak boleh melebihi /16.
  - Untuk penerapan multi-akun, kedua VPC harus ditetapkan ke akun yang sama.
- Ubah alamat IP: Ubah alamat IP dasar CIDR VPC sambil menjaga panjang awalan yang sama. AWS Transform secara otomatis menerjemahkan semua CIDR subnet dengan offset yang sama. Misalnya, mengubah VPC dari `10.0.0.0/16` untuk `10.20.0.0/16` menggeser subnet dari ke. `10.0.1.0/24` `10.20.1.0/24`

  Aturan grup keamanan yang sama persis dengan CIDR VPC lama diperbarui secara otomatis. Aturan yang sebagian tumpang tindih atau tidak cocok dengan CIDR lama tidak diubah - tinjau aturan ini setelah perubahan.
- Ganti nama: Ubah nama VPC agar selaras dengan konvensi penamaan organisasi Anda untuk alokasi biaya, pelacakan kepatuhan, dan standar operasional.
- Ubah ukuran: Ubah panjang awalan CIDR VPC untuk memperluas atau mengurangi rentang alamat IP.
  - Panjang awalan berkurang (lebih banyak IP, misalnya /20 hingga/16): Subnet masih muat dalam kisaran yang lebih besar. Tidak perlu perubahan subnet.
  - Peningkatan panjang awalan (IP lebih sedikit, misalnya /16 hingga /20): Subnet yang berada di luar rentang baru harus diubah ukurannya terlebih dahulu menggunakan operasi pengubahan ukuran subnet.

  Aturan grup keamanan yang sama persis dengan CIDR VPC lama diperbarui secara otomatis. Aturan yang sebagian tumpang tindih atau tidak cocok dengan CIDR lama tidak diubah - tinjau aturan ini setelah mengubah ukuran.

  Ubah ukuran persyaratan:

  - CIDR baru harus antara /16 dan /28.
  - CIDR baru tidak boleh tumpang tindih dengan VPC lain di jaringan (topologi Hub dan Spoke).

  - Saat mengurangi CIDR, semua subnet yang ada harus sesuai dengan CIDR baru. Ubah ukuran subnet terlebih dahulu jika diperlukan.
- Split: Bagilah VPC menjadi dua VPC berdasarkan batas CIDR yang Anda berikan. Subnet ditugaskan ke VPC baru yang CIDR-nya berisi mereka. Grup keamanan dikloning ke kedua VPC baru, tetapi CIDR aturan grup keamanan tidak diperbarui secara otomatis - tinjau aturan Anda setelah pemisahan untuk memastikan komunikasi lintas VPC berfungsi seperti yang diharapkan. VPC asli digantikan oleh dua VPC baru.

  Persyaratan split:

  - Anda harus memberikan tepat dua rentang CIDR.
  - Kedua CIDR tidak boleh tumpang tindih.
  - Setiap CIDR harus antara /16 dan /28.
  - Setiap subnet harus sesuai dengan salah satu dari dua CIDR. Jika ada subnet yang tidak cocok, operasi ditolak.

Operasi berikut tersedia untuk subnet:

- Ubah alamat IP: Ubah alamat IP dasar CIDR subnet sambil menjaga panjang awalan yang sama.
- Hapus: Hapus subnet secara permanen dari konfigurasi tanpa memengaruhi VPC induk.
- Ubah ukuran: Ubah panjang awalan CIDR subnet untuk memperluas atau mengurangi rentang alamat IP.

  Persyaratan pengubahan ukuran subnet:

  - CIDR baru harus antara /16 dan /28.
  - CIDR baru tidak boleh tumpang tindih dengan subnet lain di VPC yang sama.
  - CIDR baru harus berada dalam CIDR VPC induk.

Setelah setiap operasi, AWS Transform mengevaluasi kembali referensi grup keamanan, yang mungkin mengubah CIDR-based aturan menjadi referensi grup keamanan atau sebaliknya. Tinjau aturan grup keamanan Anda setelah Anda membuat perubahan untuk memverifikasi bahwa aturan tersebut memenuhi persyaratan Anda.

Rekomendasi jaringan terpandu

AWS Transform menganalisis jaringan Anda yang dipetakan dan memberikan rekomendasi yang diprioritaskan melalui antarmuka obrolan. Rekomendasi didasarkan pada data jaringan Anda dan memerlukan konfirmasi Anda sebelum perubahan apa pun diterapkan.

AWS Transform mungkin merekomendasikan pengoptimalan berikut:

- Resolusi konflik CIDR: Bendera rentang CIDR yang tumpang tindih antara VPC yang dipetakan dan VPC yang ada di semua akun di Organisasi Anda. AWS VPC yang saling bertentangan ditampilkan terlebih dahulu. Anda dapat menyelesaikan konflik dengan menangani ulang VPC yang dipetakan, mengecualikan atau menghapusnya, atau mengakui konflik dan menyelesaikannya sendiri setelah penerapan.
- Standarisasi penamaan: Menandai nama VPC yang tidak mengikuti pola yang konsisten (misalnya, nama yang berisi referensi perangkat keras). AWS Transform meminta konvensi penamaan cloud Anda sebelum mengusulkan penggantian.
- Tinjauan lingkup: Mengidentifikasi segmen jaringan yang mungkin tidak perlu dimigrasi AWS, seperti sistem lama atau konstruksi yang menunggu penonaktifan. AWS Transform meminta konfirmasi Anda sebelum mengecualikan konstruksi apa pun.
- Ukuran kanan kapasitas VPC: Permukaan VPC di mana CIDR tampak terlalu besar atau berukuran kecil untuk subnet yang dikandungnya. AWS Transform menyajikan data kapasitas saat ini dan memungkinkan Anda memutuskan apakah akan mengubah ukuran.
- Tinjauan keamanan: Menandai aturan grup keamanan yang mengizinkan lalu lintas masuk yang tidak dibatasi (0.0.0. 0/0) untuk ulasan Anda.
- Konsolidasi VPC: Mengidentifikasi VPC terfragmentasi yang tampak terpisah oleh batas infrastruktur fisik daripada persyaratan isolasi logis, dan menyarankan untuk menggabungkannya.

> **Note**
>
> Semua rekomendasi memerlukan konfirmasi eksplisit Anda sebelum AWS Transform menerapkan perubahan apa pun. AWS Transform menyajikan trade-off ketika rekomendasi memengaruhi beberapa aspek jaringan Anda.

Langkah 6: Diagram jaringan

Setelah meninjau konfigurasi VPC yang dihasilkan, Anda dapat secara opsional menghasilkan diagram jaringan untuk memvisualisasikan topologi jaringan Anda. AWS Transform mendukung format diagram berikut:

- Kode putri duyung (.mmd): Format ini menghasilkan file definisi diagram berbasis teks yang dapat Anda render dengan alat. Mermaid-compatible
- Gambar (.png): Format ini menghasilkan gambar yang dirender dari topologi jaringan Anda.

Langkah 7: Konfigurasikan penandaan sumber daya

Sumber daya jaringan Anda ditandai untuk peluncuran dan replikasi. Anda juga dapat menambahkan tag kustom dan tag AWS Migration Acceleration Program (MAP).

Tag otomatis untuk peluncuran dan replikasi

AWS Transform secara otomatis menandai sumber daya jaringan Anda yang dimigrasi (VPC, subnet, grup keamanan, dan tabel rute) dengan tag berikut:

- Kunci: **`CreatedBy`** Nilai: `AWSApplicationMigrationService`
- Kunci: **`ATWorkspace`** Nilai: *`workspace-id`*

Tag ini memungkinkan penggunaan VPC dan subnet untuk meluncurkan instance pengujian dan cutover di. AWS

> **Note**
>
> VPC dan subnet yang dimigrasi tidak menyertakan konektivitas internet secara default, sehingga tidak cocok sebagai area pementasan untuk replikasi.

Untuk juga menggunakan VPC dan subnet sebagai area pementasan (replikasi), tambahkan tag berikut secara manual:

- Kunci: **`CreatedFor`** Nilai: `AWSTransform`
- Kunci: **`ATWorkspace`** Nilai: *`workspace-id`*

Anda juga dapat menerapkan tag ini ke sumber daya AWS jaringan yang ada untuk membuatnya tersedia untuk replikasi.

Temukan ID ruang kerja Anda di URL aplikasi web AWS Transform: https://... / workspace//*workspace-id*-id job/job

Tag kustom

Selain tag yang diterapkan secara otomatis oleh AWS Transform, Anda dapat menambahkan tag khusus secara opsional untuk mengatur, melacak biaya, dan mengelola kepatuhan untuk sumber daya jaringan yang dimigrasi. Anda dapat menerapkan tag khusus pada dua tingkat:

- Job-level tag: Terapkan ke semua sumber daya yang dibuat oleh pekerjaan ini, termasuk semua VPC, subnet, grup keamanan, dan tabel rute.
- VPC-level tag: Terapkan ke VPC tertentu dan secara otomatis mengalir ke semua sumber daya terkait (subnet, grup keamanan, tabel rute).

> **Note**
>
> Maksimal 40 tag per permintaan. Setiap tag membutuhkan kunci dan nilai. AWS konvensi penandaan berlaku.

AWS Transform menerapkan tag ini ketika menghasilkan Infrastruktur sebagai template Kode.

AWS Program Akselerasi Migrasi

Jika migrasi Anda merupakan bagian dari AWS Migration Acceleration Program (MAP 2.0), AWS Transform akan menerapkan tag MAP ke sumber daya Anda. Jika Anda memberikan ID MPE Anda sebelumnya dalam proses migrasi, tag diterapkan secara otomatis. Jika tidak, setelah Anda selesai meninjau konfigurasi VPC yang dihasilkan AWS , Transform menanyakan apakah Anda memiliki perjanjian MAP dan meminta Anda untuk memberikan ID MPE Anda — kode 10 karakter menggunakan huruf besar dan digit (misalnya, ABCDE12345). Tag yang diterapkan menggunakan format:

- Kunci: **`map-migrated`** Nilai: `mig`*`MPE_ID`*

Langkah 8: Menyebarkan jaringan Anda

Setelah menandai, pilih strategi penerapan Anda:

- AWS Transform-managed deployment: AWS Transform menggunakan CloudFormation template untuk menyebarkan jaringan Anda dan menjalankan Reachability Analyzer untuk memeriksa konektivitas antar subnet di beberapa VPC dan dalam VPC yang sama.

  > **Note**
  >
  > Anda harus mendapatkan persetujuan eksplisit sebelum permintaan penyebaran jaringan Anda berjalan. Lihat Proses persetujuan penerapan.

- Self-deployment: AWS Transform menghasilkan template Infrastructure as Code (IAc). CloudFormation template dihasilkan secara default. Anda juga dapat memilih format output tambahan:
  - AWS CDK menghasilkan TypeScript proyek untuk penyebaran infrastruktur terprogram.
  - HashiCorp Terraform menghasilkan template HashiCorp Configuration Language (HCL) untuk mengelola sumber daya jaringan.
  - Landing Zone Accelerator (LZA) menghasilkan file network-config.yaml untuk konfigurasi jaringan LZA.

> **Note**
>
> Saat Anda menerapkan melalui pipeline Landing Zone Accelerator (LZA), akun AWS Transform dan instalasi LZA Anda harus berada di Organisasi yang sama. AWS Penerapan akan gagal jika ada ketidakcocokan antara ID Organisasi.

Untuk penyebaran sendiri, gunakan tautan yang disediakan untuk mengunduh file zip yang berisi templat yang dihasilkan. Folder zip menyertakan README.md file yang menjelaskan cara menggunakan templat yang dihasilkan.

Untuk memverifikasi file yang diunduh tidak rusak atau dirusak, buat dan unduh checksum, lalu bandingkan dengan hash yang dihasilkan secara lokal menggunakan perintah. `openssl dgst -sha256 -binary <file.zip> | base64`

Proses persetujuan penerapan

Untuk memastikan bahwa perubahan jaringan sesuai dengan standar keamanan dan persyaratan arsitektur organisasi Anda, semua permintaan penerapan melalui alur kerja persetujuan. Anda harus mendapatkan persetujuan eksplisit sebelum permintaan penyebaran jaringan Anda berjalan. Saat Anda mengirimkan permintaan penerapan, permintaan tersebut secara otomatis merutekan ke pemberi persetujuan resmi melalui tab AWS Transform Approvals. Pemberi persetujuan memvalidasi CloudFormation templat dan konfigurasi jaringan untuk memastikan kepatuhan dengan standar keamanan dan persyaratan arsitektur. Setiap pengiriman memicu siklus peninjauan baru, dan penerapan hanya dilanjutkan setelah menerima konfirmasi. Jika pemberi persetujuan menolak permintaan Anda, hubungi mereka secara langsung untuk mendiskusikan modifikasi yang diperlukan. AWS Transform melacak semua keputusan persetujuan untuk tujuan audit dan mempertahankan riwayat penerapan.

Hapus sumber daya jaringan yang digunakan

Jika Anda perlu memutar kembali penerapan, Anda dapat menghapus sumber daya jaringan yang digunakan AWS Transform. Anda dapat menghapus sumber daya segera setelah penerapan selesai. Jika Anda mengubah sumber daya jaringan yang digunakan setelah penerapan, sumber daya tidak dapat dihapus secara otomatis.

- AWS Transform-managed deployments: AWS Transform menghapus semua CloudFormation tumpukan yang dibuat selama penerapan. Tindakan ini memerlukan persetujuan melalui tab AWS Transform Approvals.
- Self-deployments: Anda harus menghapus sumber daya yang digunakan secara manual melalui AWS Management Console atau AWS CLI.

Ekstraksi file konfigurasi

Jika lingkungan sumber Anda menggunakan Cisco ACI, Palo Alto Networks, atau Fortinet FortiGate, Anda perlu mengekstrak file konfigurasi untuk menyediakan Transform. AWS Anda dapat menggunakan file-file ini sebagai file sumber mandiri untuk menghasilkan infrastruktur jaringan dan grup keamanan, atau sebagai file pelengkap bersama unggahan RVTools untuk menambahkan pembuatan grup keamanan. Proses ekstraksi sama dalam kedua kasus.

Untuk mengekstrak file konfigurasi dari firewall dan lingkungan jaringan Anda, ikuti prosedur ini. Konsultasikan dokumentasi vendor untuk informasi terbaru.

Fortinet FortiGate

- Versi firmware harus v7.0 atau yang lebih baru.
- Anda membutuhkan `super_admin` atau `super_admin_readonly` hak istimewa di tingkat global.
- Langkah:
  1. Connect ke firewall melalui SSH atau built-in CLI client
  2. Jalankan: `show | grep ""` (`| grep ""`menonaktifkan pagination)
  3. Simpan semua output ke file mulai dari `show` perintah

Jaringan Palo Alto

- Versi firmware harus 10.1 atau lebih baru.
- Anda membutuhkan peran superadmin.
- Connect ke firewall melalui SSH, jalankan perintah berikut untuk menonaktifkan pagination, mengatur format output, masuk ke mode konfigurasi, dan ekspor konfigurasi dan objek yang telah ditentukan. Simpan output:

```
set cli pager off
set cli config-output-format set
configure
show              # Save as palo-conf.txt
show predefined   # Save as palo-default.txt
```

Cisco ACI

- Versi firmware harus 6.0 atau lebih baru.
- Anda memerlukan peran Admin dengan semua hak istimewa dan tujuan Secure Copy Protocol (SCP), SSH File Transfer Protocol (SFTP), atau File Transfer Protocol (FTP) yang dikonfigurasi.
- Langkah:
  1. Connect ke Application Policy Infrastructure Controller (APIC) melalui browser
  2. Buka menu Admin dan pilih Config Rollbacks
  3. Dalam dialog Ambil snapshot, pilih opsi lokasi jarak jauh dan pilih Buat snapshot sekarang.
  4. Setelah menerima pesan “Transfer sukses”, sambungkan ke server lokasi jarak jauh dan ambil file snapshot terbaru (file.gz)

Migrasi server

AWS Transform menggunakan AWS Transform MGN (MGN) untuk meng-host kembali server Anda ke Amazon EC2. Alur kerja server migrasi memandu Anda dalam menyiapkan setiap gelombang migrasi, memvalidasi inventaris server Anda, menerapkan agen replikasi, memantau replikasi data, menguji instance yang dimigrasi, dan melakukan pemotongan akhir. Untuk membaca lebih lanjut tentang itu, lihat Apa itu AWS Transform MGN? di Panduan Pengguna MGN.

Migrasi server diatur oleh gelombang. Setiap gelombang mewakili sekelompok server yang dimigrasikan bersama. Untuk setiap gelombang, Anda menyelesaikan fase berikut:

Untuk gelombang dengan strategi migrasi containerize, AWS Transform menjalankan alur kerja containerization kode sumber alih-alih langkah rehost yang dijelaskan di bawah ini. Alur kerja kontainerisasi memandu Anda melalui kloning kode sumber, menghasilkan artefak Docker, menerbitkan gambar kontainer, dan menerapkan ke Amazon Elastic Container Service atau Amazon Elastic Kubernetes Service. Untuk alur kerja kontainerisasi lengkap, lihat. Kontainerisasi kode sumber

1. Prasyarat dan Konfigurasi Default Migrasi
2. Langkah 1: Siapkan gelombang migrasi
3. Langkah 2: Validasi dan konfirmasi inventaris
4. Langkah 3: Menyebarkan agen replikasi
5. Langkah 4: Replikasi data
6. Langkah 5: Pengujian
7. Langkah 6: Cutover

Prasyarat dan Konfigurasi Default Migrasi

Prasyarat

Sebelum memulai migrasi rehost, pastikan Anda memiliki yang berikut:

> **Note**
>
> Jika Anda menyelesaikan semua langkah pekerjaan migrasi ujung ke ujung di AWS Transform, akun target dan file inventaris Anda sudah disiapkan — file inventaris akan dibuat untuk Anda selama langkah perencanaan migrasi. Infrastruktur jaringan yang disiapkan

melalui migrasi jaringan AWS Transform juga siap. Jika Anda tidak membangun infrastruktur jaringan melalui AWS Transform, pastikan sudah disiapkan terlebih dahulu sebelum memulai migrasi rehost.
Sebelum memulai migrasi rehost, verifikasi bahwa Anda memiliki sumber daya jaringan dan infrastruktur untuk meng-host server Anda. Anda dapat menggunakan AWS Transform landing zone dan kemampuan migrasi jaringan atau alat lainnya untuk itu.

- Sistem operasi yang didukung — Server sumber harus menjalankan sistem operasi yang didukung. Untuk daftar lengkapnya, lihat Sistem operasi yang didukung di Panduan Pengguna MGN.
- Akun target untuk migrasi — Akun AWS ID tempat Anda memerlukan server untuk dimigrasi. Anda dapat menggunakan AWS Transform landing zone atau alat lainnya untuk menyiapkan infrastruktur Anda.
- Infrastruktur jaringan di tempat - VPC, subnet, dan grup keamanan dikerahkan dan dikonfigurasi. Anda dapat menggunakan AWS Transform Network Migration atau alat lainnya untuk menyiapkan infrastruktur jaringan Anda.
- File inventaris — Disiapkan dengan detail server, penetapan gelombang, informasi akun target, dan preferensi jenis instans Amazon EC2. Anda dapat menggunakan AWS Transform perencanaan migrasi untuk menghasilkan file ini.

Konfigurasikan Default Migrasi

Sebelum memulai eksekusi migrasi multi-akun, Anda harus mengonfigurasi pengaturan default yang berlaku untuk semua akun target Anda. Default ini menentukan cara instans Amazon EC2 diluncurkan dan cara migrasi umum dikonfigurasi. Anda dapat mengganti default ini pada tingkat gelombang selama pengaturan gelombang.

Preferensi rekomendasi Amazon EC2

AWS Transform memberikan rekomendasi jenis instans Amazon EC2 berdasarkan spesifikasi pemanfaatan VM sumber Anda. Anda dapat mengonfigurasi preferensi rekomendasi Amazon EC2 untuk mengontrol cara jenis instans dipilih untuk server yang dimigrasi.

Untuk informasi selengkapnya tentang membuat rekomendasi Amazon EC2, lihat Menghasilkan rekomendasi Amazon EC2 di. AWS Migration Hub

> **Note**
>
> Anda dapat mengubah jenis instans Amazon EC2 yang disarankan untuk menyertakan rekomendasi dari Penilai Migrasi, Penilaian AWS Pengoptimalan, dan Lisensi (OLA), atau pekerjaan penilaian Transform. AWS

Inisialisasi migrasi

Untuk memulai migrasi, AWS Transform menginisialisasi MGN untuk setiap Wilayah AWS tempat Anda berencana untuk bermigrasi, serta semua akun target yang akan digunakan layanan. Selama proses inisialisasi:

- Peran dan kebijakan IAM yang diperlukan dibuat.
- Templat default yang diperlukan dikonfigurasi.

Untuk informasi tentang proses inisialisasi, lihat Menginisialisasi AWS Transform MGN dengan konsol di Panduan Pengguna MGN.

Templat peluncuran Amazon EC2

Pengaturan peluncuran mencakup dua bagian: pengaturan peluncuran umum, dan templat peluncuran Amazon EC2, yang menentukan bagaimana instance pengujian atau cutover diluncurkan untuk setiap server sumber. AWS

Pengaturan peluncuran, termasuk template peluncuran Amazon EC2, dapat ditentukan pada tingkat akun dan kemudian diterapkan ke setiap server sumber secara otomatis setiap kali Anda menambahkan server sumber ke AWS Transform MGN. Pengaturan peluncuran default yang ditentukan di bagian ini dapat diterapkan ke semua akun target Anda secara otomatis.

AWS Transform menyajikan daftar pengaturan template peluncuran yang tersedia. Anda dapat memilih untuk melanjutkan dengan default atau mengkonfigurasi template peluncuran Anda. Jika Anda memilih untuk mengonfigurasi, AWS Transform menyediakan tautan ke tinjauan human-in-the-loop (HITL) yang berisi semua parameter pengaturan template peluncuran. Anda juga dapat melakukan modifikasi langsung melalui antarmuka obrolan untuk parameter apa pun yang Anda inginkan.

Server sumber dibuat dengan pengaturan template peluncuran akun. Setelah server sumber dibuat dengan pengaturan default ini, Anda dapat mengubahnya di tingkat pengaturan peluncuran server

sumber. Anda dapat mengubah pengaturan server sumber pada parameter apa pun menggunakan antarmuka obrolan, atau untuk operasi massal menggunakan file Excel inventaris selamaLangkah 2: Validasi dan konfirmasi inventaris.

Untuk meninjau daftar lengkap pengaturan dan detail templat peluncuran, lihat Meluncurkan pengaturan umum di Panduan Pengguna MGN.

Perubahan template peluncuran Amazon EC2 tambahan

Untuk perubahan template peluncuran Amazon EC2 tambahan, Anda harus melakukannya pada ID templat untuk setiap akun target. Opsi ini tersedia di dalam pengaturan gelombang. AWS Transform memandu Anda melaluinya dan menyediakan tautan yang sesuai.

Langkah 1: Siapkan gelombang migrasi

Pada fase ini, AWS Transform menyiapkan gelombang migrasi dengan mengonfigurasi akun target, memverifikasi izin layanan, menyiapkan tag sumber daya, menambahkan data jaringan ke inventaris Anda, dan mengonfigurasi pengaturan replikasi dan peluncuran.

Mode migrasi dan konfigurasi akun

AWS Transform mendukung dua mode migrasi:

- Single-account migrasi — Semua server dalam gelombang bermigrasi ke akun target yang sama yang dikonfigurasi di konektor Anda.
- Multi-account migrasi — Server bermigrasi ke akun target berbeda yang ditentukan dalam file inventaris Anda. Untuk migrasi multi-akun, file inventaris Anda harus menyertakan `mgn:account-id` kolom dengan ID akun target untuk setiap server.

AWS Transform mengonfirmasi konfigurasi akun target dan memverifikasi bahwa MGN diinisialisasi di setiap akun target. Jika MGN belum diinisialisasi, AWS Transform memberikan instruksi untuk menyelesaikan inisialisasi. Selama inisialisasi, MGN membuat peran layanan IAM berikut untuk operasi replikasi dan peluncuran:

- `AWSApplicationMigrationReplicationServerRole`
- `AWSApplicationMigrationConversionServerRole`
- `AWSApplicationMigrationMGHRole`
- `AWSApplicationMigrationLaunchInstanceWithDrsRole`

- `AWSApplicationMigrationLaunchInstanceWithSsmRole`
- `AWSApplicationMigrationAgentRole`

Untuk mempelajari lebih lanjut tentang peran ini, lihat Menginisialisasi MGN dengan konsol atau Menginisialisasi MGN dengan API di Panduan Pengguna MGN.

Untuk migrasi multi-akun, AWS Transform juga membuat peran berikut selama langkah inisialisasi:. `AWSTransformRehostSharingRole_<management-or-delegated-admin-account-id>` Peran ini diterapkan di semua akun target migrasi.

Verifikasi penandaan sumber daya

Setelah izin layanan dikonfirmasi, AWS Transform memverifikasi bahwa semua sumber daya yang diperlukan ditandai dengan benar agar migrasi berhasil dioperasikan oleh agen. Jika sumber daya tidak memiliki tag yang diperlukan, AWS Transform menyediakan tautan ke halaman penandaan tempat Anda dapat menerapkan tag yang hilang sebelum melanjutkan. Tag berikut diperlukan:

- Server sumber yang ada harus memiliki tag `CreatedBy: AWSTransform` dan`ATWorkspace: <workspace_id>`. Jika Anda sudah memulai replikasi pada server sumber dan membuatnya di layanan AWS Transform MGN, Anda perlu menandai server ini agar AWS Transform dapat menghubungkannya dengan server sumber yang ditemukan dari lingkungan lokal Anda, dan menghindari pembuatan server sumber duplikat yang tidak perlu. AWS Transform secara otomatis berkorelasi di antara mereka menggunakan ID yang disediakan pengguna, FQDN, atau kunci nama host.
- Sumber daya jaringan harus diberi tag dengan benar untuk replikasi (area pementasan) dan instance peluncuran. AWS Transform menampilkan daftar lengkap sumber daya jaringan di akun target Anda, dengan indikasi apakah setiap sumber daya sudah ditandai atau belum. Anda dapat meninjau daftar dan memilih sumber daya yang tidak ditandai yang ingin Anda tambahkan. Untuk setiap sumber daya yang Anda pilih, AWS Transform menerapkan tag yang relevan:
  - `CreatedBy: AWSTransform`atau`CreatedFor: AWSTransform`, tergantung pada jenis sumber daya.
  - `ATWorkspace: <workspace_id>`diterapkan ke semua sumber daya yang dipilih.

  VPC dan subnet yang dibuat oleh agen migrasi jaringan AWS Transform secara otomatis ditandai.
- Selain VPC dan subnet, AWS Transform juga menampilkan semua Elastic Network Interfaces (ENI) yang ada di akun target Anda. Jika Anda ingin AWS Transform menggunakannya sebagai bagian dari peluncuran instans Anda, mereka harus ditandai dengan `CreatedFor:`

`AWSTransform` dan`ATWorkspace: <workspace_id>`. Untuk informasi selengkapnya tentang cara melampirkan atau menambahkan ENI ke templat peluncuran Amazon EC2, lihat Pertimbangan terperinci di Panduan Pengguna MGN.

Tambahkan data jaringan ke inventaris

AWS Transform menambahkan informasi jaringan dari migrasi jaringan Anda ke file inventaris. Langkah ini memetakan server Anda ke subnet target dan grup keamanan yang sesuai berdasarkan konfigurasi jaringan yang dihasilkan selama fase jaringan migrasi.

Pengaturan replikasi dan peluncuran

Konfigurasi pengaturan replikasi

Pengaturan replikasi menentukan bagaimana data direplikasi dari server sumber Anda ke. AWS Konfigurasikan pengaturan replikasi di template replikasi sebelum menambahkan server sumber ke AWS Transform MGN. AWS Transform menunjukkan kepada Anda semua parameter pengaturan replikasi — Anda dapat mengonfigurasinya melalui HITL khusus atau melalui antarmuka obrolan.

Untuk detail selengkapnya tentang parameter pengaturan replikasi, lihat Templat pengaturan replikasi di Panduan Pengguna MGN.

Luncurkan pengaturan template

Template peluncuran memungkinkan Anda untuk mengontrol cara AWS Transform MGN meluncurkan instance. AWS Konfigurasi default yang ditentukan dalam template secara otomatis diterapkan ke setiap server yang baru ditambahkan. Anda dapat mengonfigurasi pengaturan template peluncuran melalui HITL khusus atau melalui antarmuka obrolan.

Untuk detail selengkapnya tentang parameter pengaturan template peluncuran, lihat Peluncuran template di Panduan Pengguna MGN.

AWS Transform juga menyediakan tautan ke ID template peluncuran Amazon EC2 yang terkait dengan template peluncuran, memungkinkan Anda mengubah atribut template peluncuran Amazon EC2 tambahan. Untuk mengedit template peluncuran Amazon EC2, ikuti petunjuk di Template Luncurkan di Panduan Pengguna MGN.

Strategi penugasan IP

Anda memilih bagaimana alamat IP ditetapkan ke server yang dimigrasi:

- IP statis — Alamat IP server sumber dipertahankan. Jika transformasi CIDR diperlukan, AWS Transform secara otomatis mengonversi alamat IP agar sesuai dengan CIDR baru.
- Dynamic IP (DHCP) — Setiap server diberi alamat IP baru dari kumpulan IP subnet.

> **Note**
>
> Jika Anda memilih strategi pemetaan grup keamanan MAP selama migrasi jaringan, hanya penetapan IP statis yang tersedia. Untuk detail selengkapnya, lihat Pemetaan grup keamanan.

Langkah 2: Validasi dan konfirmasi inventaris

Sebelum memuat data server Anda ke MGN, AWS Transform menyiapkan file inventaris untuk ditinjau. Anda dapat mengunduh file dalam format CSV atau XLSX, meninjau konfigurasi server, dan membuat perubahan jika diperlukan.

File inventaris mencakup detail seperti nama server, sistem operasi, rekomendasi jenis instans Amazon EC2, subnet target, grup keamanan, penetapan IP, dan opsi lisensi. Bidang yang diperlukan meliputi:

- Informasi server — Nama server, VMID, dan spesifikasi sumber.
- Penugasan gelombang - Pengelompokan gelombang migrasi.
- Pengelompokan aplikasi — Asosiasi aplikasi logis.
- Konfigurasi target — Akun target, Wilayah, dan jenis instans Amazon EC2.
- Konfigurasi jaringan — Target subnet dan grup keamanan.

Anda dapat memodifikasi file untuk menyesuaikan konfigurasi Amazon EC2, mengubah opsi lisensi sistem operasi (BYOL atau Termasuk Lisensi), dan memperbarui pengaturan penyewaan.

Setelah meninjau inventaris, Anda dapat menerimanya seperti yang ditunjukkan atau mengunggah versi modifikasi. AWS Transform kemudian memuat data ke MGN, yang menciptakan catatan server sumber untuk setiap server dalam gelombang.

> **Note**
>
> Jangan menghapus kolom atau mengubah header kolom dalam file inventaris. AWS Transform membutuhkan struktur file asli untuk memproses data dengan benar.

> **Note**
>
> AWS Transform memungkinkan satu impor ke target Akun AWS dan target tertentu Wilayah AWS pada suatu waktu. Jika Anda mengerjakan lebih dari satu gelombang secara bersamaan, atau jika ada lebih dari satu pekerjaan migrasi yang berjalan dengan akun target yang sama, Anda harus menunggu impor selesai sebelum dapat melakukan impor lain dalam gelombang atau pekerjaan yang berbeda.

Anda dapat mengontrol opsi lisensi sistem operasi (BYOL atau Termasuk Lisensi) dan penyewaan dengan menentukan konfigurasi di kolom file inventaris dan. `mgn:launch:placement:operating-system-licensing` `mgn:launch:placement:tenancy` Untuk informasi selengkapnya, lihat Mengimpor parameter di Panduan Pengguna MGN.

Langkah 3: Menyebarkan agen replikasi

Untuk mulai mereplikasi data dari server sumber Anda AWS, Anda menginstal Agen AWS Replikasi di setiap server sumber. AWS Transform menawarkan tiga metode instalasi:

- Alat organisasi — Gunakan alat penyebaran organisasi Anda yang ada (seperti SCCM, Ansible, atau Chef) untuk menginstal agen di seluruh server Anda. AWS Transform menyediakan perintah instalasi dengan parameter tambahan untuk instalasi diam, termasuk`--no-prompt`,`--aws-access-key-id`,`--aws-secret-access-key`, dan`--aws-session-token`.
- Konektor MGN - Gunakan konektor MGN untuk mengotomatiskan instalasi agen. Konektor terhubung ke mesin sumber melalui SSH (Linux) atau WinRM (Windows) dan menginstal agen replikasi secara otomatis. Setelah dikonfigurasi, konektor dapat digunakan kembali di beberapa gelombang dan target Akun AWS yang berbeda. Untuk informasi selengkapnya tentang konektor MGN, lihat Mengatur Konektor MGN di Panduan Pengguna MGN.

> **Note**
>
> Sebelum menggunakan konektor MGN dengan AWS Transform, Anda harus menandai instance terkelola konektor di AWS Systems Manager Fleet Manager dengan tag berikut:
>
> - Kunci: `CreatedFor` Nilai: `AWSTransform`
> - Kunci: `ATWorkspace` Nilai: *`workspace-id`*
>
> Untuk menandai instance terkelola, buka AWS Systems Manager konsol, navigasikan ke Fleet Manager di bawah Node Tools, pilih instance terkelola konektor MGN Anda, dan terapkan tag di atas. Temukan ID ruang kerja Anda di URL aplikasi web AWS Transform:`https://.../workspace/`*`workspace-id`*`/job/`*`job-id`*.

- Instalasi manual — Instal agen langsung di setiap server sumber. Metode ini membutuhkan akses langsung ke setiap server tetapi memberi Anda kontrol penuh atas proses instalasi.

AWS Ubah pengaturan konektor MGN

Konektor AWS Transform MGN mengotomatiskan penyebaran agen replikasi ke server sumber Anda. Konektor adalah klien ringan yang digunakan pada mesin Linux khusus di lingkungan lokal Anda. Ini terhubung ke server sumber melalui SSH (Linux) atau WinRM (Windows) untuk menginstal dan mengkonfigurasi agen replikasi, menghilangkan kebutuhan untuk berkoordinasi secara manual di beberapa layanan. AWS

Cara kerja konektor

Konektor beroperasi melalui komponen-komponen berikut:

- Connector client — Diterapkan pada mesin Linux khusus di lingkungan Anda.
- Agen SSM - Diinstal pada mesin yang sama untuk memungkinkan komunikasi yang aman dengan AWS.
- SSM Hybrid Activation — Menautkan mesin konektor ke AWS Systems Manager untuk eksekusi perintah yang aman.
- Manajemen kredensil - Mengambil kredensi server sumber dari Secrets Manager. AWS

Saat Anda menggunakan agen, AWS Transform mengirimkan dokumen SSM ke mesin konektor. Konektor kemudian mengambil kredensi server sumber dari AWS Secrets Manager, membuat

koneksi ke setiap server sumber, memvalidasi bahwa server sumber memenuhi prasyarat, menginstal dan mengonfigurasi agen replikasi, dan memverifikasi instalasi yang berhasil.

Persyaratan mesin konektor

Persyaratan	Detail
Sistem operasi	Sistem operasi Linux yang didukung. Untuk daftar lengkapnya, lihat prasyarat konektor MGN di Panduan Pengguna MGN.
Akses jaringan	Harus mencapai semua server sumber (Linux melalui SSH, Windows melalui WinRM)
Konektivitas internet	HTTPS keluar (443) ke AWS titik akhir (Systems Manager, Secrets Manager, MGN)
Ruang disk	Minimal 200 MB gratis
Izin	Akses root atau sudo

> **Note**
>
> Konektor harus diinstal pada mesin Linux, tetapi dapat menyebarkan agen ke server sumber Linux dan Windows.

Proses penyiapan

AWS Transform memandu Anda melalui langkah-langkah berikut untuk mengatur konektor:

Langkah 1: Konfigurasi konektor

Berikan nama untuk konektor Anda, atau gunakan nama default yang dibuat secara otomatis. Konektor dapat diinstal pada akun manajemen atau pada akun administrator yang didelegasikan di MGN. Untuk migrasi multi-akun, konektor dapat menyebarkan agen ke server di seluruh akun anggota.

Langkah 2: pengaturan AWS sumber daya

AWS Transform membuka halaman penyiapan yang berjalan di browser Anda menggunakan AWS kredensil Anda. Anda harus masuk ke Konsol AWS Manajemen dengan akun manajemen atau akun administrator yang didelegasikan. Ini harus akun yang sama dengan konektor target AWS Transform Anda terhubung.

Halaman penyiapan secara otomatis membuat sumber daya berikut:

- Peran IAM (dibuat idempotently — dilewati jika sudah ada):
  - `AWSApplicationMigrationConnectorManagementRole`— Digunakan selama instalasi agen untuk mengakses kredensil.
  - `AWSApplicationMigrationConnectorSharingRole_<ACCOUNT-ID>`- Berisi izin untuk instalasi agen.
- SSM Hybrid Activation — periode kedaluwarsa 30 hari. Menautkan mesin konektor ke AWS Systems Manager dan menghasilkan kredenal aktivasi yang aman.

Atau, Anda dapat mengunduh CloudFormation template dari halaman penyiapan untuk menerapkan peran IAM sendiri.

Halaman pengaturan menghasilkan perintah instalasi satu baris dengan semua kredensi dan konfigurasi yang diperlukan.

> **Important**
>
> Biarkan halaman penyiapan tetap terbuka sampai instalasi selesai. Menutupnya akan membutuhkan restart proses. Semua kredensil hanya ada di browser Anda dan tidak disimpan oleh AWS Transform.

Langkah 3: Instalasi konektor

Instal konektor pada mesin Linux di lingkungan Anda:

1. Salin tautan instalasi dari halaman pengaturan.
2. SSH ke mesin Linux pilihan Anda.
3. Tempel dan jalankan perintah instalasi.
4. Tunggu hingga instalasi selesai (biasanya 2-3 menit).

Langkah 4: Lampirkan server sumber

Setelah instalasi, AWS Transform mengidentifikasi semua server sumber yang termasuk dalam gelombang saat ini dan secara otomatis menempelkannya ke konektor MGN.

Langkah 5: Konfigurasikan kredensil

Berikan AWS Secrets Manager ARN untuk kredensi server sumber Anda. AWS Transform menawarkan tiga opsi konfigurasi kredensi:

- Rahasia tunggal untuk server Linux - Satu rahasia bersama yang berisi kunci SSH atau username/password untuk semua server sumber Linux.
- Rahasia tunggal untuk server Windows - Satu rahasia bersama yang berisi nama pengguna dan kata sandi untuk semua server sumber Windows.
- Beberapa rahasia per server — Rahasia berbeda per server atau grup server. Gunakan ini ketika server memiliki kredensil yang berbeda. AWS Transform menghasilkan file CSV yang telah diisi sebelumnya dengan daftar server Anda. Anda mengisi `secret_arn` kolom untuk setiap server dan mengunggah file yang sudah selesai.

> **Note**
>
> Anda dapat menggabungkan opsi rahasia tunggal Linux dan Windows jika Anda memiliki kedua jenis server dengan satu rahasia bersama masing-masing. Opsi rahasia per-server saling eksklusif dengan opsi rahasia tunggal.

Format rahasia kredensi. Untuk membaca lebih lanjut tentang hal itu, lihat kredenal konektor MGN di Panduan Pengguna MGN:

```
{
  "WinConnectionProtocol": "HTTPS",
  "WinUserName": "windows_username",
  "WinPassword": "windows_password",
  "LinuxUserName": "linux_username",
  "LinuxPrivateKey": "linux_private_key",
  "LinuxHostKeyValidation": false
}
```

Penyebaran agen

Setelah kredensil dikonfigurasi dan diverifikasi, AWS Transform menyebarkan agen replikasi ke server sumber Anda. Anda dapat menyebarkan ke semua server dalam gelombang saat ini atau memilih server tertentu.

Proses penyebaran untuk setiap server:

1. AWS Transform mengirimkan perintah penerapan ke konektor melalui SSM.
2. Konektor mengambil kredensil dari Secrets Manager AWS .
3. Konektor terhubung ke server sumber menggunakan kredensi yang dikonfigurasi.
4. Konektor memvalidasi bahwa server sumber memenuhi semua prasyarat yang diperlukan untuk menjalankan agen replikasi.
5. Konektor menginstal dan mengkonfigurasi agen replikasi.
6. Konektor memverifikasi instalasi dan konektivitas yang berhasil.

Anda dapat memantau kemajuan penerapan secara real-time dengan pelacakan status per-server, termasuk langkah penginstalan saat ini, waktu yang telah berlalu, dan perkiraan waktu yang tersisa. Jika ada server yang gagal, AWS Transform menampilkan alasan kegagalan dan menawarkan opsi coba lagi per server. Server yang berhasil digunakan dapat melanjutkan secara independen sementara server yang gagal dicoba ulang.

Penggunaan kembali konektor dan siklus hidup

Saat menerapkan agen untuk gelombang berikutnya, Anda dapat menggunakan kembali konektor yang ada atau membuat yang baru. AWS Transform mencantumkan semua konektor yang dikonfigurasi di akun Anda, menampilkan nama konektor, status (Aktif atau Kedaluwarsa), jumlah server terlampir, dan tanggal kedaluwarsa Aktivasi Hybrid.

- Konektor aktif — Aktivasi Hybrid masih berlaku. AWS Transform memverifikasi peran IAM untuk gelombang baru dan melanjutkan ke konfigurasi kredensi. Tidak diperlukan Aktivasi Hybrid baru.
- Konektor kedaluwarsa - Aktivasi Hibrid SSM telah kedaluwarsa. Aktivasi yang kedaluwarsa tidak dapat diperpanjang. Anda harus memilih konektor yang berbeda atau membuat yang baru.

Aktivasi Hibrida SSM berakhir setelah 30 hari. Aktivasi hanya diperlukan untuk menginstal konektor pada mesin Linux. Setelah konektor diinstal, Anda dapat terus menggunakannya untuk menginstal agen replikasi pada server sumber bahkan setelah aktivasi berakhir. Jika Anda perlu memasang

konektor pada mesin baru setelah aktivasi kedaluwarsa, Anda perlu membuat konektor baru melalui proses pengaturan.

Instalasi agen manual

Untuk instalasi manual, pertama-tama Anda menghasilkan AWS kredensyal (sementara atau permanen) dan kemudian menginstal agen di setiap server sumber.

Opsi kredensi:

- Kredensyal sementara (disarankan) — Buat peran IAM dengan kebijakan `AWSApplicationMigrationAgentInstallationPolicy` terkelola, lalu gunakan `aws sts assume-role` untuk menghasilkan kredensyal sementara. Untuk membaca lebih lanjut tentang hal itu, lihat Izin instalasi agen di Panduan Pengguna MGN.
- Kredensyal permanen — Buat pengguna IAM dengan kebijakan `AWSApplicationMigrationAgentInstallationPolicy` terkelola dan buat kunci akses.

Langkah-langkah instalasi:

Untuk server Linux, unduh dan jalankan penginstal:

```
wget -O ./aws-replication-installer-init \
  https://aws-application-migration-service-region.s3.region.amazonaws.com/latest/
linux/aws-replication-installer-init
sudo chmod +x aws-replication-installer-init
sudo ./aws-replication-installer-init --region region --user-provided-id server-
identifier
```

Untuk server Windows, unduh dan jalankan penginstal yang sesuai menggunakan PowerShell sebagai Administrator:

```
Invoke-WebRequest -Uri "https://aws-application-migration-
service-region.s3.region.amazonaws.com/latest/windows/
AwsReplicationWindowsInstaller.exe" `
  -OutFile "C:\AwsReplicationWindowsInstaller.exe"
C:\AwsReplicationWindowsInstaller.exe --region region --user-provided-id server-
identifier
```

> **Important**
>
> parameter `--user-provided-id` diperlukan. Ganti *`server-identifier`* dengan nilai yang tepat dari `mgn:server:user-provided-id` kolom di file inventaris Anda. Pengidentifikasi ini menghubungkan server fisik ke catatan server sumber MGN-nya.

Untuk informasi selengkapnya tentang instalasi agen, lihat Agen Linux dan agen Windows di Panduan Pengguna MGN.

Setelah instalasi, AWS Transform memverifikasi bahwa semua agen berhasil terhubung dengan memeriksa bahwa server menunjukkan status replikasi atau. `INITIATING` `INITIAL_SYNC`

> **Note**
>
> AWS Transform tidak mendukung replikasi tanpa agen MGN. Untuk informasi tentang replikasi tanpa agen, lihat ikhtisar replikasi tanpa agen di Panduan Pengguna MGN.

> **Note**
>
> Anda harus menginstal agen replikasi di semua server dalam gelombang. Putuskan sambungan dan arsipkan server tempat Anda tidak menginstal agen replikasi. Anda dapat menggunakan `disconnect-from-service` perintah untuk memutuskan server, dan `mark-as-archived` perintah untuk mengarsipkan server yang terputus. Perintah pengarsipan hanya berfungsi untuk server sumber yang status siklus hidupnya. `DISCONNECTED`

Untuk kuota terkait replikasi, lihat batas kuota layanan MGN di Panduan Pengguna MGN.

Langkah 4: Replikasi data

Setelah agen replikasi diinstal, replikasi data dimulai secara otomatis. AWS Transform menggunakan replikasi tingkat blok berkelanjutan untuk menyinkronkan data dari server sumber ke. AWS

Proses replikasi terdiri dari dua fase:

- Sinkronisasi awal — Salinan lengkap data server sumber ke AWS. Data disimpan sebagai snapshot Amazon Elastic Block Store (Amazon EBS) atau di Amazon fsX untuk volume ONTAP (FSx NetApp untuk ONTAP) di akun target, tergantung pada jenis penyimpanan target yang dikonfigurasi. Untuk informasi selengkapnya, lihat Jenis penyimpanan target di Panduan Pengguna MGN. Durasi tergantung pada volume data dan bandwidth jaringan.
- Replikasi berkelanjutan - Sinkronisasi berkelanjutan dari blok yang diubah dengan dampak minimal pada kinerja server sumber. Mempertahankan salinan terbaru di AWS.

Server replikasi adalah instans Amazon EC2 sementara yang digunakan di subnet area pementasan. Mereka menerima data yang direplikasi dari server sumber dan secara otomatis dikelola oleh MGN. Untuk membaca selengkapnya, lihat Pengaturan server replikasi di Panduan Pengguna MGN.

AWS Transform memantau kemajuan replikasi dan menyediakan pembaruan status, termasuk status replikasi, jeda replikasi (perbedaan waktu antara sumber dan data yang direplikasi), dan penggunaan bandwidth.

Selama replikasi, setiap server berkembang melalui status berikut:

- Belum siap — Server sedang menjalani proses sinkronisasi awal dan belum siap untuk pengujian.
- Siap untuk pengujian - Server telah berhasil ditambahkan dan replikasi data telah dimulai. Contoh pengujian atau cutover sekarang dapat diluncurkan.

Setelah semua server dalam gelombang telah berkembang melampaui `NOT_READY` status, fase replikasi data selesai dan Anda dapat melanjutkan ke pengujian.

Anda dapat mengontrol replikasi untuk server individual atau seluruh gelombang kapan saja:

- Jeda replikasi - Jeda sementara replikasi untuk server tertentu atau seluruh gelombang.
- Lanjutkan replikasi - Lanjutkan replikasi yang dijeda sebelumnya.
- Hentikan replikasi — Hentikan replikasi secara permanen. Replikasi yang dihentikan dapat dimulai ulang, tetapi dimulai dari sinkronisasi awal.

Langkah 5: Pengujian

Setelah replikasi data selesai, Anda dapat meluncurkan instance pengujian untuk memvalidasi server yang dimigrasi sebelum melakukan pemotongan akhir. Untuk membaca selengkapnya, lihat

Meluncurkan instance pengujian di Panduan Pengguna MGN. AWS Transform mendukung dua opsi pengujian:

- Pengujian gelombang penuh - Luncurkan instance pengujian untuk semua server dalam gelombang.
- Pengujian selektif — Luncurkan instance pengujian untuk server tertentu yang Anda pilih dengan memberikan ID yang disediakan pengguna dari file inventaris.

AWS Transform meluncurkan instans Amazon EC2 dari data yang direplikasi dan menyediakan ID instans sehingga Anda dapat terhubung dan memvalidasi instance pengujian. Setelah pengujian, Anda dapat:

- Lanjutkan ke cutover jika pengujian berhasil.
- Luncurkan instance pengujian baru untuk menguji ulang.
- Hentikan instance pengujian dan atasi masalah apa pun sebelum menguji ulang.

Langkah 5b: Tandai aplikasi sebagai siap untuk cutover

Setelah pengujian selesai dan Anda puas dengan hasilnya, tandai aplikasi Anda sebagai siap untuk dipotong. AWS Transform meninjau status replikasi setiap aplikasi dan menyelesaikan setiap peringatan replikasi sebelum memungkinkan Anda untuk melanjutkan. Hanya aplikasi dengan status replikasi bersih yang dapat ditandai untuk cutover.

Langkah 6: Cutover

Cutover adalah langkah migrasi terakhir tempat beban kerja produksi Anda dipindahkan. AWS Untuk membaca selengkapnya, lihat Meluncurkan instance cutover di Panduan Pengguna MGN. Mirip dengan pengujian, AWS Transform mendukung cutover gelombang penuh atau cutover selektif untuk server tertentu.

Selama cutover, AWS Transform meluncurkan instans Amazon EC2 dari data yang direplikasi terbaru dan menyediakan ID instans untuk setiap server. Setelah memverifikasi instance cutover, Anda menyelesaikan cutover, yang menghentikan replikasi mesin sumber yang sedang berlangsung.

Proses cutover meliputi langkah-langkah berikut:

1. Luncurkan instans cutover — AWS Transform meluncurkan instans Amazon EC2 untuk server yang dipilih. Anda dapat memilih cutover gelombang penuh atau cutover selektif.

2. Verifikasi instance cutover — Hubungkan ke instans yang diluncurkan dan verifikasi bahwa instans tersebut berfungsi dengan benar.
3. Selesaikan cutover - Konfirmasikan cutover untuk menghentikan replikasi mesin sumber. Anda dapat menyelesaikan semua server dalam gelombang atau memilih server tertentu. Finalisasi menghentikan agen replikasi mengirim data, menghapus agen replikasi dari server sumber, dan mengunci status siklus hidup server. Tindakan ini tidak dapat dengan mudah dibatalkan. Untuk membaca lebih lanjut tentang hal itu, lihat Menyelesaikan cutover di Panduan Pengguna MGN.
4. Server sumber arsip (opsional) — Setelah finalisasi, Anda dapat menandai server sumber sebagai diarsipkan untuk membebaskan kuota server sumber di akun Anda.

> **Important**
>
> Menyelesaikan cutover menghentikan replikasi mesin sumber yang sedang berlangsung. Pastikan Anda telah memverifikasi instance cutover Anda sebelum menyelesaikan.

> **Note**
>
> Downtime terjadi antara shutdown sumber dan ketersediaan instance cutover. Rencanakan jendela cutover Anda sesuai dengan itu.

Status siklus hidup server

Selama migrasi, setiap server berkembang melalui status siklus hidup berikut. Untuk membaca selengkapnya, lihat Siklus hidup server sumber di Panduan Pengguna MGN.

- Belum siap — Server sedang menjalani proses sinkronisasi awal dan belum siap untuk pengujian.
- Siap untuk pengujian - Replikasi data telah dimulai dan contoh pengujian atau cutover dapat diluncurkan.
- Tes sedang berlangsung - Sebuah contoh pengujian sedang diluncurkan.
- Siap untuk cutover - Server telah diuji dan siap untuk cutover.
- Cutover sedang berlangsung - Instans cutover saat ini sedang diluncurkan.
- Cutover selesai — Server telah dipotong. Semua data telah dimigrasikan ke instance AWS cutover.
- Terputus - Server telah terputus dari MGN.

Anda dapat menanyakan AWS Transform tentang status server Anda kapan saja selama migrasi. AWS Transform menyediakan tabel status gelombang interaktif yang menampilkan semua informasi server yang relevan termasuk siklus hidup migrasi, status replikasi, dan langkah selanjutnya yang direkomendasikan. Anda juga dapat bertanya dalam bahasa alami, misalnya:

- Apa status server saya?
- Bagaimana status gelombang saya?
- Apa status langkah yang saya ikuti saat ini?

Selama migrasi gelombang, Anda dapat meminta AWS Transform untuk memperbarui atau mengubah status server individual. Misalnya, jika 9 dari 10 server dalam gelombang Anda lulus fase pengujian tetapi satu gagal, Anda dapat mengizinkan AWS Transform untuk terus memindahkan 9 server ke fase berikutnya saat menjalankan kembali pengujian di server yang gagal.

Persetujuan penyebaran

Beberapa operasi migrasi memerlukan persetujuan eksplisit sebelum eksekusi. Saat operasi memerlukan persetujuan, AWS Transform merutekan permintaan ke pemberi persetujuan resmi melalui tab Persetujuan. Hanya pengguna dengan peran Admin di AWS Transform yang dapat menyetujui permintaan penerapan. Penerapan dilanjutkan hanya setelah menerima konfirmasi.

Catatan rilis

Catatan rilis berikut mencakup perubahan terbaru keMigrasi (termasuk VMware). Untuk wilayah AWS Transform yang didukung, lihat Wilayah yang Didukung. Untuk wilayah target yang didukung, lihat halaman penyiapan konektor akun.

Juni 2026

- AWS Transform for migrations sekarang mendukung lokalisasi. Anda dapat mengubah bahasa tampilan aplikasi web saat bekerja dengan alur kerja migrasi. Pelajari lebih lanjut tentang setelan bahasa.
- AWS Transform sekarang memungkinkan Anda untuk mengonfigurasi pengaturan replikasi dan pengaturan peluncuran untuk migrasi Anda. Anda dapat mengatur konfigurasi per akun target Anda atau di seluruh server sumber tertentu. Pelajari lebih lanjut tentang pengaturan replikasi dan peluncuran.

- AWS Transform sekarang memungkinkan Anda untuk melampirkan Elastic Network Interfaces (ENIs) yang ada ke template peluncuran Anda. Anda dapat menandai ENI di akun target Anda sehingga tersedia untuk digunakan saat instance diluncurkan. Pelajari lebih lanjut tentang penandaan sumber daya jaringan.
- AWS Transform for migrations sekarang mendukung semua wilayah AWS komersial sebagai target migrasi, tidak termasuk Timur Tengah (Bahrain) dan Timur Tengah (UEA). Pelajari lebih lanjut tentang wilayah target yang didukung.

Mei 2026

- AWS Transform sekarang mendeteksi VPC yang ada di akun target Anda selama peninjauan migrasi jaringan. Anda dapat melihat VPC yang ada di samping VPC yang dipetakan, mengidentifikasi konflik CIDR, dan menyelesaikannya sebelum penerapan. Pelajari lebih lanjut tentang deteksi VPC yang ada.
- AWS Transform mendukung replatforming repositori kode sumber ke kontainer selama migrasi ke. AWSPelajari lebih lanjut tentang kontainerisasi.

April 2026

- Menambahkan pembuatan landing zone secara langsung dalam alur kerja migrasi. AWS Transform memeriksa fondasi yang ada, merekomendasikan struktur Unit Organisasi (OU) dan akun target, dan menawarkan penerapan otomatis atau Infrastructure as Code (IAc) output (CloudFormation, AWS CDK, atau format Landing Zone Accelerator (LZA)). Pelajari lebih lanjut tentang pembuatan landing zone.
- Menambahkan dukungan untuk DHCP dengan pemetaan grup keamanan selama migrasi jaringan. Pelajari lebih lanjut tentang pembuatan grup keamanan.

Maret 2026

- Menambahkan API publik untuk migrasi jaringan, memungkinkan mitra dan akses terprogram ke kemampuan migrasi jaringan. Lihat referensi API migrasi jaringan.

Februari 2026

- Migrasi jaringan sekarang mendukung pengambilan file konfigurasi firewall atau Software-Defined Networking (SDN) tanpa memerlukan data penemuan RVTools. Anda dapat menggunakan firewall atau file konfigurasi SDN secara independen, atau menggabungkannya dengan ekspor RVTools.

Januari 2026

- Menambahkan dukungan untuk penandaan Migration Acceleration Program (MAP) dan penandaan yang ditentukan pengguna pada elemen migrasi jaringan, memungkinkan Anda melacak sumber daya migrasi untuk kredit MAP dan tujuan organisasi.

Desember 2025

- Menambahkan pengalaman migrasi baru dengan alur kerja yang didesain ulang dan meningkatkan kegunaan. Untuk informasi selengkapnya, lihat Mempercepat migrasi VMware: Pengalaman baru AWS Transform.
- Menambahkan konfigurasi jaringan Landing Zone Accelerator (LZA) untuk pengaturan landing zone otomatis.
- Menambahkan dukungan untuk beberapa AWS akun target dalam eksekusi migrasi, memungkinkan Anda untuk memigrasikan beban kerja di seluruh akun.
- Menambahkan dukungan migrasi jaringan untuk format sumber Cisco ACI, Palo Alto FortiGate, dan ModelizeIt.

Oktober 2025

- Eksekusi migrasi diperluas ke 16 wilayah target dengan penambahan Asia Pasifik (Osaka).

September 2025

- Menambahkan dukungan untuk konfigurasi lisensi dalam gelombang rehost, memungkinkan Anda untuk menentukan Bring Your Own License (BYOL) atau opsi yang disertakan lisensi per gelombang.

- Menambahkan dukungan untuk Terraform dalam migrasi jaringan, memungkinkan Anda menerapkan infrastruktur jaringan menggunakan templat Terraform.
- Menambahkan dukungan untuk memperbarui konfigurasi subnet replikasi, menghapus persyaratan untuk VPC default di akun target.

Agustus 2025

- Eksekusi migrasi yang diperluas (migrasi jaringan dan rehost) ke 15 wilayah target dengan penambahan AS Timur (Ohio), Eropa (Stockholm), dan Eropa (Irlandia).
- Menambahkan kemampuan jangkauan CIDR yang fleksibel untuk migrasi jaringan, memungkinkan Anda untuk menyesuaikan rentang alamat IP selama penyebaran jaringan.

Mei 2025

- Peluncuran layanan awalMigrasi (termasuk VMware), menyediakan kemampuan migrasi ujung ke ujung termasuk penemuan, penilaian, perencanaan migrasi, migrasi jaringan, dan rehost server.
- Eksekusi migrasi yang didukung di 12 wilayah target: AS Timur (Virginia N.), AS Barat (Oregon), Kanada (Tengah), Amerika Selatan (Sao Paulo), Eropa (Frankfurt), Eropa (London), Eropa (Paris), Asia Pasifik (Mumbai), Asia Pasifik (Seoul), Asia Pasifik (Tokyo), Asia Pasifik (Singapura), dan Asia Pasifik (Sydney).

Kontainerisasi kode sumber

AWS Transform mendukung replatforming aplikasi ke kontainer selama migrasi ke. AWS Bab ini menjelaskan kemampuan AI agen AWS Transform untuk mengotomatiskan containerisasi kode sumber Anda. Anda dapat melakukan migrasi dan memodernisasi secara paralel, mengurangi waktu dan kerumitan perpindahan dari arsitektur lokal ke arsitektur cloud-native. Anda dapat menyimpan kode sumber dari GitHub, Bitbucket, atau file.zip, menghasilkan gambar Docker GitLab, mempublikasikan ke Amazon Elastic Container Registry (Amazon ECR) Registry (Amazon ECR), dan menyebarkan ke Amazon Elastic Container Service (Amazon ECS) atau Amazon Elastic Kubernetes Service (Amazon EKS). Ini membawa kontainerisasi ke dalam alur kerja yang sama yang Anda gunakan untuk merencanakan dan menjalankan migrasi rehost.

> **Note**
>
> Kontainerisasi kode sumber dirancang untuk aplikasi yang belum dikemas dalam wadah. Ini memerlukan akses ke kode sumber aplikasi Anda dan tidak mendukung migrasi beban kerja kontainer yang ada. Untuk memigrasikan container yang ada AWS, gunakan metode penerapan standar untuk Amazon Elastic Container Service atau Amazon Elastic Kubernetes Service.

Kemampuan dan fitur utama

AWS Transform menawarkan kemampuan berikut untuk mengkontainerisasi aplikasi Anda.

- AI-driven analisis kode sumber — Menganalisis kode sumber aplikasi Anda dan secara otomatis menghasilkan artefak Docker, termasuk Dockerfiles dan file konfigurasi terkait.
- Pembuatan dan penerbitan gambar kontainer — Membuat dan menguji gambar kontainer menggunakan build AI-driven Docker, dan menerbitkannya ke Amazon Elastic Container Registry dengan pemindaian kerentanan otomatis.
- Infrastructure as Code generation — Menghasilkan infrastruktur deployment untuk Amazon Elastic Kubernetes Service (Helm charts) atau Amazon Elastic Container Service (modul Terraform), dengan validasi otomatis dan pemindaian keamanan.
- Dukungan ketergantungan pribadi - Anda dapat secara opsional mengonfigurasi AWS CodeArtifact repositori (Maven, PyPI, npm) dan gambar dasar Amazon ECR pribadi sebagai sumber ketergantungan untuk build Anda.

- Pengujian berulang dan penerapan cutover — Menyebarkan infrastruktur pengujian untuk validasi, lalu menyebarkan infrastruktur yang telah diselesaikan untuk pemotongan produksi.

Cara kerja kontainerisasi

AWS Transform menggunakan AI-powered agen untuk memandu Anda melalui proses kontainerisasi dalam alur kerja berbasis obrolan. AWS Transform mengkoordinasikan tugas-tugas khusus seperti analisis kode sumber, pembuatan gambar Docker, dan pembuatan infrastruktur. Pada poin-poin penting dalam alur kerja, Anda meninjau dan menyetujui output sebelum melanjutkan.

Anda mengakses kontainerisasi kode sumber dengan membuat pekerjaan migrasi VMware. Di dalam pekerjaan, Anda dapat menjalankan containerization sebagai alur kerja mandiri, atau sebagai bagian dari migrasi ujung ke ujung saat strategi migrasi wave disetel ke containerize. Untuk informasi selengkapnya tentang migrasi VMware, lihat. Migrasi (termasuk VMware)

Prasyarat

Sebelum Anda mulai, pastikan Anda memiliki yang berikut:

- Ruang kerja AWS Transform. Untuk informasi tentang mendapatkan ruang kerja, lihat Memulai.
- Kode sumber aplikasi Anda dalam repositori Git dapat diakses melalui AWS CodeConnections, atau dikemas sebagai file zip untuk diunggah. File individual tidak boleh melebihi 1 GB, dan ukuran total semua kode sumber tidak boleh melebihi 8 GB.
- (Opsional) Repositori Amazon ECR untuk menerbitkan gambar kontainer. Anda dapat mengonfigurasi akses Amazon ECR nanti di alur kerja saat Anda siap untuk mempublikasikan.
- Untuk penerapan Amazon EKS: Kluster Amazon EKS yang ada, atau izin untuk membuat infrastruktur yang diperlukan.
- Untuk penerapan Amazon ECS: Izin untuk membuat klaster, layanan, dan sumber daya terkait Amazon ECS menggunakan Terraform,, atau. AWS CloudFormation AWS Cloud Development Kit (AWS CDK)

Alur kerja kontainerisasi

Alur kerja kontainerisasi terdiri dari langkah-langkah berikut. Agen AWS Transform memandu Anda melalui setiap langkah di antarmuka obrolan.

1. the section called “Langkah 1: Penafian keamanan”— Tinjau dan terima penafian keamanan.
2. the section called “Langkah 2: Kloning kode sumber”— Berikan kode sumber aplikasi Anda.
3. the section called “Langkah 3: Kontainerisasi”— Agen AI menganalisis kode Anda dan menghasilkan artefak Docker.
4. the section called “Langkah 4: Tinjau artefak”— Tinjau artefak yang dihasilkan dan setujui perubahan kode.
5. the section called “Langkah 5: Publikasikan gambar”— Publikasikan gambar kontainer ke Amazon ECR.
6. the section called “Langkah 6: Hasilkan IAc”— Hasilkan templat penyebaran Amazon EKS atau Amazon ECS.
7. the section called “Langkah 7: Menyebarkan infrastruktur pengujian”— Menyebarkan dan memvalidasi infrastruktur pengujian.
8. the section called “Langkah 8: Bersihkan infrastruktur pengujian”— Meruntuhkan sumber daya uji.
9. the section called “Langkah 9: Menyebarkan infrastruktur cutover”— Menyebarkan infrastruktur produksi.

Memulai pekerjaan kontainerisasi

Untuk mengkontainerisasi aplikasi Anda, pertama-tama Anda membuat pekerjaan migrasi VMware di ruang kerja Transform Anda. AWS Dari dalam pekerjaan, Anda dapat memilih untuk menjalankan alur kerja kontainerisasi mandiri atau alur migrasi ujung ke ujung yang menyertakan kontainerisasi.

- Kontainerisasi mandiri — Kontainerisasi kode sumber Anda tanpa melakukan migrasi VMware penuh. Pilih opsi ini ketika Anda ingin mengkontainerisasi aplikasi secara independen dari migrasi infrastruktur apa pun.
- End-to-end migrasi dengan containerization - Jalankan alur kerja migrasi VMware lengkap dengan strategi migrasi containerize yang ditetapkan ke satu atau beberapa gelombang. Alur kerja kontainerisasi berjalan sebagai bagian dari migrasi untuk gelombang tersebut. Untuk informasi selengkapnya tentang migrasi VMware, lihat. Migrasi (termasuk VMware)

Untuk memulai pekerjaan kontainerisasi

1. Di halaman landing ruang kerja Anda, pilih Buat pekerjaan.
2. Pilih migrasi VMware.

3. Pilih apakah akan menjalankan alur kerja kontainerisasi mandiri atau alur migrasi ujung ke ujung yang menyertakan kontainerisasi.
4. AWS Transform memandu Anda melalui langkah-langkah alur kerja, dimulai denganthe section called “Langkah 1: Penafian keamanan”.

Izin kontainerisasi

Selama alur kerja kontainerisasi, AWS Transform menerapkan peran IAM ke Anda Akun AWS yang digunakan untuk membangun image kontainer dan menyebarkan infrastruktur. Anda diminta untuk meninjau dan menyetujui pembuatan peran ini sebelum alur kerja berlanjut.

Peran diberi nama `AWSTransformCodeBuildExecutionRole` dan digunakan melalui AWS CloudFormation tumpukan. Ini termasuk kebijakan terkelola berikut.

Kebijakan dasar

Memberikan izin inti untuk alur kerja kontainerisasi:

- Amazon S3 - Membaca dan menulis objek dalam ember `aws-transform-*`
- Amazon ECR - Otentikasi, tarik gambar (ditandai dengan`Project: atx-migration`), dan dorong gambar ke repositori yang ditandai `CreatedBy: AWSTransform`
- AWS CodeArtifact — Baca dari repositori yang ditandai dengan`Project: atx-migration`, daftar repositori, dapatkan token otorisasi, dan baca titik akhir repositori. Izin ini mendukung resolusi ketergantungan publik dan sumber ketergantungan pribadi yang dikonfigurasi selama alur kerja
- Amazon CloudWatch Logs - Buat grup log dan aliran, dan tulis peristiwa log untuk dan grup log Amazon ECS
- AWS KMS — Jelaskan dan dekripsi kunci (dicakup ke Amazon S3 melalui kondisi layanan)
- Amazon EC2 - Membuat dan mengelola antarmuka jaringan untuk proyek VPC-enabled
- AWS CodeConnections — Gunakan koneksi untuk mengakses repositori kode sumber melalui ARN yang dikonfigurasi CodeConnections

Kebijakan jaringan

Mengelola sumber daya jaringan untuk aplikasi yang digunakan:

- Elastic Load Balancing - Membuat, mendeskripsikan, dan mengelola penyeimbang beban, grup target, pendengar, dan aturan (ditandai dengan) `CreatedBy: AWSTransform`
- Route 53 - Membuat dan mengelola zona yang dihosting dan catatan DNS
- AWS Cloud Map — Membuat dan mengelola ruang nama dan layanan untuk penemuan layanan (ditandai dengan) `CreatedBy: AWSTransform`

Kebijakan penyimpanan

Mengelola sumber daya penyimpanan untuk aplikasi yang digunakan:

- Amazon S3 — Membuat dan mengelola bucket, termasuk enkripsi, pembuatan versi, siklus hidup, dan kebijakan akses
- Amazon EFS — Membuat dan mengelola sistem file, memasang target, dan titik akses (ditandai dengan`CreatedBy: AWSTransform`)
- Amazon EBS - Membuat dan mengelola volume dan snapshot (ditandai dengan) `CreatedBy: AWSTransform`

AWS KMS kebijakan

Mengelola operasi kunci enkripsi:

- Jelaskan kunci, daftar alias, dan baca kebijakan kunci
- Mengenkripsi, mendekripsi, dan menghasilkan kunci data (tercakup ke Amazon Logs CloudWatch , Amazon EFS, Amazon EC2, dan Amazon S3 melalui kondisi layanan)

Kebijakan Amazon ECS

Mengelola sumber daya Amazon Elastic Container Service untuk penerapan kontainer:

- Membuat dan mengelola cluster, layanan, dan definisi tugas (ditandai dengan) `CreatedBy: AWSTransform`
- Daftarkan dan deregister definisi tugas, jalankan tugas
- Lulus peran IAM ke tugas dan layanan Amazon ECS
- Baca informasi peran IAM dan sertifikat ACM
- Jelaskan VPC Amazon EC2, subnet, grup keamanan, dan antarmuka jaringan

Kebijakan Amazon EKS

Mengelola resource Amazon Elastic Kubernetes Service untuk penerapan Kubernetes:

- Akses API Kubernetes, jelaskan klaster, dan daftar add-on
- Lulus peran IAM ke Amazon EKS

> **Note**
>
> Semua izin dicakup untuk Anda dan. Akun AWS Wilayah AWS Sumber daya yang dibuat oleh AWS Transform ditandai dengan`CreatedBy: AWSTransform`, dan operasi penulisan dibatasi untuk sumber daya dengan tag ini jika berlaku.

Langkah 1: Tinjau penafian keamanan

Sebelum alur kerja containerization dimulai, AWS Transform menyajikan disclaimer keamanan yang harus Anda tinjau dan terima.

Apa yang dicakup oleh disclaimer

Proses kontainerisasi memerlukan akses internet untuk mengunduh dependensi aplikasi (seperti paket dari pendaftar publik seperti npm, PyPI, atau Maven Central). Penafian keamanan memberi tahu Anda bahwa:

- Lingkungan build akan memiliki akses internet keluar untuk menyelesaikan dan mengunduh dependensi yang ditentukan dalam konfigurasi build aplikasi Anda.
- Anda bertanggung jawab untuk meninjau dependensi yang dibutuhkan aplikasi Anda dan memastikannya memenuhi kebijakan keamanan organisasi Anda.
- Anda dapat secara opsional mengonfigurasi sumber ketergantungan pribadi (seperti AWS CodeArtifact repositori atau gambar dasar Amazon ECR pribadi) di langkah selanjutnya untuk menghindari pengunduhan dari pendaftar publik.

Apa yang perlu Anda lakukan

Untuk meninjau dan menerima penafian keamanan

1. Baca penafian keamanan yang disajikan AWS Transform.
2. Terima penafian untuk melanjutkan alur kerja kontainerisasi.

Setelah Anda menerima disclaimer, AWS Transform melanjutkan ke langkah berikutnya di mana Anda memberikan kode sumber Anda.

> **Note**
>
> Konfigurasi konektor untuk AWS CodeConnections dan Amazon ECR tidak lagi diperlukan di muka. AWS Transform meminta Anda untuk mengonfigurasi akses ke layanan ini nanti dalam alur kerja, hanya bila diperlukan. Misalnya, Anda mengonfigurasi CodeConnections ketika Anda memilih untuk mengkloning dari repositori Git, dan Anda mengonfigurasi akses Amazon ECR saat Anda siap untuk mempublikasikan gambar kontainer.

Langkah 2: Kloning kode sumber

Pada langkah ini, Anda memberikan kode sumber aplikasi Anda ke AWS Transform. Anda dapat memberikan kode sumber dari repositori Git atau dengan mengunggah file zip.

Opsi 1: Repositori Git

Anda dapat memberikan kode sumber Anda dari repositori Git. Jika Anda belum mengkonfigurasi AWS CodeConnections, AWS Transform meminta Anda untuk memberikan CodeConnections ARN Anda pada saat ini. Anda memiliki dua cara untuk menentukan repositori Anda:

- Unggah file CSV — Siapkan file CSV dengan kolom `repo_name` dan. `branch` `repo_name`Harus dalam `owner/repo-name` format. `branch`Kolom ini opsional; biarkan kosong untuk menggunakan cabang default repositori. Contoh:

```
repo_name,branch
owner/my-app,main
owner/api-service,develop
owner/another-repo,
```

- Berikan detail repositori dalam obrolan — Beri tahu AWS Transform nama repositori dan cabang langsung di obrolan alih-alih mengunggah CSV.

AWS Transform memvalidasi bahwa setiap repositori ada dan dapat diakses melalui CodeConnections konfigurasi Anda, lalu mengkloning repositori.

> **Note**
>
> File individual tidak boleh melebihi 1 GB, dan ukuran total semua repositori tidak boleh melebihi 8 GB. AWS Ubah klon beberapa repositori secara bersamaan untuk pemrosesan yang lebih cepat.

> **Important**
>
> Aplikasi yang menjangkau beberapa repositori tidak didukung melalui opsi klon Git. Jika aplikasi Anda memerlukan kode sumber dari beberapa repositori, paketkan mereka bersama-sama sebagai file zip dan gunakan opsi unggah zip sebagai gantinya.

Opsi 2: Unggahan file zip

Jika Anda tidak mengonfigurasi CodeConnections, atau lebih suka mengunggah kode sumber Anda secara langsung, Anda dapat mengunggah file zip.

Untuk mengunggah kode sumber sebagai file zip

1. Saat diminta, unggah satu file zip per aplikasi. Anda dapat menyeret file ke obrolan atau menggunakan tombol lampiran.
2. Setiap file zip harus berisi kode sumber untuk satu aplikasi.
3. File individual tidak boleh melebihi 1 GB, dan ukuran total semua unggahan tidak boleh melebihi 8 GB.

Opsional: Konfigurasikan dependensi pribadi

Jika aplikasi Anda bergantung pada paket dari pendaftar pribadi, Anda dapat mengonfigurasi sumber ketergantungan pribadi selama langkah ini. AWS Transform mendukung jenis ketergantungan pribadi berikut:

- AWS CodeArtifact repositori - Konfigurasikan repositori Maven, PyPi, atau npm yang dihosting di. AWS CodeArtifact AWS Transform menyajikan daftar repositori yang tersedia di akun Anda untuk Anda pilih.
- Gambar dasar ECR Amazon pribadi - Jika file Dockerfile Anda menggunakan gambar dasar dari repositori ECR Amazon pribadi, berikan detail repositori sehingga lingkungan build dapat menarik gambar ini.

Saat Anda mengonfigurasi dependensi pribadi, AWS Transform meminta Anda untuk terhubung ke Akun AWS tempat sumber daya ini dihosting. Konfigurasi konektor ini hanya diperlukan jika Anda menggunakan dependensi pribadi.

Mengonfirmasi preferensi kontainerisasi

Setelah kode sumber Anda tersedia, AWS Transform menanyakan apakah Anda memiliki instruksi khusus tentang bagaimana container harus dilakukan. Anda dapat menentukan secara opsional:

- Apakah akan membuat Dockerfile baru atau menggunakan kembali yang sudah ada.
- Apakah akan memproses layanan tertentu dalam monorepo, atau semua layanan.
- Preferensi lain untuk proses kontainerisasi.

Defaultnya adalah:

1. Gunakan image dasar Amazon Linux 2023 (`public.ecr.aws/amazonlinux/amazonlinux:latest`).
2. Memproses semua aplikasi di repositori.
3. Gunakan kembali Dockerfile yang ada jika ada, dengan sedikit modifikasi jika diperlukan.

Anda dapat mengonfirmasi untuk melanjutkan dengan default, atau memberikan instruksi khusus.

Langkah 3: Kontainerisasi

Pada langkah ini, agen AWS Transform AI menganalisis kode sumber aplikasi Anda dan menghasilkan artefak Docker. AWS Transform memeriksa struktur, dependensi, dan persyaratan runtime aplikasi Anda untuk menghasilkan Dockerfile yang sesuai dan file konfigurasi terkait.

Apa yang terjadi selama kontainerisasi

Agen kontainerisasi melakukan tugas-tugas berikut:

1. Analisis kode sumber — AWS Transform memeriksa file proyek aplikasi Anda, dependensi, konfigurasi build, dan persyaratan runtime. AWS Transform juga menggantikan nilai hardcode (seperti alamat IP database atau nama DNS), mengidentifikasi variabel lingkungan, mendeteksi volume, dan mencoba untuk mengarahkan log ke output standar.
2. Pembuatan Dockerfile - Berdasarkan analisis, AWS Transform menghasilkan Dockerfile yang disesuaikan dengan aplikasi Anda.
3. Container image build and test — AWS Transform membangun image container dan menjalankan pengujian untuk memverifikasi bahwa Dockerfile menghasilkan image yang berfungsi. Jika build gagal, AWS Transform iterasi pada Dockerfile secara otomatis.
4. Analisis keamanan — AWS Transform memindai artefak yang dihasilkan untuk nilai hardcode seperti kredensyal, kunci API, atau data sensitif lainnya, dan menandai temuan apa pun untuk tinjauan Anda.

Jika Anda menyediakan beberapa repositori atau monorepo dengan beberapa layanan, proses setiap layanan secara paralel.

Jenis aplikasi yang didukung

AWS Transform mendukung containerizing jenis aplikasi berikut:

- .NET 7 dan yang lebih baru - Kontainerisasi yang dioptimalkan untuk aplikasi.NET yang berjalan pada versi 7 atau yang lebih baru.
- Aplikasi umum — Kontainerisasi untuk aplikasi yang dibangun dengan bahasa atau kerangka kerja apa pun, termasuk Java, Python,, Go Node.js, dan lainnya.
- Aplikasi yang tidak didukung - Windows, Xcode, atau lingkungan build berpemilik apa pun tidak didukung.

Apa yang perlu Anda lakukan

Langkah ini berjalan secara otomatis. AWS Transform menampilkan pembaruan kemajuan saat menganalisis dan mengkontainer setiap aplikasi. Jika AWS Transform mengalami masalah, Transform mungkin meminta Anda untuk klarifikasi atau informasi tambahan.

Saat kontainerisasi selesai, AWS Transform bergerak ke langkah berikutnya di mana Anda meninjau artefak yang dihasilkan. Output disimpan di Toko Artifact.

Langkah 4: Tinjau artefak Docker dan perubahan kode

Pada langkah ini, Anda meninjau artefak Docker dan perubahan kode yang dihasilkan AWS Transform selama containerization.

Meninjau perubahan dari repositori Git

Jika Anda memberikan kode sumber dari repositori Git, AWS Transform melakukan perubahan kontainerisasi ke cabang baru di setiap repositori. Jika Anda memiliki beberapa repositori, AWS Transform menyajikan semua perubahan sebagai batch untuk tinjauan konsolidasi.

Untuk meninjau dan menyetujui perubahan

1. Tinjau tautan diff kode yang disediakan AWS Transform. Setiap diff menampilkan semua file yang ditambahkan atau dimodifikasi, termasuk Dockerfiles, file konfigurasi, dan setiap perubahan kode sumber.
2. Saat AWS Transform menyajikan dialog persetujuan, setujui atau tolak perubahan. Untuk beberapa repositori, Anda menyetujui semua perubahan dalam satu operasi batch.
3. Jika Anda menyetujui, AWS Transform mendorong perubahan ke cabang baru di repositori Anda.
4. Konfirmasikan bahwa Anda siap untuk melanjutkan ke langkah berikutnya.

Meninjau perubahan dari unggahan zip

Jika Anda mengunggah kode sumber sebagai file zip, AWS Transform menyediakan tautan unduhan untuk artefak kontainer. Unduh dan tinjau artefak, yang mencakup Dockerfiles yang dihasilkan dan modifikasi kode sumber apa pun.

Meminta perubahan

Jika artefak yang dihasilkan memerlukan modifikasi, Anda dapat memberikan umpan balik ke AWS Transform memulai ulang langkah kontainerisasi dengan instruksi yang diperbarui dan menghasilkan artefak baru untuk tinjauan Anda.

Langkah 5: Publikasikan gambar

Pada langkah ini, AWS Transform menerbitkan gambar kontainer Anda ke Amazon Elastic Container Registry. Jika Anda belum mengonfigurasi akses Amazon ECR, AWS Transform meminta Anda untuk memberikan detail repositori Amazon ECR Anda pada saat ini.

Apa yang terjadi selama penerbitan

AWS Transform melakukan tugas-tugas berikut untuk setiap gambar kontainer:

1. Konfigurasi konektor - Jika Anda belum mengonfigurasi akses ECR Amazon, AWS Transform menyajikan formulir konektor tempat Anda menyediakan ARN repositori Amazon ECR dan terhubung ke ARN. Akun AWS
2. Permintaan persetujuan — AWS Transform menyajikan daftar gambar yang akan dipublikasikan dan meminta persetujuan Anda. Anda dapat menyetujui atau menolak operasi penerbitan.
3. Image push - Setelah disetujui, AWS Transform menerbitkan gambar kontainer ke repositori Amazon ECR Anda menggunakan. Jika Anda memiliki beberapa layanan, AWS Transform menerbitkan semua gambar dalam satu operasi batch.
4. Pemindaian kerentanan - Amazon ECR secara otomatis memindai gambar yang dipublikasikan untuk mengetahui kerentanan yang diketahui. AWS Transform melaporkan hasil pemindaian. Untuk analisis yang lebih komprehensif, aktifkan pemindaian AWS ECR yang ditingkatkan dan Pemantauan AWS GuardDuty Runtime.

> **Note**
>
> Penerbitan gambar berjalan secara asinkron menggunakan. Jika sesi Anda terputus selama penerbitan, AWS Transform secara otomatis memulihkan operasi saat Anda menyambung kembali.

Apa yang perlu Anda lakukan

- Tinjau daftar gambar yang akan dipublikasikan.
- Menyetujui atau menolak operasi penerbitan. Jika Anda menolak, AWS Transform melewatkan penerbitan gambar dan Anda dapat melanjutkan ke langkah berikutnya tanpa gambar yang dipublikasikan.
- Tinjau hasil pemindaian kerentanan setelah penerbitan selesai.

Setelah penerbitan, AWS Transform menyediakan URI gambar yang digunakan dalam langkah-langkah penerapan infrastruktur.

Langkah 6: Hasilkan Infrastruktur sebagai Kode

Pada langkah ini, AWS Transform menghasilkan templat AWS infrastruktur untuk menerapkan aplikasi kontainer Anda. Anda memilih antara Amazon Elastic Kubernetes Service dan Amazon Elastic Container Service sebagai platform target Anda.

Layanan Amazon Elastic Kubernetes (Kubernetes)

Jika Anda memilih Amazon EKS, AWS Transform menghasilkan bagan Helm dengan manifes Kubernetes untuk aplikasi Anda. Template yang dihasilkan meliputi:

- Penerapan dan layanan
- Aturan masuk
- ConfigMaps dan SecretProviderClass sumber daya
- Akun layanan
- Klaim volume persisten (EBS dan EFS)
- Route 53 catatan DNS
- Konfigurasi pemeriksaan kesehatan

Opsional: CloudWatch pencatatan

AWS Transform menanyakan apakah Anda ingin menyertakan CloudWatch logging untuk penerapan Amazon EKS Anda. Jika Anda ikut serta, AWS Transform akan menghasilkan DaemonSet konfigurasi Bit Lancar yang mengumpulkan log pod dan mengirimkannya ke CloudWatch Log.

Jika Anda memilih CloudWatch logging, administrator klaster Anda harus menyelesaikan prasyarat berikut sebelum penerapan:

1. Buat `amazon-cloudwatch` namespace di cluster.
2. Buat peran IAM untuk akun layanan Fluent Bit dengan izin menulis CloudWatch Log. Anda diminta untuk peran ARN selama penerapan.

Layanan Kontainer Elastis Amazon (Terraform)

Jika Anda memilih Amazon ECS, AWS Transform menghasilkan modul Terraform untuk menerapkan aplikasi Anda. Template yang dihasilkan meliputi:

- Kluster ECS
- Layanan ECS dengan definisi tugas
- Application Load Balancer dengan akses logging
- Amazon EFS untuk penyimpanan persisten
- Zona host pribadi untuk penemuan layanan
- Konfigurasi pemeriksaan kesehatan

Validasi otomatis dan pemindaian keamanan

AWS Transform secara otomatis memvalidasi template infrastruktur yang dihasilkan sebelum menyajikannya kepada Anda. Validasi meliputi:

- Untuk Terraform: `terraform fmt` dan `terraform validate` untuk memeriksa sintaks dan kebenaran konfigurasi.
- Untuk Helm: Validasi bagan untuk memverifikasi bahwa manifes yang dihasilkan terbentuk dengan baik.
- Pemindaian keamanan menggunakan Checkov untuk mengidentifikasi kemungkinan kesalahan konfigurasi keamanan dalam kode infrastruktur yang dihasilkan.

AWS Transform melaporkan masalah validasi apa pun dan menyelesaikannya sebelum melanjutkan.

Menghasilkan IAc tanpa gambar yang dipublikasikan

Anda dapat membuat template infrastruktur bahkan jika Anda tidak mempublikasikan gambar kontainer pada langkah sebelumnya. Template yang dihasilkan menggunakan URI gambar placeholder yang dapat Anda perbarui nanti dengan lokasi gambar Anda yang sebenarnya.

Apa yang perlu Anda lakukan

Untuk menghasilkan templat infrastruktur

1. Saat diminta, pilih platform target Anda: Amazon EKS atau Amazon ECS.
2. Jika Anda memilih Amazon EKS, AWS Transform menanyakan apakah CloudWatch akan menyertakan logging. Tanggapi ya atau tidak.
3. Jika diminta, berikan detail titik akhir pemeriksaan kesehatan untuk aplikasi Anda (seperti `/health` jalur).
4. AWS Transform menghasilkan dan memvalidasi template infrastruktur. Langkah ini berjalan secara otomatis dan AWS Transform menampilkan pembaruan kemajuan, termasuk validasi atau hasil pemindaian keamanan apa pun.
5. Tinjau template yang dihasilkan saat AWS Transform menyajikannya.

Langkah 7: Menyebarkan infrastruktur pengujian

Pada langkah ini, AWS Transform menerapkan infrastruktur pengujian sehingga Anda dapat memvalidasi aplikasi kontainer Anda sebelum pemotongan produksi.

Apa yang terjadi selama penerapan pengujian

AWS Transform menyebarkan template infrastruktur yang dihasilkan pada langkah sebelumnya ke Anda Akun AWS. Untuk penerapan Amazon EKS, ini menggunakan Helm. Untuk penerapan Amazon ECS, ini menggunakan Terraform.

Sebelum menerapkan, AWS Transform menyajikan formulir nilai infrastruktur tempat Anda dapat mengonfigurasi parameter penerapan seperti nama sumber daya, jumlah replika, dan pengaturan khusus lingkungan. AWS Transform juga menghasilkan pratinjau penerapan yang menunjukkan sumber daya apa yang akan dibuat, sehingga Anda dapat meninjau perubahan sebelum menyetujui.

Apa yang perlu Anda lakukan

Untuk menyebarkan dan memvalidasi infrastruktur pengujian

1. Lengkapi formulir nilai infrastruktur dengan konfigurasi penerapan Anda. Formulir ini mencakup parameter seperti nama cluster, ruang nama, jumlah replika, dan pengaturan khusus lingkungan lainnya.
2. Tinjau pratinjau penerapan yang menunjukkan sumber daya yang akan dibuat atau dimodifikasi.
3. Menyetujui penerapan infrastruktur saat AWS Transform memintanya.
4. Tunggu hingga penerapan selesai. AWS Transform menampilkan hasil penerapan.
5. Uji aplikasi yang Anda gunakan untuk memverifikasi bahwa aplikasi tersebut berjalan dengan benar dalam wadah dan infrastruktur memenuhi persyaratan Anda.
6. Beri tahu AWS Transform apakah Anda puas dengan penerapan, atau apakah Anda ingin memodifikasi infrastruktur dan menerapkan ulang.

Iterasi pada infrastruktur

Jika Anda perlu memodifikasi templat infrastruktur, Anda dapat mengunggah templat yang diperbarui dan menerapkan ulang:

1. Beri tahu AWS Transform Anda ingin memodifikasi IAc.
2. Unggah file zip yang berisi templat infrastruktur Anda yang dimodifikasi.
3. AWS Transform menggantikan template dan melakukan pembaruan tambahan (Terraform apply atau Helm upgrade).
4. Tinjau hasil penerapan yang diperbarui.

Anda dapat mengulangi siklus ini sampai Anda puas dengan penerapan.

Pemulihan sesi

Penerapan infrastruktur berjalan secara asinkron menggunakan. Jika sesi Anda terputus selama penerapan, AWS Transform secara otomatis memulihkan operasi saat Anda menyambung kembali dan menampilkan hasil penerapan.

Langkah 8: Bersihkan infrastruktur pengujian

Setelah Anda menyelesaikan validasi, AWS Transform meruntuhkan infrastruktur pengujian untuk menghindari biaya yang tidak perlu.

Apa yang perlu Anda lakukan

Untuk membersihkan infrastruktur pengujian

1. AWS Transform memberi tahu Anda bahwa infrastruktur pengujian akan dirobohkan.
2. Menyetujui operasi penghancuran saat AWS Transform memintanya.
3. Tunggu sampai pembersihan selesai. AWS Transform menampilkan hasil ketika semua sumber daya pengujian telah dihapus.

Untuk penerapan Amazon ECS, langkah ini menghapus semua sumber daya yang dibuat selama penerapan pengujian, termasuk sumber daya komputasi, penyeimbang beban, penyimpanan, dan komponen jaringan.

Untuk penerapan Amazon EKS, pembersihan dicakup berdasarkan label ke proyek spesifik Anda. Hanya sumber daya yang ditandai untuk alur kerja kontainerisasi ini yang dihapus. Beban kerja lain yang berjalan pada cluster Amazon EKS yang sama tidak terpengaruh.

Langkah 9: Menyebarkan infrastruktur cutover

Pada langkah terakhir ini, AWS Transform menyebarkan infrastruktur akhir untuk penggunaan produksi. Penerapan cutover menggunakan templat infrastruktur yang sama dengan yang Anda validasi selama langkah penerapan pengujian.

Apa yang perlu Anda lakukan

Untuk menyebarkan infrastruktur cutover

1. Lengkapi formulir nilai infrastruktur dengan konfigurasi penerapan produksi Anda. Anda dapat menyesuaikan parameter dari penerapan pengujian atau menggunakan nilai yang sama.
2. Tinjau pratinjau penerapan yang menunjukkan sumber daya produksi yang akan dibuat.
3. Menyetujui penerapan produksi saat AWS Transform memintanya.
4. Tunggu hingga penerapan selesai. AWS Transform menampilkan hasil penerapan cutover.

5. Verifikasi bahwa aplikasi Anda berjalan dengan benar di lingkungan produksi.

> **Important**
>
> Tinjau konfigurasi infrastruktur yang dihasilkan dengan tim Anda untuk memastikan bahwa mereka memenuhi persyaratan keamanan, kepatuhan, dan bisnis organisasi Anda. Meskipun AWS Transform menyediakan rekomendasi konfigurasi otomatis, Anda bertanggung jawab untuk memvalidasi dan menyesuaikan pengaturan agar sesuai dengan kebutuhan Anda sebelum melanjutkan dengan penerapan cutover.

Pemulihan sesi

Penerapan produksi berjalan secara asinkron menggunakan. Jika sesi Anda terputus selama penerapan, AWS Transform secara otomatis memulihkan operasi saat Anda menyambung kembali dan menampilkan hasil penerapan.

Modernisasi aplikasi mainframe

AWS Transform dirancang untuk mempercepat modernisasi aplikasi mainframe lama. Ini mengatur analisis basis kode mainframe, menghasilkan dokumentasi, mengekstrak logika bisnis, menguraikan struktur monolitik, mengubah kode warisan, dan mengelola perjalanan keseluruhan dengan input manusia (HITL) bila diperlukan. Kemampuan transformasi AWS Transform untuk memodernisasi dan memigrasi aplikasi mainframe memberdayakan Anda untuk memodernisasi aplikasi mainframe kritis Anda lebih cepat, sambil mempertahankan logika kritis bisnis Anda selama proses transformasi.

Topik

Kemampuan dan fitur utama

AWS Transform menyediakan kemampuan berikut untuk modernisasi mainframe:

- Mendukung modernisasi aplikasi z/OS mainframe yang ditulis dalam COBOL (Common Business-Oriented Language) dan (Programming Language One) dengan transaksi JCL PL/I (Job Control Language), CICS (Customer Information Control System), layar BMS (Basic Mapping Support), database Db2, dan file data VSAM (Virtual Storage Access Method).
- Mendukung refactoring aplikasi mainframe Fujitsu GS21 dengan PSAM (Presentation Service Access Method), set karakter Jepang, dan file data NDB (Network Data Base).
- Melakukan penalaran berbasis tujuan, analisis, dekomposisi, perencanaan, pembuatan dokumentasi, dan refactoring kode.
- Secara otomatis memfaktorkan ulang beban kerja COBOL-based mainframe menjadi aplikasi Java modern yang dioptimalkan untuk cloud.

- Mengatur dan mengintegrasikan secara mulus dengan alat dasar yang mengeksekusi analisis, dokumentasi, dekomposisi, perencanaan, dan refactoring kode.
- Membantu Anda mengatur lingkungan cloud untuk aplikasi mainframe modern dengan menyediakan template Infrastructure as Code (IAc) siap pakai.

High-level penelusuran

Berikut adalah panduan AWS Transform tingkat tinggi untuk memodernisasi dan memigrasi aplikasi mainframe.

1. Mulai obrolan dengan AWS Transform, dan masukkan tujuan.
2. Berdasarkan tujuan Anda, AWS Transform mengusulkan rencana modernisasi —— memecah tujuan tingkat tinggi menjadi langkah-langkah menengah.
3. Bergantung pada tujuan yang Anda berikan, AWS Transform dapat:
   - Menilai dan menata ulang basis kode
   - Buat pekerjaan khusus berdasarkan tujuan

Sepanjang jalan, AWS Transform mungkin meminta informasi dari Anda untuk menjalankan tugas.

> **Note**
>
> Untuk informasi selengkapnya tentang AWS Transform untuk kemampuan refactor mainframe, lihat Membuat proyek dalam dokumentasi AWS Transform for mainframe refactor.

Manusia dalam lingkaran (HITL)

Sepanjang transformasi aplikasi mainframe, Anda dapat memantau kemajuan dan status tugas transformasi melalui pengalaman web AWS Transform.

AWS Transform akan mengumpulkan informasi tambahan dari Anda dalam skenario berikut:

- Ketika informasi tambahan diperlukan untuk menjalankan tugas.
- Ketika persetujuan diperlukan untuk artefak menengah (misalnya, dekomposisi domain atau perencanaan gelombang).

- Ketika masalah muncul, AWS Transform tidak dapat diselesaikan secara otomatis.

Jenis file yang didukung untuk transformasi aplikasi mainframe

Jenis file yang didukung untuk z/OS meliputi:

- Artefak COBOL dan CPY terkait (Copybooks)
- PL/I file sumber
- JCL (Job Control Language) dan JCL Procedure (PROC)
- Definisi Sistem CICS (CSD)
- BMS (Dukungan Pemetaan Dasar)
- Database Db2
- VSAM (Metode Akses Penyimpanan Virtual)
- IMS TM (Manajer Transaksi)

Jenis file yang didukung untuk Fujitsu GS21 meliputi:

- PSAM (Metode Akses Layanan Presentasi)
- ADL (Bahasa Definisi AIM)
- NDB (Basis Data Jaringan)

Untuk informasi selengkapnya tentang Fujitsu GS21, lihat topik ini di panduan AWS Transform for mainframe migrasi:

- GS21
- Tangkap & Putar Ulang - Terminal GS21
- Mainframe, AS400, Buka VMS dan GS21

Wilayah dan kuota yang didukung untuk AWS Ubah mainframe

Untuk daftar Wilayah yang didukung, lihatWilayah yang Didukung untuk AWS Transform.

> **Note**
>
> Data Anda mungkin diproses di Wilayah yang berbeda dari Wilayah tempat Anda menggunakan AWS Transform. Untuk informasi tentang pemrosesan lintas wilayah, lihatthe section called “Cross-region pengolahan”.

Untuk batas kuota, lihatKuota.

High-level ikhtisar perjalanan modernisasi mainframe

Modernisasi aplikasi mainframe biasanya AWS dicapai dalam empat fase: (a) menilai, (b) memobilisasi, (c) bermigrasi dan memodernisasi, dan (d) mengoperasikan, mengoptimalkan, dan berinovasi. Masing-masing fase ini dijelaskan lebih lanjut secara rinci dengan tujuan dan kegiatan terkait untuk mendukung perjalanan modernisasi Anda.

Fase

- Fase 1: Menilai
- Fase 2: Memobilisasi
- Fase 3: Migrasi dan modernisasi
- Tahap 4: Mengoperasikan, mengoptimalkan, dan berinovasi

Fase 1: Menilai

Tujuan: Untuk memahami kesiapan modernisasi Anda dan mengembangkan rencana modernisasi awal.

Dengan AWS Transform, Anda dapat mengevaluasi aplikasi, infrastruktur, dan operasi mainframe Anda untuk menentukan pendekatan modernisasi yang tepat untuk kebutuhan bisnis Anda.

> **Note**
>
> Setiap fase berisi daftar semua aktivitas yang mungkin dapat Anda lakukan saat memodernisasi aplikasi mainframe Anda. Anda dapat melewati beberapa aktivitas yang tidak selaras dengan kasus penggunaan atau tujuan Anda.

Aktivitas

- Mulailah dengan percakapan informal tentang modernisasi dan kerjakan kualifikasi peluang dengan mendapatkan orang yang tepat di ruangan
- Tentukan kualifikasi peluang
- Lakukan presentasi panggilan pertama dengan rencana pemetaan ketergantungan aplikasi dan arsitektur aplikasi
- Menentukan ROM (Urutan kasar besarnya) yang mencakup biaya infrastruktur selama migrasi, tenaga kerja migrasi, biaya pasca produksi, durasi proyek
- Deteksi portofolio cepat:
  - Jalankan penilaian panduan pola & alat
  - Lakukan analisis kode sumber
  - Identifikasi kandidat bukti konsep (POC)
- Buat pelatihan pemberdayaan teknis:
  - Hari perendaman
  - Lokakarya dengan topik umum
- Membuat laporan penilaian kesiapan Migrasi (MRA) untuk proyek modernisasi mainframe besar atau pelanggan baru AWS
- Buat pernyataan kerja untuk fase Mobilisasi

Fase 2: Memobilisasi

Tujuan: Untuk membangun fondasi dan memvalidasi pendekatan melalui bukti konsep yang berhasil atau modernisasi percontohan.

Dengan fase mobilisasi, Anda dapat menetapkan kerangka kerja, alat, dan proses yang diperlukan untuk mendukung modernisasi mainframe dalam skala besar.

Aktivitas

- Mulailah rencana modernisasi dengan pilot dan POC yang diidentifikasi:
  - Analisis pilot dan atur kode untuk modernisasi
  - Membangun dan menguji data yang dimodernisasi pilot
  - Memberikan hasil dan meninjau pilot untuk membuat solusi rinci untuk modernisasi data yang lebih besar

- Untuk platform:
  - Buat rencana keamanan, kepatuhan, dan pemantauan
  - Siapkan lingkungan landing zone dengan AWS mainframe landing zone dan struktur akun
  - Menentukan infrastruktur arsitektur, aplikasi, integrasi, dan banyak lagi
- Pikirkan tentang orang dan proses:
  - Cloud Center of Excellence adalah sumber daya yang bagus
  - Model operasi
- Buat analisis portofolio komprehensif untuk skenario aplikasi yang berbeda:
  - Penemuan komprehensif yang mencakup dokumentasi tingkat tinggi
  - Rencana dekomposisi (untuk semua aplikasi)
  - Panduan pola dan alat untuk semua aplikasi
  - Munculkan modernisasi awal dan rencana migrasi
- Buat solusi terperinci dengan mempertimbangkan percontohan, platform, orang dan proses, dan analisis portofolio yang komprehensif
- Tentukan kasus bisnis
- Selesaikan pilot yang sukses dengan solusi untuk pernyataan kerja untuk fase berikutnya

Fase 3: Migrasi dan modernisasi

Tujuan: Untuk menyelesaikan modernisasi dan migrasi dalam timeline dan anggaran yang direncanakan.

Selama fase ini, Anda dapat memigrasi dan memodernisasi aplikasi dan data mainframe Anda ke. AWS

Aktivitas

- Alokasikan sumber daya ke kode aplikasi siap produksi
- Meninjau dan menganalisis desain menggunakan dokumentasi yang ada atau dengan mewawancarai pakar materi pelajaran (UKM)
- Tentukan skenario pengujian
- Siapkan lingkungan modernisasi dengan runtime, database, dan perangkat lunak pihak ketiga yang sesuai

- Migrasi dan modernisasi kode sumber dan data:
  - Anda dapat memodernisasi aplikasi Anda secara iteratif berdasarkan domain bisnis dalam basis kode aplikasi
  - Skalakan lingkungan modernisasi sesuai kebutuhan untuk mendukung beban kerja modernisasi Anda
- Membangun pipa CI/CD (integrasi berkelanjutan, pengiriman berkelanjutan)
- Migrasikan beban kerja yang dipilih untuk memulai modernisasi Anda.
- Transfer program aplikasi dan sesuaikan penskalaan lingkungan sesuai kebutuhan:
  - Anda dapat meningkatkan atau menurunkan lingkungan tergantung pada kebutuhan Anda
  - Modernisasi paket kerja dengan menjalankan tes kewarasan kode
  - Sinkronisasi ulang dengan kode sumber seperlunya
- Jalankan pengujian komprehensif dan verifikasi migrasi Anda:
  - Kumpulkan data negara bagian awal
  - Rekam skenario pengujian pada mainframe
  - Lingkungan panggung, aplikasi, dan data
  - Jalankan skenario pengujian di cloud
  - Bandingkan hasil tes sumber dan target
  - Membangun pipeline eksekusi skenario pengujian otomatis
  - Menghasilkan KPI dan laporan
- Validasi data Anda untuk penyiapan produksi
- Validasi kemajuan Anda dengan:
  - Menjalankan integrasi dan pengujian sistem
  - Menjalankan pengujian kinerja
  - Menjalankan pengujian penerimaan pengguna
  - Menjalankan HA/DR pengujian
- Tentukan rencana peluncuran dengan pemilik aplikasi dan buat rencana untuk pelatihan dan transfer pengetahuan
- Pertimbangkan pemotongan dan jalankan pemotongan produksi
- Menetapkan dukungan berkelanjutan untuk kegiatan pasca produksi seperti operasi, pemantauan, perencanaan kemampuan, dan garansi untuk fungsi aplikasi

Tahap 4: Mengoperasikan, mengoptimalkan, dan berinovasi

Tujuan: Untuk mendukung keunggulan operasional dan peningkatan berkelanjutan dari aplikasi modern.

Setelah modernisasi, Anda dapat mengoptimalkan kinerja, keamanan, dan biaya aplikasi Anda sambil memanfaatkan AWS layanan untuk inovasi.

Aktivitas

- Aplikasi monitor untuk:
  - Metrik kinerja
  - Praktik keamanan
  - Pedoman kepatuhan
  - Biaya
- Detail pencatatan
- Mengelola operasi aplikasi:
  - Manajemen infrastruktur
  - Transaksi dan manajemen pekerjaan
  - Dukungan lingkungan runtime
- Optimalisasi dan modernisasi aplikasi
- Pengembangan dan pemeliharaan aplikasi:
  - Pengkodean IDE
  - Build
  - Integrasi
  - Pengujian
  - Deployment

Transformasi aplikasi mainframe

AWS Transform mempercepat transformasi aplikasi modernisasi mainframe Anda. Topik ini menjelaskan kemampuan yang tersedia.

Topik

Prasyarat: Siapkan masukan proyek di S3

AWS Transform mampu menangani basis kode mainframe yang kompleks. Untuk menggunakan basis kode, pastikan Anda memiliki semua aset di lokasi S3 Anda.

Input proyek utama:

- Kode sumber: Anda harus mengunggah file kode sumber mainframe Anda ke S3. Ini termasuk program COBOL, skrip JCL, copybook, dan file sumber lain yang relevan.
- File data: Jika Anda memiliki file VSAM atau file data lain yang digunakan aplikasi mainframe Anda, ini perlu diunggah ke S3.
- File konfigurasi: Sertakan file konfigurasi khusus untuk lingkungan mainframe Anda.

Input proyek lainnya

- Catatan Fasilitas Manajemen Sistem (SMF): Jika berlaku, unggah catatan SMF ke folder baru di bucket S3 tempat kode sumber disimpan, catatan ini harus dalam format file.zip.
- Pemformatan: Untuk pembuatan dokumentasi teknis, Anda dapat memanfaatkan file konfigurasi opsional untuk menghasilkan dokumen PDF yang selaras dengan format dan standar yang diperlukan, termasuk header, footer, logo, dan informasi yang disesuaikan.
- Glosarium: AWS Transform AWS Transform memanfaatkan otomatisasi dengan Generative AI untuk pembuatan dokumentasi dan ekstraksi aturan bisnis. Menyertakan file CSV glosarium dengan informasi tentang singkatan dan terminologi penting di direktori root file zip Anda akan membantu meningkatkan kualitas dokumentasi yang dihasilkan.

- Data uji: Jika tersedia, unggah kumpulan data pengujian yang dapat digunakan untuk memvalidasi aplikasi yang dimodernisasi. Data ini harus disimpan dalam folder baru di bucket S3 tempat kode sumber disimpan.

Detail tentang input proyek dapat ditemukan di sini.

Koleksi artefak sumber mainframe

Topik ini menguraikan proses pengumpulan kode sumber mainframe dan artefak terkait untuk mendukung inisiatif modernisasi mainframe menggunakan Transform. AWS Pengumpulan dan persiapan artefak sumber yang tepat sangat penting untuk keberhasilan analisis, perencanaan migrasi, dan implementasi.

Inventaris kode sumber

Jenis kode sumber aplikasi

Jenis kode sumber aplikasi berikut didukung:

- Program COBOL
- Skrip JCL
- Program assembler
- PL/I program
- Transaksi CICS
- Program alami
- Skrip REXX
- Program Easytrieve
- Copybook dan termasuk
- Perpustakaan proc

Aset basis data

Aset database berikut didukung:

- Skrip DB2 DCLGEN dan DDL
- Definisi database IMS (PSB dan DBD)
- Definisi file VSAM

- Skema database fisik dan logis
- Prosedur dan pemicu basis data

Ekstensi file kode sumber

Jika Anda tidak yakin dengan ekstensi, biarkan kosong atau .txt dan AWS Transform mengklasifikasikannya untuk Anda.

Tabel berikut mencantumkan ekstensi file umum untuk artefak sumber mainframe.

Ekstensi file kode sumber mainframe

Language/Type	Ekstensi umum	Deskripsi
COBOL	.cbl, .cob, .cobol	Program sumber COBOL
JCL	.jcl	Skrip Bahasa Kontrol Pekerjaan
Perakit	.asm	Program bahasa Assembler
PL/I	.pl1	PL/I program bahasa
Definisi CICS	.csd	Definisi sistem CICS (CSD)
Alami	.nat	Program alami
REXX	.rex, .rexx	Skrip REXX
Easytrieve	.ezt	Program laporan Easytrieve
Copybook	.cpy	Copybook COBOL
BMS	.bms	Dukungan Pemetaan Dasar (BMS)
PL/I copybook	.pl1_salinan	PL/I copybook
Db2 DCLGEN	.dcl	Generator deklarasi Db2
Definisi Db2	.sql, .ddl	Definisi database Db2
Definisi IMS	.ims	Kumpulan data definisi sumber daya IMS (IMS Tahap 1)

Language/Type	Ekstensi umum	Deskripsi
MFS	.mfs	IMS MFS (Layanan Format Pesan)
PSB	.psb	IMS PSB (Blok Spesifikasi Program)
DBD	.dbd	Definisi Basis Data IMS
JCL Proc	.prc, .proc	Pustaka Prosedur JCL
JCL Termasuk	.inc	JCL Sertakan file
Makro	.mac	Makro assembler
Kartu Kontrol	.ctl	Kartu kontrol Utilitas JCL

Metode pengumpulan

Ekstraksi kode sumber

Gunakan klien FTP aman (WinSCP/FileZilla) untuk ekstraksi kode sumber interaktif dari pustaka PDS mainframe.

Ekstraksi skema DB2

Perangkat lunak klien DB2 LUW (Linux, UNIX, dan Windows) mencakup `db2look` utilitas sebagai bagian dari paket instalasinya. `db2look`Utilitas mengekstrak pernyataan DDL (Data Definition Language) dari database DB2.

Jalankan perintah berikut:

```
db2look -d DATABASE_NAME -i userid -w password -a -e -m -l -x -f -createdb -printdbcfg
 -o DATABASE_NAME.sql
```

Daftar berikut menjelaskan parameter:

- `-d` *`DATABASE_NAME`*: Menentukan nama database untuk mengekstrak skema dari
- `-i` *`userid`*: User ID untuk otentikasi database
- `-w` *`password`*: Kata sandi untuk otentikasi basis data
- `-a`: Mengekstrak informasi otorisasi (hibah)

- `-e`: Ekstrak semua objek database (tabel, tampilan, indeks, dan sebagainya)
- `-m`: Ekstrak statistik dan karakteristik fisik
- `-l`: Ekstrak tabel dan kolom komentar
- `-x`: Ekstrak menjelaskan tabel
- `-f`: Memformat output untuk keterbacaan yang lebih baik
- `-createdb`: Termasuk pernyataan pembuatan database
- `-printdbcfg`: Termasuk parameter konfigurasi database
- `-o` *`DATABASE_NAME`*`.sql`: Menentukan nama file output

Mengekstrak informasi katalog dari mainframe

Perintah LISTCAT di IDCAMS adalah utilitas untuk mengambil informasi katalog tentang kumpulan data pada mainframe. Ini berguna selama proyek modernisasi mainframe untuk menginventarisasi kumpulan data dan memahami karakteristiknya.

```
//LSTCATJ JOB 'LISTCAT',CLASS=A,MSGCLASS=X,NOTIFY=&SYSUID
//*
//STEP1 EXEC PGM=IDCAMS
//*
//SYSPRINT DD DSN=AWS.M2.CATALOG.LIST,
//            DISP=(NEW,CATLG,DELETE),
//            SPACE=(CYL,(1,1),RLSE),
//            DCB=(RECFM=FBA,LRECL=133,BLKSIZE=0)
//*
//SYSIN DD *
  LISTCAT ENT('AWS.M2.CARDDEMO.*') ALL
  LISTCAT ENT('SYS1.*') ALL
/*
```

Mengekstrak definisi CICS

Utilitas IBM-provided DFHCSDUP dapat mengekstrak definisi sumber daya CICS dari file CICS System Definition (CSD).

Gunakan JCL berikut untuk membongkar definisi yang ada dari DFHCSD:

```
//CSDEXTJ JOB 'EXTRACT CSD',CLASS=A,MSGCLASS=X,NOTIFY=&SYSUID
//*
//STEP1 EXEC PGM=DFHCSDUP,REGION=0M,
```

```
//          PARM='CSD(READONLY),PAGESIZE(60),NOCOMPAT'
//*
//STEPLIB  DD DSN=your-hlq.SDFHLOAD,DISP=SHR
//DFHCSD   DD DSN=your-hlq.DFHCSD,DISP=SHR
//*
//CBDOUT   DD DSN=your-hlq.CSD,
//            DISP=(NEW,CATLG,DELETE),
//            SPACE=(CYL,(1,1),RLSE),
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=0)
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
  EXTRACT GROUP(cics-group) USERPROGRAM(DFH0CBDC) OBJECTS
/*
```

Mengekstrak definisi IMS

Untuk mengekstrak definisi IMS, Anda dapat menghasilkan file Transaksi IMS dengan menggunakan utilitas DFSURDD0. Utilitas ini membuat pernyataan makro tahap 1 yang dapat Anda gunakan untuk tujuan migrasi. Hanya IMS Tahap 1 yang saat ini didukung.

Gunakan JCL berikut untuk mengekstrak definisi IMS:

```
//IMSEXTJ JOB 'IMS STAGE 1',CLASS=A,MSGCLASS=X,NOTIFY=&SYSUID
//*
//STEP1 EXEC PGM=DFSURDD0,MEMLIMIT=12G
//*
//STEPLIB  DD DISP=SHR,DSN=your-hlq.SDFSRESL
//*
//RDDSDSN DD DISP=SHR,DSN=your-rdds
//*
//SYSOUT   DD DSN=your-hlq.STAGE1.IMS,
//            DISP=(NEW,CATLG,DELETE),
//            SPACE=(CYL,(1,1),RLSE),
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=0)
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
  OUTPUT=MAC
/*
```

Mengekstrak informasi CA-7 penjadwal

CA-7 Informasi jadwal kerja terdiri dari laporan LJOB. Gunakan utilitas LJOB dengan `LIST=NODD` parameter untuk menghasilkan laporan pada mainframe. Laporan harus memiliki `.ca7`

perpanjangan. Setiap laporan harus memiliki karakter kontrol pengangkutan ANSI di kolom 1. Anda dapat membuat laporan dengan karakter kontrol carriage di kolom 1 dengan menentukan `DCB=RECFM=FBA` (atau FA) di JCL yang digunakan untuk menjalankan utilitas LJOB.

Misalnya: `LJOB,JOB=ZBN*,LIST=ALL`

Untuk informasi selengkapnya, lihat dokumentasi Broadcom.

Mengekstrak data SMF

Catatan Fasilitas Manajemen Sistem (SMF) menyediakan metrik aktivitas yang digunakan oleh AWS Transform untuk menganalisis pekerjaan batch dan penggunaan transaksi CICS. Jenis catatan SMF berikut didukung: 14, 15, 30 (sub tipe 5), 64, 102, 110 (sub tipe 1 dengan kelas data 1, 3, dan 4).

Persyaratan format untuk catatan SMF:

- Harus menyertakan jenis rekaman 30 dan 110 (minimal)
- Harus dalam format EBCDIC biner mentah dengan byte RDW (Record Descriptor Word) disertakan
- Harus disediakan sebagai salah satu dari berikut ini:
  - File hingga 600 MB dikompresi dalam format.zip
  - File hingga 5 GB dikompresi dalam format.tar.gz atau.gz
- Harus disimpan dalam folder terpisah dari kode sumber Anda di Amazon S3

Berikan catatan SMF minimal 13 bulan untuk menangkap siklus batch tahunan dan pola musiman yang mungkin terlewatkan oleh kerangka waktu yang lebih pendek.

Opsi 1: Ekstrak dan urutkan catatan SMF

Gunakan contoh JCL berikut untuk mengekstrak dan mengurutkan catatan SMF:

```
//SMFEXTR  JOB (ACCT),'EXTRACT SMF',CLASS=A,
//             MSGCLASS=X,NOTIFY=&SYSUID
//*--------------------------------------------------------------------*
//* SMF RECORD EXTRACTION TEMPLATE
//* CUSTOMIZE: Job card, dataset names, date range, record types
//*--------------------------------------------------------------------*
//*
//* STEP 1: EXTRACT SMF RECORDS
//*
//EXTRACT  EXEC PGM=IFASMFDP
//SYSPRINT DD SYSOUT=*
```

```
//*
//* INPUT: Specify your SMF datasets (MAN files or GDG)
//*
//DUMPIN   DD DISP=SHR,DSN=SYS1.MAN1
//         DD DISP=SHR,DSN=SYS1.MAN2
//         DD DISP=SHR,DSN=SYS1.MAN3
//*
//* OUTPUT: Extracted SMF records
//*
//DUMPOUT  DD DSN=your.hlq.SMF.EXTRACT,
//            DISP=(NEW,CATLG,DELETE),
//            UNIT=SYSDA,
//            SPACE=(CYL,(500,100),RLSE),
//            DCB=(RECFM=VBS,LRECL=32760,BLKSIZE=27998)
//*
//* CONTROL CARDS
//*
//SYSIN    DD *
  INDD(DUMPIN,OPTIONS(DUMP))
  OUTDD(DUMPOUT,TYPE(14,15,30,110))
  DATE(yyyy/ddd,yyyy/ddd)
  START(0000)
  END(2359)
/*
//*
//* STEP 2: SORT EXTRACTED RECORDS (OPTIONAL)
//*
//SORT     EXEC PGM=SORT
//SORTIN   DD DISP=SHR,DSN=your.hlq.SMF.EXTRACT
//SORTOUT  DD DSN=your.hlq.SMF.SORTED,
//            DISP=(NEW,CATLG,DELETE),
//            UNIT=SYSDA,
//            SPACE=(CYL,(500,100),RLSE),
//            DCB=(RECFM=VBS,LRECL=32760)
//SORTWK01 DD UNIT=SYSDA,SPACE=(CYL,(50))
//SORTWK02 DD UNIT=SYSDA,SPACE=(CYL,(50))
//SORTWK03 DD UNIT=SYSDA,SPACE=(CYL,(50))
//SYSOUT   DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *
  SORT FIELDS=(11,4,CH,A,7,4,CH,A,15,4,CH,A),EQUALS
/*
```

Sesuaikan nilai-nilai berikut di JCL sebelumnya:

1. Job card: Update account, class, dan msgclass
2. DUMPIN: Arahkan ke kumpulan data SMF Anda
3. DUMPOUT: Tentukan nama dataset keluaran dan HLQ
4. JENIS: Pilih jenis rekaman (14, 15, 30, 110 atau lainnya)
5. TANGGAL: Format yyyy/ddd (misalnya, 2026/055,2026/056)
6. SPACE: Sesuaikan berdasarkan volume data yang diharapkan
7. Hapus langkah SORT jika tidak diperlukan

Opsi 2: Ekstrak data SMF dengan DFSORT

Anda dapat menggunakan DFSORT untuk mengkonsolidasikan dan melaporkan data SMF historis. Ini berguna jika Anda menyimpan catatan SMF Anda dalam GDG berdasarkan hari atau minggu. Pekerjaan berikut menggunakan DFSORT untuk mengkonsolidasikan catatan SMF dari file SORTIN dan memfilter berdasarkan catatan yang ditentukan dalam parameter INCLUDE COND.

```
//EXTRSMF JOB 'EXTRACT SMF RECORDS',CLASS=S,MSGCLASS=H,
// MSGLEVEL=(1,1),REGION=0M,NOTIFY=&SYSUID,TIME=1440
//SORTJES2 EXEC PGM=SORT,REGION=0M
//SYSOUT DD SYSOUT=*
//SORTMSG DD SYSOUT=*
//SORTIN DD DISP=SHR,DSN=your.hlq.SMF.EXTRACT
//SORTOUT DD DSN=your.hlq.SMF.OUTPUT,
// DISP=(MOD,CATLG,DELETE),
// DCB=(BUFNO=20,
// RECFM=VBS,LRECL=32760,BLKSIZE=27998),
// SPACE=(CYL,(1400,900),RLSE)
//SORTWK1 DD UNIT=SYSDA,SPACE=(CYL,(30,50))
//SORTWK2 DD UNIT=SYSDA,SPACE=(CYL,(30,50))
//SORTWK3 DD UNIT=SYSDA,SPACE=(CYL,(30,50))
//SORTWK4 DD UNIT=SYSDA,SPACE=(CYL,(30,50))
//SYSIN DD *
OPTION COPY
INCLUDE COND=(6,1,BI,EQ,X'0E',OR,6,1,BI,EQ,X'0F',OR,
6,1,BI,EQ,X'1E',OR,6,1,BI,EQ,X'40',OR,
6,1,BI,EQ,X'50',OR,6,1,BI,EQ,X'59',OR,
6,1,BI,EQ,X'5C',OR,6,1,BI,EQ,X'65',OR,
6,1,BI,EQ,X'66',OR,6,1,BI,EQ,X'6E')
```

Untuk mengunduh data sambil menyimpan data RDW, Anda dapat menggunakan server FTP untuk menerima file yang diekstrak. Pastikan data ditransfer dalam format biner dan opsi RDW digunakan sehingga RDW disertakan dalam transfer data. Pekerjaan berikut menunjukkan contoh pekerjaan FTP untuk mentransfer file ini.

```
//EXTRSMF JOB 'EXTRACT SMF RECORDS',CLASS=A,MSGCLASS=H,
// MSGLEVEL=(1,1),REGION=0M,NOTIFY=&SYSUID,TIME=1440
//STEP1 EXEC PGM=FTP,REGION=2048K
//SYSIN DD *
10.10.10.10
userid
password
type i
LOCSITE RDW
put 'your.hlq.SMF.OUTPUT' SMFREQS.SMF
quit
```

Mengunggah laporan SCRT

Anda juga dapat mengunggah Sub-Capacity laporan Reporting Tool (SCRT) ke AWS Transform.

Unggahan inventaris aplikasi

Setelah Anda mengumpulkan semua artefak, selesaikan langkah-langkah berikut:

1. Atur file ke dalam subfolder berdasarkan jenis artefak (misalnya, /cobol, /jcl, /copybooks, /db2, /ims, /cics, /ca7).
2. Kompres folder tingkat atas menjadi satu file.zip.
3. Unggah file.zip ke bucket Amazon S3 yang terhubung ke ruang kerja Transform AWS Anda.

> **Note**
>
> Catatan SMF harus ditempatkan di folder terpisah dari kode sumber Anda dalam bucket Amazon S3, dan harus dalam format.tar.gz atau.gz.

Untuk petunjuk mendetail tentang menghubungkan bucket Amazon S3 Anda ke AWS Transform, lihat. the section called “Siapkan konektor”

Sign-in dan menciptakan pekerjaan

Untuk masuk ke pengalaman web AWS Transform, ikuti semua instruksi di Memulai dengan AWS Transformasi bagian dokumentasi.

Untuk membuat dan memulai pekerjaan: Ikuti langkah-langkah di Mulai proyek Anda.

Buat ruang kerja: Beri nama dan jelaskan ruang kerja Anda tempat pekerjaan, kolaborator, dan artefak terkait akan disimpan.

Buat pekerjaan: Buat pekerjaan dengan memilih dari rencana pekerjaan yang telah dikonfigurasi sebelumnya, atau sesuaikan rencana pekerjaan berdasarkan tujuan Anda dengan memilih dari daftar kemampuan yang didukung.

> **Important**
>
> AWS Transform akan menolak operasi dari Anda jika Anda tidak memiliki izin yang tepat. Misalnya, kontributor tidak dapat membatalkan transformasi pekerjaan aplikasi mainframe atau menghapus pekerjaan. Hanya administrator yang dapat melakukan fungsi-fungsi ini.

Saat Anda membuat pekerjaan, Anda dapat memilih dari kemampuan di bawah ini, tetapi langkah Kickoff selalu diperlukan karena di situlah lokasi kode sumber untuk proyek berada. Pada pekerjaan pertama yang disiapkan di ruang kerja, Anda harus menyiapkan konektor ke bucket Amazon S3 Anda.

Melacak kemajuan transformasi

Anda dapat melacak kemajuan transformasi di seluruh proses dengan dua cara:

- Worklog — Ini menyediakan log terperinci dari tindakan yang dilakukan AWS Transform, bersama dengan permintaan masukan manusia, dan tanggapan Anda terhadap permintaan tersebut.
- Dasbor — Ini memberikan ringkasan tingkat tinggi dari transformasi aplikasi mainframe. Ini menunjukkan metrik pada jumlah pekerjaan yang diubah, transformasi yang diterapkan, dan perkiraan waktu untuk menyelesaikan transformasi aplikasi mainframe. Anda juga dapat melihat detail setiap langkah termasuk, baris kode menurut jenis file, dokumentasi yang dihasilkan oleh setiap jenis file, kode yang didekomposisi, dan rencana migrasi.

Siapkan konektor

AWS Transform menggunakan konektor untuk mengakses sumber daya di akun Anda yang diperlukan untuk fungsi modernisasi mainframe. Konektor Anda secara otomatis dikonfigurasi dengan pekerjaan pertama yang Anda jalankan di ruang kerja. Bergantung pada rencana pekerjaan yang Anda pilih, AWS Transform memandu Anda untuk membuat konektor.

Konektor menata ulang mainframe

Konektor tata ulang mainframe adalah untuk pekerjaan yang menjalankan penilaian dan menata ulang alur kerja. Ini menggunakan bucket S3 untuk mengakses dan menyimpan sumber daya transformasi, serta cluster Amazon Neptunus untuk menyimpan artefak yang diekstraksi. Cluster Neptunus berfungsi sebagai grafik pengetahuan terpadu yang menyimpan semua artefak yang diekstraksi tentang pekerjaan Anda, dan grafik pengetahuan menjawab semua pertanyaan di antarmuka obrolan Transform. AWS

Terapkan cluster Amazon Neptunus di AWS VPC tempat Transform dapat membuat antarmuka jaringan elastis (ENI). Ini adalah koneksi jaringan yang memungkinkan AWS Transform berkomunikasi dengan aman dengan Neptunus di VPC Anda untuk memuat data dan menanyakan grafik pengetahuan Anda.

AWS CloudFormation

Sebaiknya gunakan CloudFormation templat berikut untuk membuat sumber daya yang diperlukan untuk konektor tata ulang mainframe Anda. Gunakan bucket S3 yang sama yang Anda rencanakan untuk digunakan untuk konektor saat menjalankan CloudFormation template. Untuk informasi selengkapnya tentang menggunakan CloudFormation templat, lihat Bekerja dengan templat.

Untuk mengunduh CloudFormation templat, pilih neptune-kg-setup.yaml.

Template menerima parameter berikut:

- BucketName – Wajib. Nama bucket S3 untuk akses pemuat massal Neptunus. Anda harus menggunakan bucket S3 yang sama dengan yang Anda rencanakan untuk digunakan di konektor AWS Transform Anda.
- KmsKeyId – Opsional. Kunci KMS untuk enkripsi Neptunus. Menerima ID kunci, ARN, nama alias, atau alias ARN. Default ke kunci AWS terkelola jika dihilangkan.

Berikan detail berikut saat Anda mengonfigurasi konektor:

- Bucket S3 ARN
- Gugus Neptunus ARN
- ID sumber daya cluster Neptunus
- ID subnet aplikasi
- ID grup keamanan aplikasi
- Peran pemuat massal Neptunus S3 ARN

> **Important**
>
> Tab Output dari CloudFormation template menyertakan detail yang perlu Anda berikan untuk AWS Transform saat mengonfigurasi konektor.

Konfigurasi kustom

Jika Anda lebih suka menggunakan VPC yang ada atau alat lain untuk membuat infrastruktur yang diperlukan, lingkungan Anda harus memenuhi persyaratan berikut:

Gugus Neptunus

- Amazon Neptune Tanpa Server dengan versi mesin 1.4.5.1 atau yang lebih baru
  - Autentikasi IAM diaktifkan
  - Enkripsi penyimpanan diaktifkan (kunci AWS terkelola atau kunci KMS yang dikelola pelanggan)
- Konfigurasi penskalaan tanpa server (disarankan: 1—128 NCU)
- (Disarankan) Baca replika di Availability Zone kedua untuk ketersediaan tinggi dan peningkatan kinerja baca

Jaringan

- VPC dengan dukungan DNS dan nama host DNS diaktifkan
- Setidaknya dua subnet di Availability Zone terpisah. AWS Transform membuat ENI di subnet ini untuk mengakses cluster Neptunus, yang dapat berada di subnet yang sama atau di subnet khusus. Ukuran subnet di/20 untuk mengakomodasi titik akhir VPC dan ENI aplikasi.
- Konektivitas jaringan dari subnet ke AWS layanan berikut melalui titik akhir AWS PrivateLink antarmuka VPC ():

  - AWS Transform Agents API - Mengkoordinasikan pekerjaan modernisasi mainframe
  - Amazon Bedrock Runtime — Memanggil model dasar untuk analisis dan penalaran
  - Amazon Relational Database Service (Amazon RDS) - Manajemen cluster Neptunus
  - Amazon Elastic Compute Cloud (Amazon EC2) — Manajemen ENI
  - Amazon CloudWatch — Metrik untuk observabilitas AWS Transformasi
- Titik akhir gateway S3 yang dilampirkan ke tabel rute yang terkait dengan subnet Anda. AWS Transform menggunakan endpoint ini untuk memuat data dari bucket S3 ke cluster Neptunus.

Grup keamanan

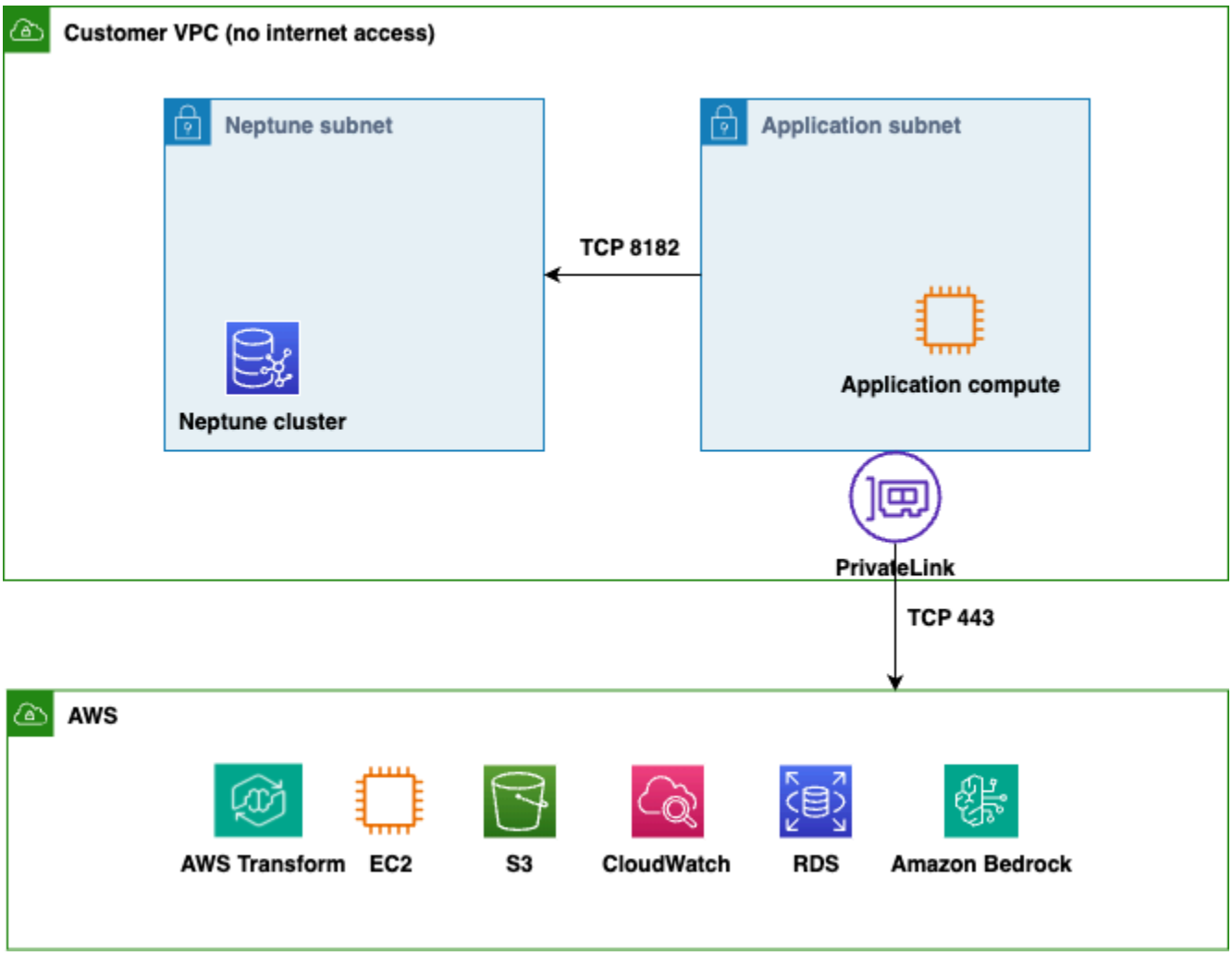
Grup keamanan dilampirkan ke ENI, bukan subnet. Bahkan ketika AWS Transform ENI dan Neptunus berada di subnet yang sama, grup keamanan terpisah mengontrol akses di antara mereka.

- Grup keamanan (dilampirkan ke ENI yang dibuat oleh AWS Transform) yang memungkinkan:
  - Keluar TCP 8182 ke grup keamanan Neptunus
  - Outbound TCP 443 untuk mencapai titik akhir VPC
- Grup keamanan (melekat pada cluster Neptunus) yang memungkinkan:
  - Inbound TCP 8182 dari grup keamanan aplikasi
- Grup keamanan titik akhir VPC (dilampirkan ke ENIS titik akhir antarmuka) yang memungkinkan:
  - TCP 443 masuk dari grup keamanan aplikasi

IAM

- Peran IAM dengan kebijakan kepercayaan untuk hibah `rds.amazonaws.com` itus`3:GetObject`,`s3:ListBucket`, dan `s3:GetBucketLocation` di ember S3 Anda
- Peran ini harus dikaitkan dengan cluster Neptunus untuk memuat data grafik pengetahuan massal

Diagram berikut menunjukkan arsitektur yang direkomendasikan.



Konektor S3

Untuk pekerjaan dengan paket khusus, AWS Transform dapat menggunakan konektor S3. Konektor S3 menggunakan bucket S3 untuk mengakses dan menyimpan sumber daya transformasi, dan bucket vektor S3 untuk output yang diindeks.

> **Important**
>
> Data Anda disimpan dan disimpan di penyimpanan artefak AWS Transform di ruang kerja Anda dan hanya digunakan untuk menjalankan pekerjaan.

Konfigurasi opsional bucket vektor S3.

Di Wilayah di mana bucket vektor S3 tersedia, AWS Transform akan menyimpan pengkodean vektor hasil pekerjaan yang dapat dicari di bucket vektor S3 ini di akun Anda untuk memberikan pengalaman pencarian dan obrolan yang didukung AI. Data tidak digunakan di luar pekerjaan ini, dan tidak digunakan untuk melatih model. Untuk mengaktifkan ini, Anda harus membuat dan menyediakan bucket vektor S3. AWS Transform secara otomatis membuat dan melampirkan peran dengan izin yang diperlukan untuk menulis ke bucket ini.

Izin CORS bucket S3

Setelah konektor Anda dikonfigurasi, tambahkan kebijakan CORS berikut ke bucket S3 agar Anda dapat melihat dan membandingkan artefak secara langsung di konsol Transform. AWS Jika kebijakan ini tidak diatur dengan benar, Anda mungkin tidak dapat menggunakan fungsi tampilan sebaris atau perbandingan file Transform. AWS

```
[
    {
        "AllowedHeaders": [],
        "AllowedMethods": [
            "GET"
        ],
        "AllowedOrigins": [
              "https://*.transform.eu-central-1.on.aws",
              "https://*.transform.ap-south-1.on.aws",
              "https://*.transform.ap-northeast-1.on.aws",
              "https://*.transform.ap-northeast-2.on.aws",
              "https://*.transform.ap-southeast-2.on.aws",
              "https://*.transform.ca-central-1.on.aws",
              "https://*.transform.eu-west-2.on.aws",
              "https://*.transform.us-east-1.on.aws",
              "https://*.transform.sa-east-1.on.aws"
        ],
        "ExposeHeaders": [],
        "MaxAgeSeconds": 0
    }
]
```

Menilai dan menata kembali rencana kerja

Menilai dan menata ulang rencana pekerjaan adalah alur kerja yang telah ditentukan untuk memodernisasi aplikasi mainframe. Ini memandu Anda melalui dua fase: menilai basis kode Anda

untuk mengidentifikasi fungsi bisnis, dan menata ulang fungsi yang Anda pilih. Untuk memilih kemampuan individu sebagai gantinya, gunakan rencana pekerjaan Kustom.

Menilai

Perjalanan modernisasi mainframe biasanya dimulai dengan mengevaluasi basis kode untuk mengidentifikasi konten dan memahami hubungan dan dependensi. Setelah evaluasi ini, Anda dapat mulai menguraikan dan memilih bagian dari kode sumber untuk dimodernisasi. Batas-batas dekomposisi ditentukan berdasarkan fungsi bisnis.

Katalog fungsi bisnis menguraikan basis kode Anda menjadi fungsi bisnis diskrit, yang dibangun di atas dasar jalur data deterministik yang memetakan setiap fungsi bisnis dari awal hingga akhir. Jalur data adalah rute yang diambil data, dimulai dengan pemicu bisnis hingga penulisan akhir dalam kode mainframe. Alih-alih membatasi analisis ke titik masuk tunggal, sistem mencakup pekerjaan batch, transaksi CICS, dan penyimpanan data bersama mereka untuk mengidentifikasi unit kerja bisnis yang koheren.

Fungsi bisnis adalah badan kerja yang dimulai dari pemicu bisnis dan berakhir pada hasil bisnis yang terukur. AWS Transform menyediakan katalog fungsi bisnis yang dapat ditindaklanjuti oleh lini bisnis, memungkinkan keputusan berdasarkan informasi tentang apa yang harus dimodernisasi terlebih dahulu.

Manfaat menilai basis kode Anda untuk memodernisasi menggunakan fungsi bisnis:

- Full-estate visibilitas — Memahami bagaimana pekerjaan batch Anda, transaksi online, dan penyimpanan data terhubung ke fungsi yang berarti untuk bisnis Anda.
- Batasan yang dapat ditindaklanjuti - Setiap fungsi bisnis yang diidentifikasi adalah unit mandiri yang dapat ditinjau dan diprioritaskan oleh pemangku kepentingan lini bisnis Anda untuk upaya modernisasi.
- Konsistensi dari kode ke spesifikasi — Blok kode yang sama yang mendefinisikan fungsi bisnis digunakan untuk menghasilkan spesifikasinya, tanpa lapisan terjemahan dan tidak diperlukan rekonsiliasi.
- Mengurangi waktu penemuan - Deteksi otomatis menggantikan wawancara pengetahuan suku manual selama berbulan-bulan dengan analisis yang sistematis dan berulang.

Hasil penilaian diberikan melalui obrolan

Hasil penilaian yang diberikan melalui obrolan mencakup daftar fungsi bisnis, termasuk jumlah jalur data, apa yang mencakup setiap fungsi dalam hal transaksi batch dan CICS, dan deskripsi

bisnis. Anda diberikan tautan ke artefak yang dihasilkan di ember Amazon S3, dan Anda juga dapat mengaksesnya melalui obrolan. Dua artefak disediakan: rincian fungsi bisnis dan ringkasan fungsi bisnis.

- Ringkasan fungsi bisnis — Menyediakan daftar fungsi bisnis dan deskripsi bahasa alami tentang fungsi apa yang dilakukan elemen-elemen dalam kode.
- Detail fungsi bisnis — Menyediakan grafik interaktif yang menggambarkan hubungan di seluruh kode sumber.

Untuk berinteraksi dengan grafik, pilih elemen untuk menampilkan detail tambahan tentangnya, atau buka elemen untuk menginterogasi detailnya. Dalam setiap halaman, ada ringkasan semua elemen pada tingkat itu dan antarmuka grafis untuk berinteraksi dan mempelajari lebih lanjut tentang setiap komponen, termasuk yang berikut ini:

- Fungsi bisnis — Menggambarkan fungsi bisnis mana yang terhubung satu sama lain dan memberikan gambaran umum dari setiap fungsi bisnis. Ketika Anda memilih fungsi bisnis, informasi berikut disediakan:
  - Jumlah jalur data
  - Jumlah baris kode
  - Deskripsi bisnis: ikhtisar fungsi bisnis yang menggambarkan, dalam bahasa alami, fungsi apa yang dilakukan elemen-elemen dalam kode.
  - Antarmuka: merinci bagaimana elemen data dalam fungsi bisnis berhubungan (misalnya, bertukar atau menulis).
- Detail fungsi bisnis — Menggambarkan koneksi di seluruh jalur data dalam satu fungsi bisnis, memungkinkan Anda untuk menyelam lebih dalam ke komposisi fungsi bisnis. Ketika Anda memilih elemen, Anda melihat bagaimana elemen tersebut terhubung di seluruh fungsi bisnis, bersama dengan rincian berikut:
  - Datastore: penulis dan pembaca untuk datastore.
  - Elemen lain: ikhtisar jalur data terkait yang menggambarkan, dalam bahasa alami, tindakan dan hasil yang ditangani dalam jalur data, daftar program, dan pembaca dan penulis.
- Jalur data — Menggambarkan detail dari satu jalur data dan bagaimana elemen dalam jalur data berhubungan satu sama lain.
  - Deskripsi: ikhtisar jalur data yang menjelaskan, dalam bahasa alami, tindakan dan hasil yang ditangani dalam jalur data.
  - Isi jalur data:

- Titik masuk
- Membaca
- Menulis
- Program

Melalui obrolan, Anda dapat menginterogasi output untuk memahami elemen yang terkandung dalam setiap fungsi bisnis, dan menjelajahi batas-batas melalui grafik fungsi bisnis.

Setelah Anda mengidentifikasi batas-batas modernisasi Anda, obrolan meminta Anda untuk memilih fungsi bisnis yang ingin Anda bayangkan kembali. Anda dapat memilih satu fungsi bisnis atau beberapa fungsi bisnis. AWS Transform mengirimkan batas-batas yang dipilih dan output penilaian yang diperlukan ke aliran penataan ulang untuk melanjutkan perjalanan modernisasi.

Setelah Anda menyelesaikan modernisasi pertama Anda, Anda dapat menggunakan obrolan untuk memilih serangkaian fungsi bisnis tambahan untuk dimodernisasi.

Bayangkan kembali

Pada fase penataan ulang, Anda dapat memilih satu atau semua fungsi bisnis yang diidentifikasi oleh penilaian dan mempersiapkannya untuk modernisasi. Ini memicu ekstraksi logika bisnis diikuti oleh pembuatan persyaratan untuk fungsi bisnis yang dipilih. Persyaratan yang dihasilkan adalah masukan utama untuk rekayasa maju, menerjemahkan pemahaman sistem lama ke dalam spesifikasi yang tepat dari apa yang harus diberikan oleh aplikasi modern.

Ekstrak logika bisnis

Setelah Anda memilih satu atau beberapa fungsi bisnis, AWS Transform mulai menghasilkan logika bisnis untuk fungsi bisnis yang dipilih. Setelah ekstraksi selesai, AWS Transform menyimpan hasilnya dalam bucket Amazon S3 dalam format JSON untuk penggunaan hilir. Anda dapat memantau kemajuan ekstraksi dan meninjau masalah apa pun secara langsung di dalam konsol AWS Transform.

Untuk meninjau hasil ekstraksi logika bisnis

1. Di panel navigasi kiri, pilih langkah Extract business logic untuk mengembangkannya.
2. Lihat yang berikut ini dari detail langkah:

    - S3 bucket link: Lokasi di mana hasil logika bisnis yang diekstraksi disimpan dalam format JSON.

- Masalah: Daftar masalah yang dihadapi selama proses ekstraksi logika bisnis.

Jika ada file yang mengalami masalah selama ekstraksi, detail berikut akan ditampilkan untuk setiap file yang terpengaruh:

- Nama file: Nama file yang mengalami masalah.
- Jenis file: Jenis file (misalnya, COBOL atau JCL).
- Jalur file: Lokasi file dalam aplikasi Anda.
- Status: Status saat ini dari ekstraksi file.
- Detail: Deskripsi masalah yang dihadapi.

> **Note**
>
> Setelah langkah ekstraksi selesai, AWS Transform secara otomatis mulai menghasilkan persyaratan. Anda tidak diminta untuk input tambahan sebelum proses ini dimulai.

Hasilkan persyaratan

Setelah ekstraksi logika bisnis selesai, agen Generate Requirements mengkonsumsi artefak yang diekstraksi (analisis kode, analisis data, aturan bisnis) dan menghasilkan persyaratan modernisasi. Persyaratan ini adalah spesifikasi fungsional formal yang teknologi-agnostik dan mencakup kriteria penerimaan yang dapat diuji. AWS Transform menghasilkan satu `requirements.md` file untuk setiap fungsi bisnis yang Anda pilih dan menyimpan output dalam bucket Amazon S3. Untuk melihat lokasi output Amazon S3, pilih langkah Hasilkan persyaratan.

Setiap `requirements.md` file diatur ke dalam bagian-bagian berikut:

- Judul — Nama fungsi bisnis dan alur kerja yang diwakilinya.
- Prasyarat global — Kondisi yang harus berlaku di seluruh alur kerja sebelum dan selama pemrosesan. Nilai yang dapat dikonfigurasi muncul sebagai placeholder parameter.
- Bagian alur kerja bernomor - Setiap bagian mewakili tahap diskrit dari fungsi bisnis. Setiap bagian berisi yang berikut:
  - Cerita pengguna yang menyatakan tujuan sebagai peran, tindakan yang ingin diambil peran, dan hasil yang dihasilkan.

- Daftar Persyaratan dalam format EARS (Easy Approach to Requirements Syntax), masing-masing dengan pengenal unik.

> **Note**
>
> Memilih rangkaian fungsi bisnis kedua akan memulai kembali langkah penataan ulang untuk fungsi yang baru dipilih. Konsol kemudian menampilkan hasil hanya untuk pilihan baru, sehingga masalah apa pun dengan aturan bisnis dari fungsi bisnis Anda sebelumnya, dan tautan Amazon S3 yang menyimpan output untuk logika dan persyaratan bisnis, tidak lagi ditampilkan. Output Anda sebelumnya tidak hilang. Anda masih dapat mengaksesnya dari tab Artefak atau di bucket Amazon S3 Anda.

Ketertelusuran antara artefak transformasi

AWS Transform mempertahankan ketertelusuran antara artefak yang dihasilkannya di seluruh alur kerja penilaian dan penataan ulang. Ketertelusuran memungkinkan Anda melacak setiap persyaratan ke aturan bisnis yang diekstraksi, atau langsung ke kode sumber asli yang menghasilkan persyaratan.

Anda dapat melihat semua detail keterlacakan untuk setiap fungsi bisnis menggunakan file. `traceability.yaml` Ini menjelaskan bagaimana setiap aturan bisnis memetakan ke suatu persyaratan.

Untuk keterlacakan interaktif, Anda dapat menggunakan plugin IDE di VS Code dan editor yang kompatibel dengan Open VSX. Untuk informasi selengkapnya, lihat Alat pengembang.

Bayangkan kembali rencana kerja

Untuk aplikasi yang telah Anda cakup untuk modernisasi dan siap untuk ditata ulang, Anda dapat menjalankan pekerjaan menata ulang mandiri. Pekerjaan menganalisis kode dan data Anda, mengekstrak logika bisnis, mengidentifikasi domain bisnis, dan menghasilkan persyaratan modernisasi untuk setiap domain bisnis.

Rencana pekerjaan khusus

Rencana pekerjaan khusus memungkinkan Anda membangun alur kerja modernisasi Anda sendiri dengan memilih kemampuan yang ingin Anda jalankan. Ini memberi Anda kontrol penuh atas

kemampuan mana yang disertakan. Pilih dari kemampuan berikut berdasarkan tujuan modernisasi Anda. Beberapa kemampuan tergantung pada yang lain dijalankan terlebih dahulu. Misalnya, analisis kode diperlukan untuk sebagian besar kemampuan lainnya.

Menganalisis kode

Setelah Anda membagikan jalur bucket Amazon S3 dengan AWS Transform, ia akan menganalisis kode untuk setiap file dengan detail seperti nama file, jenis file, baris kode, dan jalurnya.

> **Note**
>
> Anda dapat mengunduh Analisis hasil kode melalui tab artefak di tingkat pekerjaan atau ruang kerja. Pada tingkat pekerjaan, buka opsi 'Artefak' di menu navigasi kiri, dan buka folder 'hasil', atau di tingkat ruang kerja temukan nama pekerjaan, dan buka folder 'hasil'. Ini akan mengunduh file zip yang berisi file klasifikasi untuk alur kerja klasifikasi manual, daftar aset, file JSON dependensi, dan daftar file yang hilang.

Dalam rencana pekerjaan pilih Analisis kode di panel navigasi kiri untuk melihat hasil Anda. Anda dapat melihat hasil analisis kode Anda dengan beberapa cara:

- Tampilan daftar - Semua file di bucket Amazon S3 yang ingin Anda ubah untuk mainframe
- Tampilan jenis file - Semua file di bucket Amazon S3 ditampilkan per jenis file. Untuk daftar jenis file yang didukung, lihat File yang didukung.
- Tampilan folder - Semua file di bucket Amazon S3 ditampilkan dalam struktur folder.

Dalam hasil file, AWS Transform memberikan informasi berikut tergantung pada tampilan file yang Anda pilih:

- Nama
- Tipe file
- Total baris kode
- Jalur berkas
- Baris komentar
- Baris kosong
- Baris kode yang efektif

- Jumlah file
- Kompleksitas Cyclomatic - Kompleksitas siklomatik mewakili jumlah jalur independen linier melalui kode sumber program. AWS Transform akan menunjukkan kompleksitas cyclomatic untuk masing-masing file.

File hilang — File hilang dari analisis kode modernisasi mainframe. File-file ini idealnya, harus ditambahkan sebagai bagian dari input sumber di bucket Amazon S3, dan langkah analisis harus dijalankan kembali untuk hasil yang lebih baik dan kohesif.

Bernama identik — AWS Transform memberi Anda daftar file yang berbagi nama yang sama, dan mungkin karakteristik yang sama (misalnya, jumlah baris kode). Ini tidak akan memiliki kemampuan untuk membandingkan perbedaan antara isi dari dua file sekaligus.

ID Duplikat - Dengan program Cobol, bidang ID Program berfungsi sebagai pengidentifikasi unik file. ID ini harus unik karena digunakan untuk memanggil program di seluruh proyek Anda. Namun, beberapa proyek mungkin memiliki file COBOL dengan nama yang berbeda tetapi ID Program yang sama. Mendapatkan daftar file-file tersebut selama penilaian dapat membantu memahami dependensi di antara semua program.

> **Note**
>
> Ini khusus untuk kode dan file COBOL.

Ketika Anda memiliki program dengan ID duplikat, disarankan untuk mengubah ID Program dari file-file ini untuk memiliki pengenal unik untuk masing-masing dalam kode COBOL. Anda kemudian dapat menjalankan kembali pekerjaan Anda untuk mendapatkan hasil analisis kode yang lebih akurat dan komprehensif.

Dengan menyelesaikan ID Program duplikat, Anda dapat:

- Tingkatkan kejelasan dan pemeliharaan kode
- Mengurangi potensi konflik dalam panggilan program
- Meningkatkan akurasi pemetaan ketergantungan
- Sederhanakan upaya modernisasi masa depan

Masalah basis kode - Potensi masalah yang terdeteksi dalam basis kode yang harus Anda selesaikan sebelum melanjutkan proyek modernisasi. Masalah ini dapat mencakup referensi yang hilang dengan pernyataan terkait atau tautan yang tidak didukung dalam kode.

Klasifikasi pembaruan - Dengan reklasifikasi manual, Anda dapat mengklasifikasikan ulang file menggunakan fitur pembaruan massal dengan mengunggah file JSON dengan klasifikasi baru.

> **Important**
>
> Ini hanya tersedia untuk `UNKNOWN` dan `TXT` file.

Setelah reklasifikasi, Transform akan: AWS

1. Memperbarui hasil klasifikasi
2. Re-runs analisis ketergantungan dengan jenis file baru
3. Menyegarkan semua hasil analisis yang terpengaruh

> **Note**
>
> Anda dapat mengklasifikasikan ulang file hanya setelah loop analisis awal selesai.

Penampil sebaris dan perbandingan file

Penampil Inline adalah fitur dalam AWS Transform untuk kemampuan mainframe yang menyediakan dua kemampuan visualisasi utama:

- Tampilan file: Lihat konten file lama yang dipilih dari pekerjaan
- Perbandingan file: Bandingkan konten dari dua file lama secara berdampingan

Tampilan file masukan

Untuk melihat file Anda di Menganalisis kode langkah

- Di bawah Lihat hasil analisis kode, pilih file menggunakan kotak centang dalam daftar.

  Pilih tombol Lihat tindakan (diaktifkan saat 1 item dipilih).

Konten file akan ditampilkan di layar dalam komponen File View.

Perbandingan file

Untuk membandingkan file di Menganalisis kode langkah

1. Di bawah Lihat hasil analisis kode, pilih dua file menggunakan kotak centang dalam daftar.
2. Pilih tombol Bandingkan tindakan (diaktifkan hanya jika 2 item dipilih).
3. File akan ditampilkan berdampingan dalam komponen perbandingan File.

> **Note**
>
> Anda tidak dapat memilih lebih dari dua file untuk membandingkan file.

> **Important**
>
> Jika Anda mengalami masalah dengan penampil sebaris atau perbandingan file, pastikan bucket S3 sudah diatur dengan benar. Untuk informasi selengkapnya tentang kebijakan CORS bucket S3, lihat. the section called “Izin CORS bucket S3”

Analisis data

AWS Transform menyediakan analisis data untuk membantu memahami dampak hubungan data dan elemen terhadap proyek modernisasi mainframe. Dua output yang disediakan adalah:

- Garis keturunan data: Menelusuri siklus hidup data dengan memetakan hubungan antara sumber data, pekerjaan, dan program
- Kamus data: Berfungsi sebagai repositori yang mendokumentasikan metadata struktural elemen data lama

Setelah menyelesaikan analisis, akan ada dua tab yang ada untuk setiap output, dan kemudian beberapa tampilan tersedia yang dijelaskan di bawah ini untuk garis keturunan data dan kamus.

> **Note**
>
> Saat Anda meminta data anlisis. AWS Transform melakukan analisis kode, yang diperlukan untuk melakukan analisis data.

Silsilah data

AWS Transform menyediakan beberapa tampilan berdasarkan hubungan data yang perlu dipahami di seluruh basis kode yang sedang dimodernisasi. Empat tampilan tabel yang tersedia dalam garis keturunan data meliputi:

- Kumpulan data: Tampilan ini memberikan analisis dampak yang komprehensif, termasuk pelacakan operasi untuk membantu membedakan antara membaca, menulis, memperbarui, dan menghapus
- Tabel Db2: Berikan analisis dampak yang terkait dengan tabel Db2 dan operasi
- Program-to-data: Identifikasi program COBOL mana yang mereferensikan setiap dataset
- JCL-to-data hubungan: Identifikasi skrip JCL mana yang mereferensikan setiap kumpulan data

Ringkasan memberikan gambaran umum tentang sumber data, dan hubungannya dengan program dan JCL, serta informasi ringkasan tentang bagaimana sumber data dimanfaatkan dan total operasi yang ditemukan dalam basis kode.

Kamus data

Memahami elemen data dalam sumber data yang ada dalam basis kode adalah langkah selanjutnya untuk menyadari bagaimana berbagai sumber data bergantung. Kamus data adalah katalog data yang menyediakan metadata tingkat lapangan dengan deskripsi bahasa bisnis untuk pemetaan transformasi yang akurat.

- Struktur data COBOL: Menyediakan informasi lapangan di seluruh copybook COBOL dan referensi INLINE yang ada dalam basis kode, termasuk properti lapangan dan definisi bisnis
- Tabel Db2: Menyediakan properti kolom dan tabel di seluruh tabel Db2 yang ada di basis kode, termasuk kunci primer dan asing, informasi skema dan indeks, dan tipe data

Hubungan antara garis keturunan data dan kamus tersedia dengan memilih sumber data, dan kemudian memanfaatkan garis keturunan data atau tombol kamus data untuk menyelam lebih dalam

tentang hubungan antara sumber data dan elemen data. Navigasi antara garis keturunan data dan kamus, menyediakan visibilitas data terintegrasi dengan garis keturunan data yang menyediakan “apa” - struktur dan makna, di samping “di mana” - penggunaan dan hubungan.

Analisis metrik aktivitas

Analisis metrik aktivitas memungkinkan Anda menganalisis catatan Fasilitas Manajemen Sistem (SMF) tipe 14, 15, 30, 64, 102, dan 110. Analisis ini memberikan wawasan tentang bagaimana elemen digunakan dalam aplikasi mainframe Anda. Ini dapat membantu dengan menghentikan kode yang tidak digunakan atau dengan keputusan tentang arsitektur target untuk aplikasi modern. Jika Anda menyertakan analisis metrik aktivitas dalam pekerjaan Anda, menyelesaikan langkah analisis kode terlebih dahulu memberikan output SMF yang lebih kaya, meskipun tidak diperlukan.

> **Note**
>
> Saat Anda menegosiasikan rencana kerja untuk modernisasi mainframe, menyelesaikan analisis kode dan langkah dekomposisi terlebih dahulu memberikan output yang lebih kaya.

Catatan SMF onboard untuk analisis

Anda harus menyediakan lokasi rekaman SMF dalam bucket Amazon S3 Anda sebagai langkah awal untuk analisis SMF. Berikan catatan minimal 13 bulan untuk menangkap kejadian tahunan yang mungkin terlewatkan oleh kerangka waktu yang lebih pendek. Meskipun 13 bulan direkomendasikan, setiap kerangka waktu dengan catatan SMF yang dapat dideteksi menghasilkan hasil analisis.

Ekstrak SMF Anda harus memenuhi persyaratan format ini:

- Termasuk tipe 14, 15, 30 (sub-tipe 5), 64, 102, dan 110.
- Gunakan file biner mentah dalam format EBCDIC.
- Sertakan byte RDW.

Berikan tautan ke catatan SMF dalam bucket Amazon S3 Anda, memastikan catatan berada dalam folder yang terpisah dari kode sumber Anda. Opsi format termasuk .zip (hingga 600 MB) atau.tar.gz (hingga 5 GB) terkompresi. Jika Anda memberikan tautan ke folder di mana catatan SMF tidak terdeteksi, Anda menerima pesan kesalahan dan tidak ada analisis yang selesai.

Output analisis

Output analisis berisi dua komponen, tergantung pada jenis rekaman: tampilan tabular dan output.csv (tersedia di bucket Amazon S3 Anda). Dalam AWS Transform, output ditampilkan untuk tipe 30 dan 110. Melalui artefak di bucket Amazon S3 Anda, Anda dapat menjelajahi analisis pada jenis catatan lainnya. Di antarmuka pengguna, header menampilkan rentang waktu catatan yang Anda berikan sehingga Anda dapat mengidentifikasi tanggal kunci yang hilang. Misalnya, jika catatan Anda mencakup 1 Mei hingga 31 Oktober tetapi mengecualikan hari yang sibuk setelah Thanksgiving, Anda dapat melihat bahwa catatan kunci tidak ada. Stempel waktu dalam aplikasi web mencerminkan zona waktu sistem Anda dan catatan SMF.

Artefak yang tersedia di Amazon S3 menyediakan analisis untuk semua jenis catatan yang ditemukan dalam catatan yang disediakan.

Tampilan tabel

Tampilan tabular, hanya tersedia untuk SMF 30 dan 110, berisi hingga tiga komponen utama untuk pekerjaan batch dan transaksi CICS:

- Ringkasan — Menyediakan pekerjaan dan transaksi utama.
- Analisis job/CICS transaksi Batch — Menyediakan analisis agregat di seluruh pekerjaan dan transaksi.
- Perbandingan analisis kode (hanya batch) - Menyediakan data perbandingan saat langkah analisis kode berjalan sebelum analisis SMF.

Ringkasan penemuan menyediakan tiga kelompok pekerjaan dan transaksi yang membantu Anda mengidentifikasi item dengan cepat untuk analisis yang lebih dalam.

Pekerjaan Batch dan analisis transaksi CICS menyediakan analisis agregat di seluruh pekerjaan dan transaksi. Beberapa kolom dalam hasil analisis disembunyikan secara default. Pilih ikon roda gigi untuk menampilkan bidang tambahan.

Kunci transaksi menghasilkan agregasi data unik untuk transaksi CICS, menggabungkan empat bidang menjadi satu nilai kunci:

- ID Transaksi
- Nama program
- SysPlex ID

- SysID

Anda dapat mencari dengan kunci lengkap atau dengan komponen kunci apa pun dalam output analisis.

Perbandingan analisis kode menyoroti pekerjaan yang ditemukan di catatan SMF atau langkah kode analisis, tetapi tidak ada di keduanya. Ini menunjukkan pekerjaan yang belum berjalan selama kerangka waktu catatan SMF Anda atau tidak ada dalam langkah kode analisis. Output ini hanya tersedia ketika Anda menjalankan langkah analisis kode sebelum analisis SMF dalam pekerjaan Anda.

Praktik terbaik

Untuk mendapatkan output komprehensif untuk pekerjaan batch (tipe 30), pastikan bahwa nama file JCL Anda cocok dengan nama pekerjaan untuk mendapatkan output yang berarti dalam perbandingan analisis kode.

Hasilkan dokumentasi teknis

Anda dapat menghasilkan dokumentasi teknis untuk aplikasi mainframe Anda yang menjalani modernisasi. Dengan menganalisis kode Anda, AWS Transform dapat secara otomatis membuat dokumentasi terperinci dari program aplikasi Anda, termasuk deskripsi logika program, alur, integrasi, dan dependensi yang ada dalam sistem lama Anda. Kemampuan dokumentasi ini membantu menjembatani kesenjangan pengetahuan, memungkinkan Anda untuk membuat keputusan berdasarkan informasi saat Anda mentransisikan aplikasi Anda ke arsitektur cloud modern.

> **Note**
>
> Saat Anda meminta pembuatan dokumentasi teknis, AWS Transform melakukan analisis kode, termasuk analisis ketergantungan kode, yang diperlukan untuk menghasilkan dokumentasi.

Untuk menghasilkan dokumentasi teknis

1. Di panel navigasi kiri, di bawah Hasilkan dokumentasi teknis, pilih Pilih file dan konfigurasikan pengaturan.
2. Pilih file di bucket Amazon S3 yang ingin Anda buat dokumentasinya, dan konfigurasikan pengaturan di tab Kolaborasi.

> **Note**
>
> File yang dipilih harus memiliki jenis pengkodean yang sama (yaitu, semua dalam CCSID - UTF8 atau ASCII yang sama). Jika tidak, dokumentasi teknis yang dihasilkan mungkin memiliki bidang atau bagian kosong.

3. Pilih tingkat detail dokumentasi:

   - Ringkasan — Memberikan ikhtisar tingkat tinggi dari setiap file dalam ruang lingkup. Juga, berikan ringkasan satu baris dari setiap file.
   - Spesifikasi fungsional terperinci - Memberikan detail komprehensif untuk setiap file dalam lingkup transformasi aplikasi mainframe. Beberapa detail termasuk logika dan aliran, dependensi, pemrosesan input dan output, dan berbagai detail transaksi.

   > **Note**
   >
   > Dokumentasi dapat dibuat hanya untuk file COBOL dan JCL.

4. Pilih Lanjutkan.
5. Setelah AWS Transform menghasilkan dokumentasi, tinjau hasil dokumentasi dengan mengikuti jalur bucket Amazon S3 di konsol, tempat hasilnya dihasilkan dan disimpan.
6. Setelah dokumentasi dibuat, Anda juga dapat menggunakan obrolan AWS Transform untuk mengajukan pertanyaan tentang dokumentasi yang dihasilkan dan memutuskan langkah selanjutnya.

Tambahkan informasi pengguna ke dalam dokumentasi

```
ARTIFACT_ID.zip
### app/
    ### File1.CBL
    ### File2.JCL
    ### subFolder/
    #   # File3.CBL
### glossary.csv
### pdf_config.json
### header-logo.png
### footer-logo.png
```

```
# ...
```

File opsional dapat ditambahkan dalam file zip untuk membantu meningkatkan kualitas dokumentasi yang dihasilkan dan menyediakan halaman sampul PDF yang disesuaikan. Beberapa di antaranya dapat berupa:

- glossary.csv: Anda dapat memilih untuk menyediakan dan mengunggah glosarium opsional di file zip di bucket S3. Glosarium dalam format CSV. Glosarium ini membantu membuat dokumentasi dengan deskripsi yang relevan sesuai dengan kosakata pelanggan. `glossary.csv`File sampel terlihat seperti:

  ```
  LOL,Laugh out loud
  ASAP,As soon as possible
  WIP,Work in progress
  SWOT,"Strengths, Weaknesses, Opportunities and Threats"
  ```

- pdf_config.json: Anda dapat memanfaatkan file konfigurasi opsional ini untuk menghasilkan dokumen PDF yang selaras dengan format dan standar perusahaan mereka, termasuk header, footer, logo, dan informasi yang disesuaikan. Sampel `pdf_config.json` terlihat seperti:

  ```
  {
      "header": {
      "text": "Acme Corporation Documentation",
      "logo": "header-logo.png"
    },
    "customSection": {
      "variables": [
        {
          "key": "business Unit",
          "value": "XYZ"
        },
        {
          "key": "application Name",
          "value": "ABC"
        },
        {
          "key": "xxxxxxxxxx",
          "value": "yyyyyyyyyyyy"
        },
        {
          "key": "urls",
          "value": [
  ```

```
          {
            "text": "Product Intranet Site",
            "url": "https://example.com/intranet"
          },
          {
            "text": "Compliance Policies",
            "url": "https://example.com/policies"
          }
        ]
      }
    ]
  },
  "footer": {
    "text": "This document is intended for internal use only. Do not distribute
 without permission.",
    "logo": "footer-logo.png",
    "pageNumber": true
  }
}
```

- Tajuk:
  - Untuk file PDF halaman sampul, teks default akan menjadi nama proyek.
  - Untuk setiap file PDF program, teks default akan menjadi nama program.
  - Tidak ada logo default. Jika logo header tidak dikonfigurasi, tidak ada logo yang akan ditampilkan.
  - Ukuran font dan ukuran logo harus diubah secara dinamis berdasarkan jumlah kata atau ukuran file logo.
- Bagian kustom:
  - Jika bagian kustom tidak dikonfigurasi, itu akan dihilangkan dari PDF.
  - Tautan harus dapat diklik.
- Footer:
  - Tidak ada teks atau logo default untuk footer.
  - Nomor halaman akan ditampilkan di footer secara default, kecuali secara eksplisit dikonfigurasi sebaliknya.
  - Ukuran font dan ukuran logo harus diubah secara dinamis berdasarkan jumlah kata atau ukuran file logo.

Hasilkan penampil inline dokumentasi

Anda dapat melihat file PDF di langkah menghasilkan dokumentasi teknis.

Untuk melihat file PDF

1. Arahkan ke tab Tinjau hasil dokumentasi.
2. Temukan PDF dalam daftar tabel PDF yang dihasilkan.
3. Pilih rile dan kemudian pilih Lihat atau pilih elemen tautan yang dilapis pada nama file.

PDF terbuka di AWS Transform, dengan opsi untuk memperluas layar di kanan atas.

> **Note**
>
> AWS Transform juga memberi Anda kemampuan untuk mengunduh salah satu XMLversi PDF dari dokumentasi teknis yang dihasilkan.

> **Important**
>
> Jika Anda mengalami masalah dengan penampil inline dokumentasi, pastikan bucket S3 sudah diatur dengan benar. Untuk informasi selengkapnya tentang kebijakan CORS bucket S3, lihat. the section called “Izin CORS bucket S3”

Ekstrak logika bisnis

Anda dapat mengekstrak logika bisnis penting dari aplikasi mainframe Anda yang sedang mengalami modernisasi. AWS Transform secara otomatis menganalisis kode Anda untuk mengidentifikasi dan mendokumentasikan elemen bisnis penting, termasuk alur proses terperinci, dan logika bisnis yang tertanam dalam aplikasi Anda. Kemampuan ini melayani banyak pemangku kepentingan dalam perjalanan modernisasi Anda. Analis bisnis dapat memanfaatkan logika yang diekstraksi untuk menciptakan persyaratan bisnis yang tepat dan mengidentifikasi kesenjangan atau inkonsistensi dalam implementasi saat ini. Pengembang mendapatkan kemampuan untuk dengan cepat memahami fungsionalitas sistem warisan yang kompleks tanpa keahlian mainframe yang luas.

> **Note**
>
> Ketika Anda meminta ekstraksi logika bisnis. AWS Transform melakukan analisis kode, termasuk ketergantungan kode dan analisis titik masuk, yang diperlukan untuk melakukan ekstraksi logika bisnis.

Untuk mengekstrak logika bisnis

1. Di panel navigasi kiri, di bawah Extract business logic, pilih Configure settings.
2. Di tab Kolaborasi pilih bagaimana Anda ingin mengekstrak logika bisnis:

   - Tingkat aplikasi: Menghasilkan dokumen bisnis untuk semua fungsi bisnis, transaksi, pekerjaan batch, dan file. Ini memilih semua file dalam aplikasi.
   - Tingkat file: Menghasilkan dokumen bisnis hanya untuk file yang Anda pilih dari tabel file.

   > **Note**
   >
   > - Untuk salah satu opsi, Anda dapat memilih Sertakan spesifikasi fungsional terperinci sehingga AWS Transform menyertakan alur kontrol dan aturan bisnis yang komprehensif untuk file yang dipilih.
   > - File yang dipilih harus memiliki jenis pengkodean yang sama (yaitu, semua dalam CCSID - UTF8 atau ASCII yang sama). Jika tidak, dokumentasi yang dihasilkan mungkin memiliki bidang atau bagian kosong.
   > - Dokumentasi dapat dibuat hanya untuk file COBOL dan JCL.
   > - Untuk tingkat aplikasi, program yang digunakan oleh transaksi CICS dan pekerjaan batch dikelompokkan bersama, sementara semua program lainnya dikategorikan sebagai Tidak Ditugaskan.

3. Pilih Lanjutkan.
4. Setelah AWS Transform mengekstrak logika bisnis, Transform menyimpan hasilnya dalam bucket Amazon S3 dalam format JSON sehingga Anda dapat melihatnya secara online.

> **Note**
>
> Jumlah file aturan bisnis yang dihasilkan mungkin lebih besar dari pilihan awal Anda. Beberapa file yang dipilih dapat memicu ekstraksi aturan bisnis untuk menyertakan file dependen tambahan, yang juga akan muncul di tabel hasil.

Lihat dokumentasi bisnis yang diekstraksi secara inline

Anda dapat melihat logika bisnis di langkah Extract business rule. Untuk melakukan ini,

1. Arahkan ke Tinjau hasil ekstraksi logika bisnis.
2. Pilih dokumen yang ingin Anda lihat dari tabel, lalu klik tombol Lihat hasil.

Halaman dokumentasi bisnis terbuka di tab browser baru.

Dekomposisi

Anda dapat menguraikan kode Anda ke dalam domain yang memperhitungkan dependensi antara program dan komponen. Ini membantu file dan program terkait dikelompokkan dengan tepat dalam domain yang sama. Ini juga membantu menjaga integritas logika aplikasi selama proses dekomposisi.

> **Note**
>
> Saat Anda meminta dekomposisi, AWS Transform melakukan analisis kode, termasuk analisis ketergantungan kode, yang diperlukan untuk melakukan dekomposisi. Kami juga menyarankan Anda melakukan ekstraksi logika bisnis sebelum dekomposisi untuk hasil yang lebih baik.

Untuk memulai dengan menguraikan aplikasi Anda:

1. Pilih Dekomposisi kode dari panel navigasi kiri.

> **Note**
>
> Satu domain, Unassigned, secara otomatis dibuat untuk semua file yang tidak terkait dengan domain. Pada navigasi awal, semua file harus dikaitkan dengan Unsigned, kecuali domain diusulkan dari ekstraksi logika bisnis tingkat aplikasi.

2. Buat domain baru melalui menu Tindakan, lalu pilih Buat domain.
3. Di Buat domain, berikan nama domain, deskripsi opsional, dan tandai beberapa file sebagai benih.
   - File yang dikonfigurasi CICS (CSD) dan file yang dikonfigurasi penjadwal (SCL) dapat digunakan untuk deteksi benih otomatis.
   - Anda juga dapat mengatur satu domain hanya sebagai komponen umum. File dalam domain ini umum untuk beberapa domain.
4. Pilih Buat.

> **Note**
>
> Anda dapat membuat beberapa domain dengan file yang berbeda sebagai benih.

5. Setelah mengonfirmasi semua domain dan benih, pilih Dekomposisi.
6. AWS Transform akan memeriksa file kode sumber dan kemudian terurai menjadi domain dengan program dan kumpulan data dengan kasus penggunaan serupa dan dependensi pemrograman tinggi.

   AWS Transform memberi Anda tampilan tabel dan grafik domain yang didekomposisi sebagai dependensi. Tampilan grafik memiliki tiga opsi:

   - Tampilan domain - Dapat melihat bagaimana domain yang berbeda terkait satu sama lain dalam format visual.
   - Tampilan ketergantungan - Dapat melihat semua file di setiap domain sebagai grafik ketergantungan yang kompleks. Jika node yang ditambahkan ke domain tidak menerima informasi dari benih di domain yang sama, maka node ini akan diprediksi menjadi tidak ditetapkan (node tidak menerima informasi apa pun), terputus (dalam sub grafik yang tidak menerima informasi benih) atau ke domain lain (node menerima informasi dari setidaknya domain itu).
   - Tampilan subgraf - Pengguna dapat membuat subgraf untuk memvisualisasikan subset node untuk lebih memahami dampak relasional dan batas-batas kumpulan node tersebut.

- Untuk membuat subgraf pilih sekelompok node dan pilih opsi Ekstrak subgraf dari bilah alat
- Penggabungan subgraf tersedia, saat menggabungkan subgraf ketiga baru dibuat selain dua subgraf yang Anda gabungkan.

> **Note**
>
> Ulangi langkah-langkah ini untuk menambahkan lebih banyak domain atau untuk mengkonfigurasi ulang domain yang sudah Anda buat dengan kumpulan benih yang berbeda jika Anda tidak menyukai struktur domain saat ini.

7. Setelah selesai, pilih Lanjutkan.

Biji

Benih adalah input dasar untuk fase kode dekomposisi. Setiap komponen atau file (misalnya, JCL, COBOL, tabel Db2, CSD, dan file penjadwal) dapat ditetapkan sebagai benih hanya untuk satu domain, memastikan batas dan keselarasan yang jelas selama proses dekomposisi.

Identifikasi benih tergantung pada struktur aplikasi atau portofolio. Dalam kasus aplikasi warisan mainframe yang khas, seed sering dapat ditentukan dengan mengikuti konvensi penamaan yang ditetapkan, pengelompokan tingkat batch dalam penjadwal, dan pengelompokan tingkat transaksi yang ditentukan dalam sistem CICS. Selain itu, tabel database juga dapat berfungsi sebagai benih, menyediakan lapisan struktur lain untuk dekomposisi.

Impor and/or file dependensi pembaruan

Selama dekomposisi, Anda dapat mengunggah file JSON untuk dependensi yang menggantikan file yang ada yang dihasilkan oleh analisis dependensi yang dilakukan Transform. AWS

Fungsi ekspor dependensi memungkinkan Anda mengunduh file json dependensi yang dihasilkan pada langkah dekomposisi. Setelah mengunduh, Anda dapat memodifikasi file sesuai kebutuhan Anda. Kemudian, Anda dapat mengimpor dependensi menggunakan fungsionalitas unggah AWS Transform yang memungkinkan Anda mengunggah file JSON baru dari dependensi yang menggantikan file yang dihasilkan oleh analisis dependensi. Setelah itu, grafik dalam langkah dekomposisi akan diperbarui.

Untuk mengekspor, memodifikasi, dan mengimpor dependensi

1. Pada halaman Lihat hasil dekomposisi, pilih Tindakan.

2. Dalam daftar dropdown, pilih opsi Perbarui file dependensi di bawah Tindakan lain.
3. Dalam modal file dependensi Perbarui,
   a. Unduh file dependensi AWS Transform yang dibuat dari hasil analisis yang ada.
   b. Dalam file yang diunduh, ubah dependensi berdasarkan apa yang ingin Anda capai.
   c. Setelah memodifikasi, simpan dan unggah file ini menggunakan tombol Unggah file ketergantungan.

> **Note**
>
> Satu-satunya format file yang diterima adalah file JSON.

4. Selanjutnya, pilih Impor.

AWS Transform akan mengimpor file dependensi dan membuat grafik dependensi baru berdasarkan masukan Anda.

Impor and/or Perbarui Domain

Untuk pelanggan yang memiliki domain, benih, hubungan and/or file yang dipetakan sebelum langkah dekomposisi, Anda dapat mengunggah definisi domain ini melalui fungsi file Impor domain yang tersedia melalui menu Tindakan. Beberapa contoh kapan fungsi ini dapat digunakan:

- Membawa dekomposisi ke depan dari pekerjaan lain
- Pakar materi pelajaran yang menyediakan pemetaan ini

Setelah file domain diimpor, pengguna dapat menjalankan dekomposisi terhadap definisi domain, atau jika puas dapat Menyimpan dan kemudian Kirim definisi domain.

Parent/child/neighbor berkas

Dalam grafik dependensi, program berhubungan satu sama lain melalui berbagai jenis koneksi. Memahami hubungan ini membantu Anda menganalisis dependensi program selama transformasi aplikasi mainframe Anda. Ini juga membantu memahami batas-batas domain. Misalnya, jika Anda memilih domain, dan kemudian memilih induk satu tingkat, itu akan menunjukkan node yang terhubung.

Hubungan orang tua — File induk memanggil atau mengontrol program lain. Orang tua duduk di atas program dependen mereka dalam hierarki. Anda dapat memilih orang tua di satu tingkat atau di semua tingkatan.

Hubungan anak — File anak dipanggil atau dikendalikan oleh program induk. Anak-anak duduk di bawah orang tua mereka dalam hierarki file.

Hubungan tetangga — Tetangga adalah file pada tingkat hierarkis yang sama. Mereka berbagi program induk yang sama dan mungkin berinteraksi satu sama lain secara langsung.

Kode Reforge

Reforge menggunakan Large Language Models (LLM) untuk meningkatkan kualitas kode refactored. COBOL-to-Java Transformasi awal mempertahankan kesetaraan fungsional sambil mempertahankan struktur COBOL-influenced data dan nama variabel dari sistem warisan. Reforge merestrukturisasi kode ini untuk mengikuti praktik dan idiom Java modern, menggantikan COBOL-style konstruksi dengan koleksi Java asli dan konvensi penamaan. Ini membuat kode lebih mudah dibaca dan dipelihara untuk pengembang Java.

> **Note**
>
> Kuota untuk reforge adalah:
>
> - 3.000.000 baris kode per pekerjaan
> - 50.000.000 baris kode per pengguna per bulan

Reformasi kode Anda setelah refactoring dengan mengikuti langkah-langkah ini:

1. Pilih Reforge kode java di panel navigasi kiri dan kemudian pilih Configure code reforge.
2. Berikan lokasi S3 ke proyek sumber bawaan zip Anda dan pilih Lanjutkan. Gunakan struktur zip ini:

```
input.zip
        ### PROJECT-pom
        ### PROJECT-entities
        ### PROJECT-service
        ### PROJECT-tools
        ### PROJECT-web (optional)
```

```
        ### pom.xml
```

AWS Transform menganalisis paket zip Anda untuk menemukan file dalam PROJECT-service direktori sehingga dapat memberikan daftar kelas yang dapat dipilih yang dapat Anda reforge. Kelas-kelas ini memiliki akhiran`ProcessImpl.java`.

3. Selesaikan halaman Select class to reforge dan pilih Continue. Lacak status reforge pada tab Worklog.
4. Lihat hasil reforge Anda yang telah selesai di halaman Lihat hasil, yang menampilkan status reforge per kelas. Ini juga menentukan di mana menemukan hasil Reforge di bucket S3 Anda.

Setelah AWS Transform mendapatkan masukan ini dari Anda, ia memberi Anda file yang dapat diunduh dengan hasil Reforge.

Ini adalah struktur zip yang dihasilkan dari reforge yang sukses:

```
reforge.zip
### maven_project
### reforge.log
###tokenizer_map.json
```

- maven_project berisi kode sumber reforged.
  - File yang telah difaktorkan ulang tetapi kompilasi yang tidak berhasil diselesaikan berada di `/src/main/resources/reforge/originalClassName.java.incomplete` dan diberi nama. `originalClassName.java.incomplete` Bandingkan ini dengan versi asli file untuk memilih fungsi reforged yang ingin Anda simpan.
  - File sumber yang disediakan untuk AWS Transform yang berhasil difaktorkan ulang dicadangkan `src/main/resources/reforge/originalClassName.java.original` dan diberi nama. `originalClassName.java.original` Versi refactored file menggantikan file sumber yang disediakan untuk Transform. AWS

> **Note**
>
> `originalClassName.java`File diganti dengan file reforged hanya jika proses reforging berhasil. Jika tidak, mereka mempertahankan konten asli.

- reforge.log berisi log yang dapat Anda gunakan untuk mendiagnosis kegagalan pekerjaan atau memberikan AWS dukungan jika terjadi masalah.

- tokenizer_map.json berisi pemetaan ID token ke data Anda, seperti jalur file dan class/method nama, yang diberi token di log untuk perlindungan privasi. Anda dapat memberikan file ini untuk AWS mendukung jika terjadi masalah.

Rencanakan pengujian aplikasi modern Anda

Anda dapat membuat dan mengelola rencana pengujian untuk aplikasi modern mainframe Anda berdasarkan atribut kode yang diekstraksi, kompleksitas pekerjaan, dan jalur penjadwal. AWS Transform membantu memprioritaskan pekerjaan mana yang akan diuji dan mengidentifikasi artefak spesifik yang diperlukan untuk setiap kasus uji. Proses perencanaan pengujian dibagi menjadi tiga fase utama: konfigurasi, pelingkupan, dan tinjauan.

Untuk membuat rencana pengujian

1. Konfigurasikan pengaturan rencana pengujian

   a. Di panel navigasi kiri, di bawah Rencanakan pengujian Anda, pilih Konfigurasi pengaturan.

   b. (Opsional) Berikan jalur S3 untuk Ekstraksi Logika Bisnis (BLE) Anda. Artefak ini meningkatkan kualitas rencana pengujian. Tanpa artefak BLE, beberapa bidang dalam rencana pengujian mungkin tetap tidak lengkap.

2. Tentukan ruang lingkup rencana pengujian

   a. Pilih titik masuk, seperti pekerjaan batch, untuk disertakan dalam rencana pengujian Anda.

   b. Filter dan urutkan pekerjaan berdasarkan beberapa atribut:

      - Fungsi bisnis (diekstrak dari BRE BLE)
      - Domain (dari fase dekomposisi)
      - Jalur dan lokasi file
      - Kriteria pencarian kustom

   c. Pilih pekerjaan individu atau seluruh kelompok untuk pengujian.

   d. Tinjau hubungan kerja dan dependensi.

3. Tinjau dan sesuaikan rencana pengujian

   Rencana pengujian yang dihasilkan memberikan informasi yang komprehensif termasuk:

   - Urutan eksekusi yang disukai berdasarkan dependensi
   - Penugasan kelompok pekerjaan dari penjadwal

- Skor kompleksitas, yang merupakan skor agregat untuk kasus uji
- Asosiasi domain bisnis
- Metrik kompleksitas siklomatik
- Dataset dan dependensi tabel
- Baris metrik kode
- Pemetaan fungsi bisnis

Opsi kustomisasi rencana uji

Rencana pengujian Anda dapat disesuaikan untuk memenuhi kebutuhan spesifik. Misalnya, Anda dapat:

- Buat kasus uji baru dengan memilih beberapa titik masuk
- Gabungkan kasus uji yang ada untuk menggabungkan fungsionalitas terkait
- Split test case untuk pengujian yang lebih granular
- Hapus kasus uji yang tidak perlu
- Menambah atau menghapus titik masuk untuk kasus uji yang ada
- Ubah deskripsi dan atribut kasus uji
- Sesuaikan perintah eksekusi

Informasi kasus uji terperinci

Setiap kasus uji memberikan detail yang menjelaskan kontennya dan kumpulan data atau data terkait:

- Deskripsi komprehensif tentang ruang lingkup pengujian
- Daftar lengkap titik masuk termasuk
- Metrik agregat yang menunjukkan kompleksitas dan ukuran
- Aturan bisnis dan panduan kasus uji otomatis
- Dataset dan dependensi tabel dengan direction () input/output
- Visualisasi grafik ketergantungan interaktif
- Prasyarat dan persyaratan eksekusi

Fitur manajemen data

Detail ini membantu Anda memahami data yang terkait dengan kasus uji. Anda dapat:

- Filter kumpulan data berdasarkan input/output arah
- Identifikasi artefak yang diperlukan untuk eksekusi pengujian
- Lacak dependensi data antara kasus uji
- Pantau penggunaan kumpulan data di seluruh rencana pengujian

> **Note**
>
> AWS Transform secara otomatis menganalisis dependensi penjadwal dan menetapkan skor kompleksitas untuk membantu memprioritaskan upaya pengujian. Skor kompleksitas yang lebih tinggi menunjukkan pekerjaan yang mungkin memerlukan pengujian atau isolasi yang lebih menyeluruh selama proses pengujian.

Aturan bisnis dan panduan kasus uji

Rencana pengujian AWS Transform memberikan rekomendasi untuk rencana kasus pengujian Anda berdasarkan Ekstraksi Aturan Bisnis:

- Pemrosesan otomatis aturan bisnis menggunakan LLM
- Generasi kasus uji sintetis
- Panduan pengujian untuk skenario bisnis tertentu
- Ketertelusuran antara aturan dan kasus uji

Rencana pengujian akhir disimpan di lokasi S3 yang ditentukan dan mencakup semua informasi yang diperlukan untuk menjalankan strategi pengujian Anda secara efektif. Anda dapat mengekspor rencana pengujian untuk integrasi dengan alat pengujian atau sistem dokumentasi lainnya.

> **Note**
>
> Sementara panduan kasus uji sintetis disediakan, artefak uji yang sebenarnya harus dibuat secara terpisah berdasarkan panduan. Rencana pengujian berfungsi sebagai cetak biru

komprehensif untuk strategi pengujian Anda tetapi tidak menghasilkan data pengujian atau skrip eksekusi.

Aturan pembuatan kasus uji - Ringkasan

Aturan umum

- Kasus uji berisi 1 hingga banyak JCL dalam urutan eksekusi jadwal
- Kasus uji dibuat dari penjadwal yang didukung valid (CA7 dan) Control-M
- Tes mengeksekusi JCL, bukan penjadwal itu sendiri atau tugas terjadwal
- Tugas terjadwal unik dan hanya mengeksekusi satu JCL
- Satu JCL dapat dieksekusi oleh beberapa tugas terjadwal
- Satu JCL bisa ada tanpa berada dalam jadwal

Aturan pembuatan kasus uji default

- Jika satu JCL berada dalam kasus uji:
  - JCL tidak terlibat dengan jadwal
  - JCL dijalankan oleh tugas terjadwal yang menyimpang ke beberapa cabang
  - Cabang dalam jadwal hanya berisi JCL ini
- Jika beberapa JCL dalam kasus uji: mewakili urutan jalur eksekusi linier dalam cabang jadwal
- Jika JCL berada pada tugas terjadwal yang berbeda: JCL menjadi kasus uji terpisahnya sendiri
- Jika JCL berada pada tugas terjadwal konvergen: JCL dapat memulai kasus uji baru tetapi tidak akan disertakan di cabang sebelumnya
- JCL yang hilang dilewati dan eksekusi berlanjut (dengan peningkatan future yang direncanakan untuk memotong kasus uji pada titik JCL yang hilang)

Aturan operasi pengguna

- Buat Kasus Uji: pengguna memilih dari JCL yang tersedia
  - Berhasil jika JCL ada dalam urutan cabang eksekusi jadwal
  - Mengikuti perintah eksekusi jadwal
- Tambahkan JCL ke Test Case: pengguna memilih dari JCL yang tersedia

  - Berhasil jika JCL dapat ada dalam eksekusi jadwal kasus uji branch/path
- Hapus JCL dari Test Case: pengguna dapat menghapus JCL apa pun dari kasus uji
  - Diizinkan bahkan jika itu menyebabkan celah jalur eksekusi
- Gabungkan Kasus Uji: pengguna memilih dua kasus uji untuk digabungkan
  - Berhasil jika JCL dapat eksis bersama di cabang eksekusi jadwal yang sama
  - Menjaga jadwal eksekusi order
- Split Test Case: pengguna memilih satu JCL dalam kasus uji untuk dibagi
  - Membuat kasus uji baru dari titik terpisah ke depan
  - Kasus uji asli dimodifikasi untuk mengecualikan JCL melewati titik split
- Hapus Kasus Uji: pengguna dapat menghapus kasus uji apa pun yang dibuat

> **Note**
>
> Anda tidak dapat membuat, menambah, atau menggabungkan kasus uji dari penjadwal branches/path yang berbeda. Peningkatan future direncanakan untuk memungkinkan operasi di luar divergent/convergent tugas di penjadwal.

Hasilkan skrip pengumpulan data uji

Anda dapat membuat skrip JCL untuk mengumpulkan data pengujian dari sistem mainframe Anda berdasarkan rencana pengujian yang dibuat pada langkah sebelumnya. AWS Transform secara otomatis membuat skrip pengumpulan data untuk kumpulan data, tabel database, dan file berurutan yang diperlukan untuk pengujian komprehensif. Proses pengumpulan data dibagi menjadi empat fase utama: konfigurasi input, pemilihan kasus uji, konfigurasi skrip, dan pembuatan skrip.

Untuk menghasilkan skrip pengumpulan data uji

1. Berikan masukan rencana pengujian

   a. Di panel navigasi kiri, di bawah Pengumpulan data uji, pilih Berikan masukan rencana pengujian.

   b. Tentukan jalur S3 ke file JSON rencana pengujian Anda dari Rencanakan pengujian aplikasi modern Anda.

c. Bidang input diisi sebelumnya jika rencana pengujian dibuat pada langkah pekerjaan sebelumnya.

d. Anda juga dapat memilih rencana pengujian dari pekerjaan lain dengan menentukan lokasi S3 yang sesuai.

2. Pilih kasus uji untuk pengumpulan data

a. Tinjau daftar lengkap kasus uji dari rencana pengujian Anda.

b. Filter dan urutkan kasus uji berdasarkan beberapa atribut:

- Fungsi dan domain bisnis
- Dependensi tabel database
- Persyaratan kumpulan data
- Metrik kompleksitas
- Kriteria pencarian kustom

c. Pilih kasus uji individual atau gunakan opsi pemilihan massal.

d. Tinjau detail kasus uji termasuk titik masuk, metrik, dan aturan bisnis dengan mengklik satu Kasus Uji untuk melihat detailnya.

3. Konfigurasikan skrip pengumpulan data

a. Unduh contoh templat dan file konfigurasi untuk referensi.

- AWS Transform menyediakan contoh template untuk pembongkaran database Db2, file VSAM REPRO dan pemrosesan dataset sekuensial untuk digunakan sebagai panduan tentang jenis template yang diharapkan oleh proses.
- Standar dapat bervariasi dari situs ke situs sehingga harapannya adalah bahwa pelanggan akan memodifikasi atau mengganti template ini yang sesuai dengan standar mereka sendiri.
- Template yang dimodifikasi ini perlu diunggah ke bucket S3 tempat pengumpulan data pengujian dapat memprosesnya.

b. Menyediakan file konfigurasi variabel (format JSON) yang berisi:

- Awalan pengguna dan konstanta khusus lingkungan
- Parameter konfigurasi basis data
- Pengaturan titik akhir tujuan dan parameter transfer data

- Parameter lain yang diperlukan yang ditentukan oleh pengguna dan untuk digunakan dalam template JCL

c. Unggah template JCL untuk metode pengumpulan data yang berbeda:

- Template Db2: Untuk pembongkaran tabel database (sesuaikan untuk BMC, IBM DSN, atau utilitas bongkar lainnya)
- Template VSAM: Untuk pemrosesan file VSAM (biasanya menggunakan utilitas REPRO)
- Templat kumpulan data berurutan: Untuk memproses kumpulan data berurutan, kumpulan data yang dipartisi, GDG, dll.

4. Tinjau dan kelola skrip yang dihasilkan

Skrip yang dihasilkan menyediakan kemampuan pengumpulan data yang komprehensif termasuk:

- Skrip terpisah untuk pengumpulan data eksekusi uji “sebelum” dan “setelah”
- Struktur skrip terorganisir berdasarkan kasus uji dan tipe data
- Skrip secara otomatis disimpan ke dalam bucket S3 untuk memudahkan akses dan transfer
- Skrip JCL yang dihasilkan siap untuk eksekusi mainframe
- Substitusi variabel didasarkan pada konfigurasi pengguna yang ditentukan dalam template

Fitur pembuatan skrip

Skrip yang dihasilkan secara otomatis disesuaikan berdasarkan templat dan konfigurasi Anda:

- Template-based generasi: Menggunakan template JCL yang Anda berikan dengan substitusi variabel
- Lingkungan: Menggabungkan konfigurasi mainframe spesifik Anda
- Penanganan tipe data: Membuat skrip yang sesuai untuk kumpulan data berurutan, file VSAM, dan tabel database
- Koleksi untuk kasus uji: Menghasilkan skrip pengumpulan data “sebelum” dan “setelah”
- Pemrosesan kumpulan data berurutan: sampel yang AWS disediakan menyediakan fungsi transfer file, tetapi ini dapat disesuaikan dengan utilitas kompresi yang tersedia di situs atau utilitas Anda seperti Connect Direct atau transfer file terkelola, dll.

Strategi pengumpulan data

Skrip yang dihasilkan mendukung strategi pengumpulan data yang komprehensif:

- Koleksi dataset sekuensial: REPRO dan utilitas salin untuk VSAM dan file datar
- Pembongkaran tabel database: Proses pembongkaran Db2 yang dapat disesuaikan
- Pemrosesan kumpulan data berurutan: Pemrosesan pos yang dapat disesuaikan dari kumpulan data berurutan seperti kompresi, layanan transfer file terkelola, dll.
- Manajemen ketergantungan: Koleksi terkoordinasi berdasarkan definisi kasus uji

> **Note**
>
> AWS Transform menghasilkan skrip berdasarkan templat dan konfigurasi Anda. Tinjau semua JCL yang dihasilkan sebelum mengeksekusi di lingkungan mainframe Anda untuk memastikan kompatibilitas dengan konfigurasi sistem spesifik dan persyaratan keamanan Anda.

Kustomisasi template dan praktik terbaik

Pengumpulan Data Uji ATX menyediakan kemampuan kustomisasi template yang fleksibel:

- Multi-utility dukungan: Adaptasi template untuk utilitas mainframe yang berbeda (BMC, IBM, DSN)
- Variable-driven konfigurasi: Gunakan konstanta untuk parameter khusus lingkungan
- Template yang dapat digunakan kembali: Buat templat standar untuk pembuatan skrip yang konsisten
- Penanganan data: Menggabungkan persyaratan penanganan data khusus organisasi
- Integrasi keamanan: Sertakan kontrol keamanan dan akses yang sesuai
- Optimalisasi kinerja: Konfigurasikan untuk pengumpulan dan transfer data yang efisien

Struktur keluaran yang dihasilkan

Skrip yang dihasilkan diatur dalam bucket S3 Anda dengan struktur berikut:

- Organisasi kasus uji: Skrip dikelompokkan berdasarkan kasus uji terkait
- Waktu pengumpulan: Pisahkan folder untuk pengumpulan data “sebelum” dan “setelah”

- Klasifikasi tipe data: Skrip yang diatur oleh kumpulan data berurutan, tabel database, dan transfer
- File metadata: Informasi ringkasan dan panduan eksekusi
- Ready-to-transfer format: JCL diformat untuk penyebaran mainframe langsung

Koleksi skrip akhir disimpan di lokasi S3 yang ditentukan dan mencakup semua JCL yang diperlukan untuk menjalankan strategi pengumpulan data Anda secara efektif. Anda dapat mengunduh skrip untuk ditransfer ke lingkungan mainframe Anda atau mengintegrasikan dengan proses penerapan otomatis.

> **Note**
>
> Sementara skrip JCL komprehensif dihasilkan, eksekusi aktual harus dilakukan di lingkungan mainframe Anda. Skrip berfungsi sebagai alat pengumpulan data yang siap digunakan tetapi memerlukan akses mainframe yang sesuai dan izin eksekusi.

Uji pembuatan skrip otomatisasi

Anda dapat membuat skrip otomatisasi pengujian untuk menjalankan kasus pengujian pada aplikasi modern Anda berdasarkan rencana pengujian yang dibuat pada langkah sebelumnya. AWS Transform secara otomatis membuat skrip pengujian komprehensif yang memanfaatkan data yang dikumpulkan dari proses pengumpulan data pengujian. Proses pembuatan skrip otomatisasi pengujian terdiri dari tiga fase utama: konfigurasi input, pemilihan kasus uji, dan hasil pembuatan skrip.

Untuk menghasilkan skrip pengujian

1. Berikan masukan rencana pengujian

    a. Di panel navigasi kiri, di bawah Pembuatan skrip otomatisasi pengujian, pilih Berikan masukan rencana pengujian.
    b. Tentukan jalur S3 ke file JSON rencana pengujian Anda dari Rencanakan pengujian aplikasi modern Anda.
    c. Bidang input diisi sebelumnya jika rencana pengujian dibuat pada langkah pekerjaan sebelumnya.
    d. Anda juga dapat memilih rencana pengujian dari pekerjaan lain dengan menentukan lokasi S3 yang sesuai.

e. Sistem menggunakan rencana pengujian ini sebagai dasar untuk menghasilkan skrip otomatisasi.

2. Pilih kasus uji untuk pembuatan skrip

   a. Tinjau daftar lengkap kasus uji dari rencana pengujian Anda.

   b. Filter dan urutkan kasus uji berdasarkan beberapa atribut:

      - Fungsi dan domain bisnis
      - Dependensi tabel database
      - Persyaratan kumpulan data
      - Metrik kompleksitas
      - Kriteria pencarian kustom

   c. Pilih kasus uji individual atau gunakan opsi pemilihan massal dengan tombol Periksa Semua dan Hapus Centang Semua.

   d. Tinjau detail kasus uji termasuk titik masuk, metrik, dan aturan bisnis dengan mengklik kasus uji individual.

      Kasus uji yang dipilih akan memiliki skrip otomatisasi yang dihasilkan untuk dieksekusi pada aplikasi yang dimodernisasi.

3. Tinjau dan kelola skrip otomatisasi pengujian yang dihasilkan:

   - Sistem menampilkan pesan sukses yang mengonfirmasi penyelesaian pembuatan skrip.
   - Skrip pengujian yang dihasilkan secara otomatis disimpan di lokasi bucket S3 yang Anda tentukan.
   - Akses daftar lengkap skrip pengujian yang dihasilkan dengan lokasi S3 masing-masing.
   - Setiap kasus uji memiliki skrip otomatisasi yang sesuai yang disimpan di lokasi S3 individu.
   - Skrip siap untuk penyebaran dan eksekusi di lingkungan aplikasi modern Anda.

Uji fitur skrip otomatisasi

Skrip otomatisasi yang dihasilkan memberikan kemampuan pengujian yang komprehensif:

- Pengujian aplikasi modern: Skrip dirancang khusus untuk mengeksekusi kasus uji pada aplikasi Anda yang diubah

- Integrasi data: Memanfaatkan data uji yang dikumpulkan dari langkah pengumpulan data uji sebelumnya, data ini perlu disalin ke folder untuk setiap kasus uji
- Eksekusi otomatis: Skrip dapat digunakan untuk mengatur sink data, menjalankan kasus uji, dan membandingkan hasil, beberapa parameter harus diatur sesuai dengan lingkungan penerapan Anda
- Struktur terorganisir: Skrip diatur secara sistematis berdasarkan kasus uji di bucket S3 Anda
- Ready-to-deploy format: Skrip diformat untuk penyebaran langsung ke lingkungan pengujian Anda

Struktur keluaran yang dihasilkan

Skrip otomatisasi pengujian yang dihasilkan diatur dalam bucket S3 Anda dengan struktur berikut:

- Organisasi kasus uji: Setiap kasus uji memiliki skrip khusus yang disimpan di folder S3 individual
- Execution-ready format: Skrip diformat untuk penyebaran dan eksekusi segera setelah menyiapkan beberapa variabel tergantung dari lingkungan Anda
- Akses terpusat: Semua skrip dapat diakses dari satu lokasi bucket S3 untuk pengelolaan yang mudah

Strategi eksekusi uji

Skrip yang dihasilkan mendukung alur kerja eksekusi pengujian yang komprehensif:

- Pengaturan lingkungan: Skrip menyertakan kemampuan untuk menyiapkan data awal untuk menjalankan pengujian
- Persiapan data: Integrasi dengan data uji yang dikumpulkan dari langkah pengumpulan data kasus uji
- Eksekusi kasus uji: Eksekusi otomatis kasus uji individual pada aplikasi yang dimodernisasi
- Perbandingan hasil: Built-in kemampuan untuk membandingkan hasil pengujian dan memvalidasi perilaku aplikasi

> **Note**
>
> AWS Transform menghasilkan skrip otomatisasi pengujian berdasarkan rencana pengujian dan kasus pengujian yang dipilih. Skrip dirancang untuk dieksekusi di lingkungan aplikasi modern Anda dan memanfaatkan data pengujian yang dikumpulkan pada langkah

sebelumnya. Tinjau semua skrip yang dihasilkan sebelum penerapan untuk memastikan kompatibilitas dengan konfigurasi aplikasi spesifik Anda dan persyaratan pengujian.

Praktik terbaik untuk otomatisasi pengujian

- Validasi lingkungan: Pastikan lingkungan aplikasi modern Anda dikonfigurasi dengan benar sebelum eksekusi skrip
- Verifikasi data: Validasi bahwa data pengujian yang diperlukan dari fase pengumpulan tersedia dan dapat diakses
- Kustomisasi skrip: Tinjau dan sesuaikan skrip yang dihasilkan sesuai kebutuhan untuk persyaratan pengujian spesifik Anda
- Pemantauan eksekusi: Menerapkan pemantauan dan pencatatan yang tepat selama eksekusi skrip pengujian
- Analisis hasil: Menetapkan proses untuk menganalisis hasil tes dan mengidentifikasi masalah aplikasi

Koleksi akhir skrip otomatisasi pengujian menyediakan kerangka pengujian lengkap untuk memvalidasi fungsionalitas aplikasi modern Anda. Skrip dapat diintegrasikan ke dalam proses pengujian berkelanjutan Anda atau dijalankan sebagai bagian dari alur kerja validasi aplikasi Anda.

Kemampuan penyebaran di AWS Transformasi

AWS Transform membantu Anda mengatur lingkungan cloud untuk aplikasi mainframe modern dengan menyediakan template Infrastructure as Code (IAc) siap pakai. Melalui antarmuka obrolan AWS Transform, Anda dapat mengakses template bawaan yang membuat komponen penting seperti sumber daya komputasi, database, penyimpanan, dan kontrol keamanan. Template tersedia dalam format populer termasuk CloudFormation (CFN),, dan Terraform AWS Cloud Development Kit (AWS CDK), memberi Anda fleksibilitas untuk menerapkan infrastruktur Anda.

Template ini berfungsi sebagai blok bangunan yang mengurangi waktu dan keahlian yang diperlukan untuk mengonfigurasi lingkungan untuk aplikasi mainframe modern Anda. Anda dapat menyesuaikan template ini agar sesuai dengan kebutuhan Anda, memberi Anda dasar untuk membangun lingkungan penerapan Anda.

Untuk mengambil template IAC, tanyakan di obrolan AWS Transform untuk Infrastructure-as-Code templat yang mengklarifikasi pola modernisasi pilihan Anda (seperti AWS Transform untuk refactor

mainframe), topologi pilihan Anda (mandiri vs ketersediaan tinggi), dan format pilihan Anda (vs Cloud Development Kit vs Terraform). CloudFormation

Bangun dan terapkan aplikasi modern pasca-refactoring

Setelah Anda menyelesaikan proses refactoring dengan AWS Transform, Anda dapat membangun dan menerapkan aplikasi Java modern Anda. Panduan ini memandu Anda untuk mengambil kode modern, mengonfigurasi lingkungan, serta menerapkan serta menguji aplikasi Anda.

> **Note**
>
> Selain panduan yang diberikan di sini, paket kode yang dihasilkan AWS Transform akan menyertakan dokumen Pengaturan Lingkungan Pengembangan Refactor AWS Otomatis yang memberikan instruksi untuk menyiapkan IDE (Integrated Development Environment) dengan Developer Runtime.

Topik

Prasyarat

Sebelum Anda mulai, pastikan Anda memiliki:

- Berhasil menyelesaikan pekerjaan refactoring dengan Transform. AWS
- Akses ke kode modern Anda.
- Tumpukan alat perangkat lunak build yang diinstal dan dikonfigurasi pada mesin pengembangan Anda, seperti Apache Maven atau Apache Tomcat. Untuk informasi selengkapnya tentang pembuatan versi Runtime, lihat AWS Transform untuk catatan rilis Runtime mainframe.

- Menginstal dan mengkonfigurasi Amazon Corretto atau versi runtime Java. Untuk informasi lebih lanjut tentang menginstal Amazon Corretto, lihat Amazon Corretto 24.
- Akses untuk membuat dan mengonfigurasi database Amazon Aurora PostgreSQL untuk komponen Runtime, jika perlu. Untuk informasi selengkapnya tentang Membuat database PostgreSQL Aurora, lihat Bekerja dengan Amazon Aurora PostgreSQL.
- Akses administratif untuk menyebarkan aplikasi ke lingkungan runtime Anda.
- Meninjau konsep AWS Transform for mainframe Runtime untuk konsep dasar pada aplikasi yang dimodernisasi dengan AWS solusi refactoring otomatis.

Langkah 1: Ambil kode yang dimodernisasi

Untuk mengambil kode yang dimodernisasi

1. Unduh dan ekstrak paket kode yang dihasilkan.
2. Buka `codebase/app-pom/pom.xml` dan catat versi mesin runtime yang diperlukan. Sebagai contoh, `<gapwalk.version>4.6.0</gapwalk.version>`.
3. Temukan dokumen Set up the AWS Automated Refactor Development Environment dari paket kode yang diunduh untuk referensi.

Langkah 2: Bangun aplikasi modern

Untuk membangun aplikasi modern Anda

1. Akses Runtime dari AWS Transform for mainframe Toolbox.
2. Unduh dan instal versi runtime yang sesuai (diidentifikasi dalamthe section called “Langkah 1: Ambil kode yang dimodernisasi”) di mesin pengembangan lokal Anda.

   > **Note**
   >
   > Informasi tambahan untuk menginstal dependensi runtime pada mesin lokal Anda tersedia di bagian 3.1 dari dokumen Mengatur Lingkungan Pengembangan Refactor AWS Otomatis.

3. Buka command prompt dan arahkan ke direktori root aplikasi Anda.
4. Untuk membuat paket yang dapat di-deploy untuk aplikasi modern, jalankan perintah Maven build:

```
mvn package
```

Lihat halaman Organisasi Aplikasi untuk detail tentang organisasi dasar kode modern. Misalnya, untuk aplikasi modern yang berisi aplikasi web front-end, Anda mungkin mengharapkan setidaknya `.war` agregat deployable berikut selain komponen runtime:

- Proyek layanan: Berisi elemen modernisasi logika bisnis lama

```
<business-app>-service.*.war
```

- Proyek web: Berisi modernisasi elemen terkait antarmuka pengguna

```
<business-app>-web.*.war
```

Langkah 3: Konfigurasikan lingkungan pengujian

Untuk mengonfigurasi lingkungan pengujian

1. Konfigurasikan runtime aplikasi modern Anda. Untuk informasi selengkapnya, lihat bagian Mengatur konfigurasi untuk Runtime di panduan AWS Mainframe Modernization pengguna.

> **Note**
>
> Lihat bagian 5 dari panduan Menyiapkan Lingkungan Pengembangan Refactor AWS Otomatis untuk contoh konfigurasi spesifik komponen runtime.

2. Siapkan set data input dan output (I/O) untuk aplikasi modern. Aplikasi yang dimodernisasi dapat memproses kumpulan data berurutan, kumpulan I/O data VSAM, atau lainnya.

> **Note**
>
> Lihat bagian 6 dari Mengatur Lingkungan Pengembangan Refactor AWS Otomatis untuk contoh.

3. Lingkungan runtime - Anda dapat menggunakan lingkungan runtime yang ada atau membuat lingkungan runtime baru.

- Untuk mengonfigurasi lingkungan runtime yang tidak dikelola, lihat Menyiapkan aplikasi yang tidak dikelola.
- Untuk mengonfigurasi lingkungan runtime terkelola, lihat Menyiapkan aplikasi terkelola.

Setelah mengonfigurasi lingkungan pengujian, Anda pindah ke langkah berikutnya dalam menerapkan aplikasi yang dimodernisasi.

Langkah 4: Menyebarkan aplikasi yang dimodernisasi

Menyebarkan artefak aplikasi di runtime yang Anda buat and/or dikonfigurasi di bagian dan. the section called “Langkah 2: Bangun aplikasi modern” the section called “Langkah 3: Konfigurasikan lingkungan pengujian”

Panduan tambahan untuk menerapkan aplikasi modern dapat ditemukan menggunakan tautan ini:

- Terapkan di Amazon EC2
- Terapkan pada wadah di Amazon ECS dan Amazon EKS
- Buat AWS Mainframe Modernization aplikasi

Langkah 5: Uji aplikasi yang dimodernisasi

Setelah penyebaran,

1. Tinjau Runtime API yang tersedia untuk cara berinteraksi dengan aplikasi modern.
2. Uji aplikasi Anda untuk menyelaraskan kesetaraan fungsionalnya dengan aplikasi lama. Misalnya, lihat Menguji contoh aplikasi dalam panduan AWS Mainframe Modernization pengguna.

Contoh tambahan

Untuk contoh spesifik memodernisasi aplikasi mainframe dengan AWS Transform, lihat Memodernisasi aplikasi mainframe. CardDemo

Full-stack Modernisasi Windows

AWS Transform menyediakan modernisasi Windows full-stack, termasuk modernisasi .NET dan modernisasi SQL Server.

Agen AWS Transform untuk.NET dapat membantu Anda memodernisasi aplikasi.NET agar kompatibel dengan .NET lintas platform. Anda dapat memodernisasi .NET menggunakan aplikasi web.NET atau IDE.NET.

AWS Transform for SQL Server modernisasi adalah AI-powered layanan yang mengotomatiskan modernisasi full-stack database Microsoft SQL Server dan aplikasi.NET terkait dengan Amazon Aurora PostgreSQL. Layanan mengatur seluruh perjalanan migrasi dari konversi skema, migrasi data, dan memodifikasi kode aplikasi agar sesuai dengan PostgreSQL target baru, membuat tim Anda lebih produktif dengan mengotomatiskan tugas yang kompleks dan padat karya.

Topik

- Modernisasi .NET dengan AWS Transformasi
- Modernisasi SQL Server

Modernisasi .NET dengan AWS Transformasi

Agen AWS Transform untuk.NET dapat membantu Anda memodernisasi aplikasi.NET agar kompatibel dengan .NET lintas platform. Kemampuan ini disebut modernisasi .NET. Setelah AWS Mengubah lingkungan di AWS Transform, Anda dapat membuat pekerjaan transformasi modernisasi .NET.

Pengalaman

AWS Transform tersedia dalam beberapa pengalaman:

- Konsol web: untuk transformasi skala besar hingga 500 repositori sekaligus.
- Visual StudioIDE: transformasi solusi atau proyek yang dipimpin pengembang yang bekerja bersama agen secara interaktif dan berulang. Direkomendasikan untuk proyek menengah hingga besar.

Anda dapat dengan bebas beralih antara konsol web dan IDE untuk mengerjakan pekerjaan transformasi.

Anda juga dapat memanggil AWS Transform dari IDE lain dan pendamping kode AI:

- Kiro: transformasi dari Kiro menggunakan AWS Transform untuk kekuatan Kiro.
- Pendamping kode AI lainnya: ubah dari asisten pengkodean AI pilihan Anda menggunakan agen AWS Transform MCP.

Kemampuan dan fitur utama

- Analisis basis kode.NET Framework dari sistem kontrol sumber Anda, yang mencakup NuGet dukungan pribadi, mengidentifikasi dependensi lintas repositori, dan menyediakan laporan analisis.
- Transformasi otomatis aplikasi.NET Framework lama menjadi .NET lintas platform, pemberitahuan email, dan laporan ringkasan transformasi.
- Integrasi yang mudah dengan platform kontrol sumber (BitBucket, GitHub, dan GitLab) untuk menyerap kode yang ada dan mengkomit kode yang diubah ke cabang baru.
- Validasi kode yang ditransformasikan melalui pengujian unit.

Versi dan jenis proyek yang didukung

AWS Transform mendukung transformasi untuk versi .NET ini:

- .NET Framework 3.5+
- .NET Core 3.1
- .NET 5.x +
- .NET 8

AWS Transform dapat mengubah ke versi.NET target ini:

- .NET 8
- .NET 10

AWS Transform mendukung transformasi jenis proyek ini (hanya C #):

- Perpustakaan kelas
- Aplikasi konsol

- ASP.NET:
  - Model View Controller (MVC), termasuk Razor Views front-end
  - Aplikasi Halaman Tunggal (SPA) back-end (lapisan logika bisnis)
  - API Web
  - Formulir Web
- Proyek pengujian unit (NUnit, XUnit, dan MStest)
- Layanan Windows Communication Foundation (WCF)
- Proyek dengan versi lintas platform yang disediakan untuk NuGet paket pihak ketiga atau pribadi. Jika padanan lintas platform hilang atau tidak tersedia, AWS Transform for .NET akan mencoba konversi dengan upaya terbaik.

AWS Transform juga dapat mengubah jenis proyek berikut menjadi .NET modern. Ini adalah fitur pratinjau, dan mungkin tidak berubah selengkap jenis proyek yang didukung.

- WinForms proyek desktop.
- Proyek desktop WPF.
- Proyek seluler Xamarin.
- Proyek ditulis dalam VB.NET.

Batasan

Untuk informasi selengkapnya tentang kuota dan batasan untuk AWS Transform, lihatKuota untuk AWS Transformasi.

AWS Transform tidak mengubah yang berikut:

- Komponen Blazor UI
- DLL Win32 yang tidak memiliki pustaka yang kompatibel dengan inti
- Repositori yang tidak mengandung solusi apa pun.

AWS Transform tidak akan memodifikasi cabang repo asli, dan hanya dapat menulis ke cabang target terpisah yang ditentukan dalam rencana transformasi Anda.

Intervensi manusia

Selama porting aplikasi.NET Framework ke .NET lintas platform, Anda mungkin diminta untuk memberikan masukan atau persetujuan dalam skenario berikut:

- Siapkan konektor ke kode sumber dan izin Anda
- Validasi rencana modernisasi yang diusulkan
- Unggah dependensi paket yang hilang sebagai NuGets
- Tinjau dan terima kode yang diubah

Informasi selengkapnya

Anda dapat memodernisasi kode .NET Anda dengan menggunakan aplikasi web AWS Transform atau AWS Toolkit untuk. Visual Studio

- Memodernisasi kode .NET Anda dengan menggunakan AWS Mengubah aplikasi web
- Modernisasi .NET di IDE
- Praktik terbaik untuk transformasi.NET

Memodernisasi kode .NET Anda dengan menggunakan AWS Mengubah aplikasi web

AWS Transform for .NET adalah AI-powered agen generatif baru yang dirancang untuk memodernisasi aplikasi .NET lama. Anda dapat memodernisasi kode .NET lama Anda dengan menggunakan aplikasi web AWS Transform atau ekstensi Toolkit. Visual Studio AWS

Untuk memodernisasi kode .NET Anda menggunakan aplikasi web AWS Transform, navigasikan ke aplikasi web ikuti langkah-langkah rinci berikut:

1. Menciptakan AWS Mengubah rencana kerja.NET
2. Membuat konektor repositori kode sumber
3. Mengonfirmasi repositori Anda untuk mempersiapkan transformasi
4. Menyelesaikan dependensi paket untuk mempersiapkan transformasi
5. Meninjau rencana Anda untuk mempersiapkan transformasi
6. Mengubah kode .NET Anda

Menciptakan AWS Mengubah rencana kerja.NET

Setelah Anda membuat ruang kerja Anda, pada tab Jobs, pilih Buat pekerjaan. Kemudian ikuti petunjuk dari AWS Transform di panel obrolan menggunakan bahasa alami. Ini adalah langkah-langkah khas untuk membuat pekerjaan modernisasi .NET.

1. AWS Transform akan menanyakan jenis pekerjaan transformasi yang ingin Anda buat. Dalam obrolan, masukkan modernisasi .NET.
2. AWS Transform akan menyarankan nama pekerjaan dan menanyakan apakah Anda ingin mengubah nama pekerjaan. Jika Anda ingin mengubah nama pekerjaan, beri tahu AWS Transform dalam bahasa alami, misalnya, ubah nama pekerjaan menjadi ExampleCorpDotNet1. Jika tidak, dalam obrolan, Anda dapat menerima nama pekerjaan yang disarankan. Setelah Anda menerima nama pekerjaan, AWS Transform memberi tahu Anda di jendela obrolan bahwa itu membuat pekerjaan.
3. AWS Transformasi menciptakan pekerjaan transformasi.

Komponen dari AWS Mengubah rencana kerja.NET

Tampilan pekerjaan AWS Transform .NET memiliki 5 tab yang Anda pilih dari ikon vertikal di paling kiri: Tugas, Dasbor, Persetujuan, Artefak, dan Worklog. Setiap tab memiliki panel kiri dan panel obrolan tengah. Panel kolaborasi yang tepat mungkin muncul sewaktu-waktu untuk menampilkan detail tambahan dan meminta masukan manusia.

Tugas (Job Plan)

Tab ini menampilkan rencana pekerjaan Anda di panel kiri. Pekerjaan.NET memiliki 5 langkah:

1. Dapatkan sumber daya untuk diubah: Pada fase ini, Anda membuat konektor ke repositori kode Anda menggunakan. AWS CodeConnections Bergantung pada izin repositori Anda, admin repositori kode mungkin perlu menyetujui konektor dan memberikan AWS akses Transform ke repositori.
2. Temukan sumber daya untuk transformasi: Pada fase ini, AWS Transform menemukan repositori dalam kontrol sumber, dan Anda memilih beberapa atau semuanya untuk penilaian.
3. Nilai kode untuk transformasi: Repositori yang dipilih dinilai, dan Anda dapat melihat laporan penilaian.
4. Bersiaplah untuk transformasi: Pada fase ini, AWS Transform memberi tahu Anda jika ada dependensi yang hilang dari repositori Anda. Anda dapat mengunggah dependensi yang hilang

atau mengabaikannya. Jika Anda bukan admin untuk repo, admin mungkin perlu menyetujui rencana transformasi akhir.

5. Transform: Pada fase ini, AWS Transform mengubah repo Anda dan memberi Anda status berkelanjutan selama transformasi hingga selesai. Anda dapat meninjau laporan transformasi untuk memahami apa yang diubah dan mengapa.

Anda dapat melihat status untuk setiap langkah:

- Tidak dimulai
- Tunggu masukan pengguna
- Sedang Berlangsung
- Selesai

Dasbor

Tab Dasbor memberikan ringkasan transformasi tingkat tinggi. Ini menampilkan metrik untuk jumlah pekerjaan yang diubah, transformasi yang diterapkan, dan perkiraan waktu untuk menyelesaikan transformasi. Di bawah dasbor adalah tabel repositori dan statusnya - In-progress, Gagal, atau Sukses.

Persetujuan

Permintaan persetujuan untuk pekerjaan ditampilkan dan diselesaikan pada tab ini.

Artifacts

Jjob-related artefak diunggah atau diunduh dari tab ini.

Worklog

AWS Transform log tindakannya di tab Worklog. Worklog menyediakan log terperinci tentang tindakan yang diambil AWS Transform, bersama dengan permintaan input manusia, dan tanggapan Anda terhadap permintaan tersebut.

Membuat konektor repositori kode sumber

Setelah ituMenciptakan AWS Mengubah rencana kerja.NET, pada tab Tugas, panel kiri jendela AWS Transform mencantumkan fase pekerjaan transformasi. Tahap pertama adalah Dapatkan sumber

daya untuk diubah. Pada fase ini, Anda dapat Mengundang kolaborator dan Connect repositori kode sumber. Panel kanan jendela AWS Transform adalah tempat Anda menentukan detailnya.

Setiap pekerjaan transformasi diharuskan hanya memiliki satu konektor repositori kode sumber. Agen AWS Transform menggunakan konektor ini untuk mengunduh basis kode.NET Anda dari GitHub, GitLab, atau Bitbucket dengan menggunakan. AWS CodeConnections Anda harus mengatur AWS CodeConnections di Wilayah yang sama dengan pekerjaan AWS Transform Anda. Atau, Anda dapat Connect to source code di Amazon S3 alih-alih repositori kode sumber.

> **Note**
>
> Jika Anda memiliki konektor repositori kode sumber yang ada, AWS Transform akan memberi tahu Anda. Pilih Gunakan konektor yang ada untuk menggunakan konektor ini. Jika Anda tidak ingin menggunakan konektor yang ada, lihat Menghapus konektor yang ada sebelumnyaMenambahkan konektor baru. Anda hanya dapat memiliki satu konektor repositori sumber per pekerjaan.

Menambahkan konektor baru

Untuk membuat konektor repositori kode sumber baru:

1. Masukkan nomor AWS akun yang ingin Anda gunakan untuk AWS CodeConnections konektor.
2. Jika Anda belum menyiapkan AWS akun AWS CodeConnections yang sama, Siapkan AWS CodeConnections.
3. Gunakan AWS Developer Tools Console untuk membuat koneksi ke Bitbucket, GitHub Enterprise Server GitHubGitLab, atau AzureDevOpsrepositori Anda.
4. Salin Nama Sumber Daya Amazon (ARN) dari koneksi yang Anda buat.
5. Kembali ke AWS Transform, dan paste ARN dari koneksi yang Anda buat.
6. Masukkan nama untuk konektor.

   > **Note**
   >
   > Jangan masukkan informasi pribadi sebagai bagian dari nama konektor.

7. Pilih Memulai pembuatan konektor.
8. AWS Transform membuat permintaan untuk membuat konektor untuk AWS akun yang Anda masukkan, di Wilayah yang sama dengan pekerjaan transformasi saat ini, dan memberi tahu

Anda bahwa permintaan persetujuan siap disetujui oleh AWS administrator Anda di Konsol AWS Manajemen. Pilih Salin tautan verifikasi.

9. Jika Anda adalah AWS administrator, masuk ke AWS akun dan buka tautan verifikasi untuk menyetujui permintaan koneksi. Jika Anda bukan AWS administrator, berikan tautan verifikasi ke AWS administrator Anda.
10. Setelah AWS administrator menyetujui permintaan konektor, pilih Selesaikan konektor untuk menyelesaikan proses pembuatan konektor. Jika Anda ingin menggunakan konektor yang berbeda, pilih Restart, untuk memulai kembali proses pembuatan konektor.

Tab Kolaborasi juga mencakup detail tentang konektor Anda, yang meliputi:

- Status - Persetujuan yang disetujui atau Tertunda
- AWS ID akun
- AWS Wilayah
- Koneksi ARN

Menghapus konektor yang ada

> **Note**
>
> Jika Anda memiliki konektor repositori kode sumber yang ada, AWS Transform akan memberi tahu Anda. Pilih salah satu Gunakan konektor yang ada atau Hapus dan buat konektor baru.

Jika Anda memilih untuk menghapus koneksi repositori kode sumber yang ada, AWS Transform akan memperingatkan Anda sebelum benar-benar menghapus konektor.

1. Anda dapat menghapus konektor yang ada beberapa cara:
   a. Jika AWS Transform meminta Anda untuk melakukannya, Anda dapat memilih Hapus dan membuat konektor baru.
   b. Anda juga dapat memilih restart di prompt, Untuk memodifikasi konektor ini, Anda harus memulai ulang.
2. AWS Transform memperingatkan Anda bahwa restart akan menghapus konektor. Untuk menghapus konektor, pilih Restart lagi.

3. AWS Transform memperingatkan Anda lagi bahwa menghapus konektor akan menghapus koneksi ke repositori pihak ketiga Anda, seperti Bitbucket,, dan. GitHub GitLab Juga, setiap pekerjaan AWS Transform yang menggunakan konektor ini akan gagal jika konektor dihapus. Untuk mengonfirmasi penghapusan, ketik hapus dan pilih Hapus.
4. Untuk menambahkan konektor repositori kode sumber baru, lihat. Menambahkan konektor baru

Connect ke kode sumber di Amazon S3

Anda dapat menghubungkan AWS Transform ke kode sumber di Amazon S3 sebagai alternatif untuk menghubungkan repo kode sumber.

Organisasi ember S3

Kode sumber asli dan kode yang diubah disimpan dalam ember S3 umum, diatur seperti yang ditunjukkan di bawah ini. Anda harus menyiapkan bucket S3 di wilayah yang sama dengan pekerjaan AWS Transform Anda. Unggah kode sumber Anda ke bucket sebagai file zip di tingkat root. File zip harus berisi folder tingkat atas untuk setiap repositori.

Selama transformasi, AWS Transform membuat folder transform-output, dan menyimpan hasil transformasi di folder itu. Transformasi membuat file zip bernama transformed-code.zip yang berisi kode yang diubah. Ini termasuk nama file laporan perbedaan kode diff.txt yang menyoroti perubahan file pada tingkat proyek.

```
<customer-bucket>/
### source-code.zip
         ### repo 1
         ### repo 2
         ### ......
         ### repo n
### transform-output/
      ### transformed-code.zip
```

Menambahkan Konektor S3 baru

Untuk membuat konektor repositori kode sumber S3 baru:

1. Di AWS konsol, buat ember S3.

2. Unggah kode sumber Anda sebagai file zip ke bucket S3.
3. Di aplikasi web AWS Transform, mulai pekerjaan transformasi.NET. Pada langkah Connect a source code repository, pilih Connect your source code in S3 bucket dan pilih Save.
4. Di halaman berikutnya, masukkan detail bucket S3 Anda dan pilih Kirim.
   a. Nama konektor
   b. AWS ID Akun
   c. S3 Bucket Arn
   d. Kunci Enkripsi Bucket S3 (opsional)
5. Menyetujui konektor:
   a. Di halaman berikutnya, salin tautan verifikasi.
   b. Mintalah pemberi persetujuan menelusuri tautan untuk mencapai halaman permintaan pembuatan konektor AWS Transform.
   c. Pilih untuk membuat peran baru atau menggunakan peran yang ada.
   d. Setelah meninjau permintaan, pilih Simpan dan Setujui untuk menyetujuinya.
6. Tentukan lokasi file zip S3:
   a. Di aplikasi web AWS Transform, tunggu status ditampilkan Disetujui.
   b. Pada halaman Tentukan lokasi aset, masukkan URL S3 untuk file zip kode dalam format `s3://bucket-name/zip-file-name` dan pilih Kirim ke AWS Transformasi.
   c. Pekerjaan berlanjut ke langkah Discovery untuk melanjutkan transformasi.

Menghapus Konektor S3

Untuk menghapus konektor S3 yang ada:

1. Di aplikasi web AWS Transform, pilih Kelola konektor di kanan atas.
2. Di jendela Kelola konektor, pilih konektor.
3. Pilih Hapus.

Mengonfirmasi repositori Anda untuk mempersiapkan transformasi

Setelah ituMembuat konektor repositori kode sumber, konfirmasikan repositori Anda sebagai langkah pertama mempersiapkan transformasi. Anda dapat:

- Tinjau dan ubah Pengaturan Default yang memengaruhi isi rencana.

- Tinjau repositori yang ditemukan dan pilih mana yang akan dinilai untuk transformasi.
- Tinjau laporan penilaian repositori dan ngobrol dengan AWS Transform tentang isinya.
- Gunakan rencana yang dihasilkan oleh AWS Transformasi.
- Sesuaikan rencana transformasimenggunakan aplikasi web atau dengan mengunduh file JSON, memodifikasinya, dan mengunggahnya ke Transform. AWS

Pengaturan Default

Di bagian Detail pekerjaan pada tab Kolaborasi, Anda dapat mengatur yang berikut ini:

Pada tab Tugas, di Siapkan untuk Transformasi - Konfirmasikan repositori untuk mengubah langkah, Anda dapat mengatur yang berikut:

- Tetapkan nama cabang target tempat kode yang diubah ditulis ke repositori Anda.
- Tetapkan target versi.NET:
  - .NET 8: ubah semua proyek menjadi .NET 8
  - .NET 10: ubah semua proyek menjadi .NET 10
- Anda dapat memperbarui Pengaturan Job default, atau membiarkannya apa adanya. Ini termasuk:
  - Kecualikan proyek .NET Standard dari rencana transformasi Pengaturan ini dipilih secara default. Ketika dipilih, AWS Transform mengecualikan setiap proyek .NET Standard dari rencana transformasi. Jika Anda membatalkan pilihan pengaturan ini, proyek .NET Standard akan diubah, yang akan membuat mereka tidak lagi kompatibel dengan .NET Framework.
  - Periksa NuGet sumbernya dan dapatkan versi paket.NET yang kompatibel dengan AWS Izinkan Transform untuk mencari dan dapatkan versi paket yang kompatibel dengan .NET untuk kode yang ingin Anda ubah.

Tinjau laporan penilaian repositori

Laporan penilaian repositori memberikan tampilan tabel informasi tentang setiap repositori yang terdeteksi oleh Transform. AWS Hal ini mencakup:

- Pemilik Repositori
- Cabang yang Dinilai
- Jumlah solusi
- Jumlah proyek

- Total Baris Kode
- Versi.NET Terdeteksi
- Jenis Proyek Terdeteksi
- Jumlah Dependensi Nuget Pribadi
- Jumlah Dependensi Nuget Publik
- Kompleksitas Transformasi

Untuk mengunduh file:

1. Di ruang kerja Anda, pilih tab Kolaborasi.
2. Di panel Pilih repositori, pilih Unduh, dan pilih Unduh sebagai HTML atau Unduh sebagai JSON

Mengobrol tentang laporan ringkasan repositori

Ringkasan repositori Anda mungkin sangat panjang. Gunakan obrolan untuk lebih memahami isinya. Laporan Anda tersedia untuk obrolan saat Anda melihat pesan ini di tab Worklog: Laporan penilaian sekarang tersedia di obrolan untuk kueri. Untuk membuka obrolan, klik ikon heksagonal ungu di sudut kanan bawah konsol web. Berikut adalah beberapa contoh petunjuk:

- Buat daftar semua repositori yang ditemukan atau dinilai.
- Berapa banyak.NET 6 proyek yang ditemukan dalam penilaian?
- Proyek mana yang memiliki kompleksitas tertinggi?

Gunakan rencana yang dihasilkan oleh AWS Transformasi

Anda dapat memilih untuk menggunakan paket AWS Transform yang dibuat dari menemukan repositori dengan probabilitas keberhasilan transformasi yang lebih tinggi, yang mencakup repositori terakhir yang dimodifikasi, root, dan cross-dependent.

AWS Transform mencantumkan beberapa detail tingkat tinggi tentang rencana transformasi, yang meliputi:

- Jumlah total repositori yang ditemukan Jumlah repositori .NET lama yang ditemukan AWS Transform menggunakan konektor repositori kode sumber Anda. AWS Transform dapat memindai maksimal 1.000 repositori di setiap pekerjaan. Jika Anda memiliki lebih banyak repositori untuk dipindai atau diubah, Anda dapat membuat beberapa pekerjaan transformasi.

- Repositori yang dipilih Jumlah repositori.NET lama yang AWS Transform dipilih untuk transformasi.
- Dependensi yang dipilih Jumlah dependensi yang diperlukan oleh repositori yang dipilih, yang AWS Transform otomatis terdeteksi dan ditambahkan ke rencana transformasi.

Tabel repositori yang dipilih mencantumkan repositori yang dipilih. Anda dapat mencari repositori berdasarkan nama, menavigasi ke halaman repositori tambahan, dan mengunduh file berformat JSON yang mencantumkan repositori yang dipilih dan dependensinya. Anda dapat membuat perubahan apa pun dalam tampilan ini.

Tabel Repositori yang dipilih mencantumkan rincian berikut tentang repositori yang dipilih Transform untuk AWS transformasi:

- Nama repositori
- Cabang sumber
- Tanggal dan waktu terakhir yang diubah
- Baris kode Ini memungkinkan Anda untuk melihat apakah Anda mendekati batas kuota Anda. Untuk informasi selengkapnya, lihat Kuota untuk AWS Transformasi.
- Repositori dependen Anda dapat mengklik jumlah repositori dependen untuk melihat detail lebih lanjut tentang dependensi ini.

Jika Anda puas dengan rencana transformasi yang dihasilkan AWS Transform, pilih Konfirmasi repositori.

Jika Anda ingin membuat perubahan pada rencana AWS transformasi yang dihasilkan Transform, pilihSesuaikan rencana transformasi.

Sesuaikan rencana transformasi

Jika Anda memilih untuk tidak Gunakan rencana yang dihasilkan oleh AWS Transformasi melakukannya, Anda dapat menyesuaikan daftar repositori untuk diubah menggunakan salah satu opsi berikut:

1. Pilih repositori di tabel Repositori yang dipilih.
   a. Saat Anda menambahkan repositori ke paket, AWS Transform akan memilih dependensi repositori untuk Anda dan secara otomatis menambahkannya ke rencana transformasi.
   b. Anda juga dapat membatalkan pilihan repositori dalam tabel.

2. Unduh daftar repo dan cabang JSON yang dihasilkan AWS Transform.
   a. Pilih Unduh dan kemudian pilih Unduh sebagai JSON.
   b. Tinjau dan modifikasi JSON. Anda dapat menghapus repositori, atau solusi dalam repositori, yang tidak ingin Anda ubah.
   c. Unggah JSON yang direvisi dengan memilih Unggah paket yang disesuaikan.
   d. Jika JSON berhasil diunggah, tabel Repositori yang dipilih akan menampilkan pilihan yang Anda buat.
3. Anda dapat memilih Tampilkan repo dependen untuk menampilkan daftar repositori yang ditemukan AWS Transform adalah dependensi untuk repositori yang Anda pilih.
4. Anda dapat mengulangi langkah-langkah ini jika diperlukan. Ketika Anda puas dengan paket khusus Anda, pilih Konfirmasi repositori.

Menyelesaikan dependensi paket untuk mempersiapkan transformasi

Setelah ituMengonfirmasi repositori Anda untuk mempersiapkan transformasi, jika AWS Transform menemukan dependensi paket yang hilang, Anda harus menyelesaikan langkah ini. Anda dapat menjalankan PowerShell skrip Windows untuk mendapatkan dependensi paket yang hilang dari perangkat yang sama dengan lingkungan Visual Studio pengembangan Anda, atau Anda dapat mengambil paket yang hilang secara manual. Kemudian, unggah paket yang hilang.

Dependensi Paket yang Hilang dapat diperbarui dengan dua cara:

- Selesaikan menggunakan Konektor Artifact: Selesaikan dengan mengonfigurasi Repositori Artifact (Konektor ADO) NuGet melalui alur kerja HITL Connect Artifact Repository khusus, yang ditampilkan saat Konektor Artifact tidak dikonfigurasi dan paket tidak ada.
- Perbarui paket yang hilang: AWS Transform mencantumkan paket yang hilang di tabel dependensi paket Hilang. Anda dapat mencari paket yang hilang berdasarkan nama di kotak pencarian. Tabel ini mencakup rincian berikut tentang paket yang hilang:
  - Nama
  - Repositori terkait
  - Status versi kerangka kerja
  - Status versi inti

Untuk mengatasi dependensi paket yang hilang,. Unggah paket yang hilang

Unggah paket yang hilang

1. Jika Anda memilih, Anda dapat mengunduh skrip PowerShell pembantu Windows untuk mengambil dependensi paket yang hilang dari dalam lingkungan pengembangan Anda. Visual Studio Atau Anda dapat menemukan paket yang hilang secara manual.

   Untuk menggunakan PowerShell skrip Windows:

   a. Pilih Unduh PowerShell skrip Windows.

   b. Jalankan skrip secara lokal dengan koneksi aktif ke repositori yang berisi file ketergantungan paket yang hilang.

   c. Skrip ini memungkinkan Anda mengunduh dependensi paket yang hilang ke lingkungan lokal Anda.

2. Script akan membuat file zip tunggal untuk Anda unggah yang mencakup semua dependensi dalam satu arsip. Anda juga dapat mengunggah file individual `.nupkg` atau zip untuk setiap dependensi.

3. Pilih Unggah file paket.

4. Dalam modal Unggah file dependensi, pilih Pilih file dan telusuri ke lokasi file paket yang hilang terkompresi di perangkat Anda.

5. Pilih Unggah.

6. AWS Transform memvalidasi file yang Anda unggah. Selama validasi, Anda tidak dapat melakukan pembaruan apa pun. AWS Transform melaporkan status validasi di atas tabel dependensi paket Hilang.

7. AWS Transform juga memperbarui kolom status dalam tabel dependensi paket Hilang dari Hilang ke Terselesaikan. Jika paket gagal validasi, statusnya menjadi Tidak Valid. Untuk file yang tidak valid lakukan hal berikut:

   a. Dalam tabel dependensi paket Hilang, pilih paket yang tidak valid menggunakan kotak centang.

   b. Pilih Hapus file yang diunggah.

   c. Ini mengubah statusnya kembali ke Missing.

8. Setelah Anda mengunggah paket yang hilang dan menyelesaikan dependensi paket, pilih Lanjutkan untuk meninjau.

   Jika Anda memilih Lanjutkan untuk meninjau tanpa menyelesaikan dependensi paket yang hilang, AWS Transform menanyakan apakah Anda ingin memulai pekerjaan transformasi tanpa paket yang hilang. Jika Anda memilih Abaikan dependensi paket yang hilang, AWS Transform

akan menggunakan referensi assembly untuk mengubah kode. Melanjutkan tindakan ini dapat memengaruhi sumber daya terkait.

Persyaratan Struktur Folder

Untuk NuGet upload AWS Transform membutuhkan:

- `.nuspec`File yang akan ditempatkan di tingkat root untuk paket yang diunggah secara manual, atau
- Penggunaan PowerShell skrip yang disediakan untuk menghasilkan struktur yang benar

Persyaratan pencocokan versi

- Status kerangka kerja: AWS Transform mencoba untuk mencocokkan versi dependensi yang ditentukan dalam `.csproj` file (kode sumber) dengan versi dependensi yang diunggah. Pencocokan yang tepat menghasilkan status Sukses.
- Status inti: AWS Transform mengharapkan spesifikasi versi inti di bagian dependensi dari file yang diunggah. `.nuspec` Jika ditemukan maka status intinya adalah Sukses.

  Contoh format:

  ```
  <dependencies>
      <group targetFramework="net8.0" />
  </dependencies>
  ```

  Versi umum meliputi:

  - .NET 5.0 (net5.0)
  - .NET 6.0 (net6.0)
  - .NET 7.0 (net7.0)
  - .NET 8.0 (net8.0)

Meninjau rencana Anda untuk mempersiapkan transformasi

SetelahMengonfirmasi repositori Anda untuk mempersiapkan transformasi, danMenyelesaikan dependensi paket untuk mempersiapkan transformasi, administrator AWS akun harus meninjau rencana transformasi dan menyetujuinya di AWS Transform.

AWS Transform menampilkan daftar pekerjaan repositori, repositori dependen, dan paket dependen yang dipilih untuk transformasi.

Meninjau rencana transformasi

AWS Transform menampilkan daftar pekerjaan repositori, repositori dependen, dan paket dependen yang dipilih untuk transformasi.

> **Note**
>
> AWS Transform dapat mengubah maksimum 100 dependensi dan repositori per rencana transformasi.

1. Jika Anda bukan administrator AWS akun, tinjau rencana pekerjaan, dan jika Anda menerimanya, pilih Kirim untuk persetujuan.
2. Jika Anda adalah administrator AWS akun, Anda harus meninjau rencana, dan ketika siap, setujui rencana untuk memulai transformasi. Setelah Anda meninjau rencana kerja, pilih salah satu:
   a. Tolak Jika pekerjaan dibuat oleh pengguna yang bukan administrator AWS akun, kami sarankan Anda memberi tahu pembuat pekerjaan untuk memulai kembali pekerjaan.
   b. Menyetujui dan memulai transformasi.

Tinjauan pekerjaan mencakup detail berikut:

1. Ringkasan Job Ini termasuk:
   a. Cabang target tempat AWS Transform akan menempatkan kode yang diubah.
   b. Target versi.NET, .NET 8.0 atau .NET 10.
   c. Pengaturan pekerjaan:
      i. Kecualikan proyek standar.NET
   d. Jumlah repositori yang dipilih untuk transformasi
   e. Jumlah repositori dependen
   f. Jumlah NuGet paket pribadi
   g. Total baris kode untuk pekerjaan
2. Repositori dipilih Ini adalah repositori yang dipilih untuk transformasi. Mereka harus berupa MVC, Web, Windows Communication Foundation (WCF), Konsol, perpustakaan kelas, kerangka kerja UI - Halaman pisau cukur, atau paket pengujian unit. Tabel ini mencakup informasi berikut:

   a. Nama
   b. Cabang sumber
   c. Proyek yang didukung
   d. Baris kode
   e. Proyek terdeteksi
   f. Proyek dilewati
   g. Dependensi terdeteksi
3. Repositori dependen ditambahkan Ini adalah repositori dependen yang ditambahkan untuk transformasi. Mereka harus berupa MVC, Web, Windows Communication Foundation (WCF), Konsol, perpustakaan kelas, kerangka kerja UI - Halaman pisau cukur, atau paket pengujian unit. Tabel ini mencakup informasi berikut:
   a. Nama
   b. Dibutuhkan oleh
   c. Cabang sumber
   d. Proyek yang didukung
   e. Baris kode
   f. Proyek terdeteksi
   g. Proyek dilewati
4. Paket dependen Ini adalah paket dependen yang ditambahkan untuk transformasi. Mereka harus berupa MVC, Web, Windows Communication Foundation (WCF), Konsol, perpustakaan kelas, kerangka kerja UI - Halaman pisau cukur, atau paket pengujian unit. Tabel ini mencakup informasi berikut:
   a. Nama
   b. Repositori terkait
   c. Status versi kerangka kerja
   d. Status versi inti

Mengubah kode .NET Anda

Setelah administrator AWS akun selesaiMeninjau rencana Anda untuk mempersiapkan transformasi, Anda dapat melihat kemajuan transformasi di tab Dasbor pekerjaan transformasi Anda.

> **Note**
>
> Selain mengubah repositori dan dependensi, AWS Transform dapat mengeksekusi proyek pengujian unit yang sepenuhnya ditransformasikan (nol kesalahan build). AWS Transform tidak memiliki akses ke hasil pengujian unit sebelum transformasi dan hanya dapat membagikan hasil pasca-transformasi. Anda kemudian dapat membandingkan data dasar Anda, sebelum transformasi, dengan hasil pasca-transformasi untuk memahami potensi kesenjangan.

Di sudut kanan atas, Anda dapat melihat status pekerjaan, yang memiliki salah satu nilai berikut:

- Menunggu masukan pengguna
- Waktu berlalu
- Berjalan

Anda juga dapat melihat ikon berikut:

- Ikon berhenti transformasi
- Ikon penyegaran
- Ikon pengaturan

Dasbor memberikan ringkasan transformasi tingkat tinggi. Ini menunjukkan metrik untuk jumlah pekerjaan yang diubah dan transformasi diterapkan, dan perkiraan waktu untuk menyelesaikan transformasi.

Dasbor meliputi:

1. Bagian Transformation Job details, yang mencantumkan pengaturan default dan detail pekerjaan transformasi, termasuk:
   a. Target tujuan cabang Untuk mengubah kode Anda, AWS Transform membuat cabang baru untuk kode yang diubah dalam repo kode Anda.
   b. Target versi.NET, .NET 8.0 atau .NET 10
   c. ID pekerjaan AWS Transform.
   d. Pengaturan pekerjaan:
      i. Kecualikan proyek standar.NET

2. Bagian ringkasan Transformasi berisi:
   a. Jumlah repositori yang dipilih untuk transformasi
   b. Jumlah proyek yang akan diubah
   c. Total baris kode dalam repositori dan proyek ini

Setelah transformasi dimulai, diagram lingkaran muncul di bagian status Repositori, status Package, dan Status pengujian Unit yang menampilkan kemajuan secara real time. Status pengujian Unit menunjukkan status pengujian unit yang terletak di repositori Anda yang AWS Transforms berjalan setelah transformasi untuk menguji kode yang diubah. AWS Transform membagikan hasil pengujian yang dieksekusi, bersama dengan nama pengujian individual bagi pelanggan untuk meninjau daftar pengujian unit yang lulus dan gagal.

> **Note**
>
> Bagian Repositori mencantumkan repositori yang direkomendasikan AWS Transform atau yang Anda pilih untuk transformasi. Pilih Unduh JSON untuk mengunduh daftar repositori, dependensi, dan paket dalam rencana transformasi Anda.

Tinjau laporan transformasi

Setelah pekerjaan transformasi selesai, Anda dapat mengunduh Laporan Transformasi:

- Laporan ringkasan transformasi: Laporan ini memberikan gambaran umum tentang transformasi dalam format HTML atau JSON. Unduh laporan ringkasan transformasi dari tab Dasbor. Pilih Unduh, dan pilih Unduh sebagai HTML atau Unduh sebagai JSON.
- Laporan detail transformasi: Laporan ini memberikan informasi mendalam tentang transformasi dalam format HTML. Unduh laporan detail transformasi dari tab Dasbor. Pilih tautan Unduh laporan terperinci di daftar repositori untuk mengunduh laporan detail.

Mengobrol dengan AWS Transformasi tentang laporan transformasi

Anda dapat mengobrol dengan AWS Transform tentang laporan transformasi setelah repositori diproses sepenuhnya atau setelah pekerjaan transformasi selesai. Tab Worklog menampilkan pesan ini untuk setiap repositori yang tersedia untuk obrolan: Detail *`repository_name`* transformasi repositori sekarang tersedia di obrolan untuk kueri. Ini menampilkan pesan ini ketika pekerjaan transformasi selesai: AWS Transformasi pekerjaan transformasi selesai.

Untuk membuka obrolan, klik ikon heksagonal ungu di sudut kanan bawah konsol web.

Berikut adalah beberapa contoh petunjuk:

- Proyek mana yang berhasil diubah?
- Proyek mana yang sebagian porting?
- Perubahan apa yang dilakukan pada `repo_name` repositori?
- Paket apa yang ditingkatkan dalam `project_name` proyek?

Modernisasi .NET di IDE

AWS Transform mencakup asisten AI agen interaktif di AWS Toolkit untuk. Visual Studio Ini memungkinkan Anda untuk memodernisasi aplikasi melalui pengalaman percakapan, langkah demi langkah yang dipandu langsung di IDE Anda. Untuk mengubah solusi atau proyek .NET, agen AWS Transform menganalisis basis kode Anda, menentukan pembaruan yang diperlukan untuk mem-port aplikasi Anda, dan menghasilkan laporan penilaian dan rencana transformasi sebelum transformasi dimulai.

Pengalaman Visual Studio IDE hanya mendukung otentikasi oleh IAM Identity Center. Untuk informasi selengkapnya, lihat Menambahkan pengguna di Pusat Identitas IAM.

Pengalaman IDE menawarkan mode otonom dan interaktif. Gunakan mode otonom untuk transformasi tanpa pengawasan, cocok untuk pekerjaan semalam atau jangka panjang. Gunakan mode interaktif untuk bekerja bersama agen dalam obrolan, dengan visibilitas tinggi ke dalam transformasi, rencana transformasi yang dapat disesuaikan, dan tinjauan langkah demi langkah dari setiap proyek. Anda dapat mengulangi untuk menyempurnakan hasil atau mengubah arah hingga puas.

Saat pekerjaan transformasi aktif, Anda dapat berinteraksi dengan tiga jendela:

- Jendela Obrolan adalah titik interaksi utama Anda dengan agen transformasi. Gunakan obrolan untuk mengajukan pertanyaan, mendiskusikan opsi, mengarahkan transformasi, dan mengulangi.
- Jendela AWS Transform Job Plan menunjukkan langkah-langkah kemajuan dan di mana Anda dapat mengunduh atau mengunggah artefak atau melihat perbedaan perubahan kode.
- Jendela Worklog menunjukkan log aktivitas transformasi dalam bahasa alami.

AWS Transform melakukan empat tugas utama untuk mem-port aplikasi.NET ke Linux:

- Upgrade .NET dan versi bahasa - Upgrade kode .NET C# yang lebih lama ke versi .NET dan bahasa saat ini.
- Bermigrasi dari .NET Framework ke lintas platform .NET - Migrasi proyek dan paket dari .NET Framework yang bergantung pada Windows ke lintas platform .NET yang kompatibel dengan Linux.
- Menulis ulang kode untuk kompatibilitas Linux — Memfaktorkan ulang dan menulis ulang komponen kode yang tidak digunakan lagi dan tidak efisien.
- Menghasilkan laporan kesiapan kompatibilitas Linux — Untuk tugas terbuka di mana intervensi pengguna diperlukan untuk membuat kode dibangun dan dijalankan di Linux, AWS Transform menyediakan laporan rinci tindakan yang diperlukan untuk mengonfigurasi aplikasi Anda setelah transformasi.

Untuk informasi selengkapnya tentang cara AWS Transform melakukan transformasi.NET, lihatBagaimana AWS Transform memodernisasi aplikasi.NET.

Untuk memodernisasi kode .NET Anda menggunakan AWS Transform dari Visual Studio IDE, lihat yang berikut ini:

- Modernisasi .NET menggunakan AWS Transformasi di Studio Visual
- Bagaimana AWS Transform memodernisasi aplikasi.NET
- Memecahkan masalah dengan transformasi.NET di IDE

Kuota

Untuk AWS kuota transformasi Transformasi.NET di IDE, lihatKuota untuk AWS Transformasi.

Modernisasi .NET menggunakan AWS Transformasi di Studio Visual

Selesaikan langkah-langkah ini untuk mem-port Windows-based aplikasi.NET ke aplikasi .NET Linux-compatible lintas platform dengan AWS Transform pada Visual Studio tahun 2026 atau 2022.

Langkah 1: Prasyarat

Sebelum melanjutkan, pastikan Anda telah mengunduh dan menginstal ekstensi AWS Toolkit dengan Amazon Q dari Visual Studio Marketplace. Tinjau kemampuan dan batasan AWS Transform untuk modernisasi .NET untuk mengonfirmasi bahwa kode Anda dapat diubah.

Langkah 2: Otentikasi di Visual Studio

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

1. Dari menu utama Visual Studio, navigasikan ke Extensions **>** AWS Toolkit**>** Memulai untuk membuka halaman Memulai: AWS Toolkit.
2. Pilih AWS Transform authentication.
3. Pilih profil yang sudah ada atau buat yang baru. Saat membuat profil baru, masukkan nama untuk profil dan URL awal AWS Transform Anda, yang dikirim melalui email kepada Anda setelah Anda ditambahkan ke Pusat Identitas IAM oleh administrator Anda.
4. Pilih Hubungkan. Ikuti petunjuk untuk masuk dan berikan izin akses melalui browser web Anda. Di Visual Studio, Anda akan melihat Connected with IAM Identity Center.
5. Setelah masuk, Anda akan dibawa ke Dasbor AWS Transform. Dari halaman arahan ini, Anda dapat melakukan hal berikut:

   - Pilih profil AWS Transform (wilayah)
   - Pilih atau buat ruang kerja
   - Lihat status pekerjaan transformasi

Langkah 3: Ubah aplikasi Anda

Untuk mengubah solusi atau proyek .NET Anda, selesaikan prosedur berikut:

Mulai transformasi

1. Buka C# atau solusi atau proyek VB.NET berbasis apa pun Visual Studio yang ingin Anda ubah.
2. Dari Solution Explorer, klik kanan solusi atau proyek yang ingin Anda ubah, dan pilih Port dengan AWS Transform.
3. Dialog Port with AWS Transform muncul.

   a. Dalam menu tarik-turun Pilih ruang kerja, buat ruang kerja atau pilih yang sudah ada. Ruang kerja membuat aktivitas Anda terlihat di aplikasi web AWS Transform, tempat Anda dapat mengundang kolaborator untuk melihat dan mengobrol tentang pekerjaan transformasi.

   b. Di menu tarik-turun Pilih target.NET, pilih versi.NET yang ingin Anda tingkatkan.

   c. Untuk Mode Transformasi, pilih Otonom untuk transformasi tanpa pengawasan atau Interaktif untuk transformasi interaktif.

d. Pilih Mulai untuk memulai transformasi.

4. Setelah pekerjaan transformasi dimulai, jendela AWS Transform Job Plan, Chat, dan Worklog muncul.

Rencana penilaian dan transformasi

1. Setelah AWS Transform menganalisis kode Anda, laporan penilaian dan rencana transformasi ditampilkan dalam obrolan. Dalam mode interaktif, obrolan akan berhenti sejenak bagi Anda untuk memeriksanya. Jika Anda memilih untuk tidak meninjau dalam obrolan, Anda dapat menggunakan alternatif ini:

   - Untuk melihat penilaian atau rencana di editor kode, pilih Tampilkan penilaian atau Tampilkan rencana transformasi di jendela AWS Transform Job Plan.
   - Untuk mengunduh penilaian atau rencana, gunakan ikon unduhan di jendela Rencana Pekerjaan.

2. Secara opsional memodifikasi rencana transformasi. Anda dapat mendiskusikan rencana dengan obrolan, seperti menjelajahi opsi lain. Anda membuat perubahan rencana dengan memberikan instruksi dalam obrolan, atau dengan mengunggah rencana transformasi yang direvisi menggunakan ikon unduhan di jendela Rencana Pekerjaan. Anda dapat dengan bebas menambahkan, menghapus, atau memodifikasi langkah-langkah. Obrolan akan mengkonfirmasi pemahamannya tentang instruksi Anda dengan rencana yang direvisi.
3. Opsional mengunggah dokumen kemudi. Di jendela obrolan, gunakan ikon unggah untuk mengunggah dokumen kemudi seperti spesifikasi aplikasi, preferensi dan konvensi tumpukan teknologi organisasi, atau contoh pengkodean. Setelah mengunggah dokumen kemudi, jelaskan di jendela obrolan untuk memberikan konteks agen.
4. Setelah Anda puas dengan rencananya, berikan persetujuan dalam obrolan dan transformasi akan dimulai.
5. Dalam mode interaktif, Anda dapat mengatur pos pemeriksaan untuk berhenti setelah proyek diubah untuk ditinjau. Langkah-langkah di bawah Transforming code di jendela Job Plan akan memiliki ikon pos pemeriksaan bulat. Semua proyek memiliki pos pemeriksaan yang diaktifkan secara default. Gunakan kotak centang untuk menghapus atau mengatur pos pemeriksaan.

Ulasan transformasi dan pos pemeriksaan

1. AWS Transform mulai mengubah kode Anda. Anda dapat memantau kemajuan langkah transformasi di jendela AWS Transform Job Plan. Jendela Job Plan menampilkan perkiraan waktu maksimum yang tersisa dan ID Job Anda.
2. Saat proyek diubah, ringkasan ditampilkan dalam obrolan dan perubahan kode diterapkan.
3. Dalam mode interaktif, ketika pos pemeriksaan tercapai, agen akan berhenti sejenak sehingga Anda dapat meninjau hasilnya. Di jendela Job Plan, Anda akan melihat tombol Lihat Hasil untuk langkah yang dijeda di mana Anda dapat meninjau perbedaan kode. Setelah meninjau perubahan, Anda dapat meminta perubahan lebih lanjut dalam obrolan atau membuat perubahan kode sendiri. Visual Studio
4. Anda dapat mengulangi di pos pemeriksaan sebanyak yang diinginkan. Ketika Anda puas, instruksikan obrolan untuk melanjutkan, dan itu akan beralih ke proyek berikutnya.

Akhir transformasi

1. Setelah mengubah proyek Anda, agen.NET akan memicu build in lokalVisual Studio, untuk mengonfirmasi bahwa kode berhasil dibangun di mesin lokal Anda. Jika ada kesalahan, itu akan berhasil untuk menyelesaikannya.
2. Setelah transformasi selesai, mungkin ada langkah-langkah tambahan yang diperlukan - seperti tugas transformasi yang tidak dapat dilakukan AWS Transform, atau perubahan yang diperlukan untuk kompatibilitas Linux. File penurunan harga langkah selanjutnya (NextSteps.md) tersedia untuk diunduh yang dapat Anda gunakan sebagai petunjuk ke pendamping kode AI. Gunakan tombol ikon unduhan jendela Job Plan untuk mengunduh NextSteps.md.
3. Anda dapat terus berinteraksi dalam obrolan untuk mengajukan pertanyaan atau meminta perubahan tambahan.
4. Beri tahu obrolan saat Anda siap untuk mengakhiri pekerjaan transformasi Anda.

Bagaimana AWS Transform memodernisasi aplikasi.NET

Tinjau bagian berikut untuk detail tentang cara kerja transformasi.NET dengan AWS Transform.

Menganalisis aplikasi Anda dan menghasilkan rencana transformasi

Sebelum transformasi dimulai, AWS Transform membangun kode Anda secara lokal untuk memverifikasi bahwa itu dapat dibangun dan dikonfigurasi dengan benar untuk transformasi. Jika

kode tidak dibangun, AWS Transform akan meminta apakah akan melanjutkan transformasi. AWS Transform kemudian mengunggah kode Anda ke lingkungan build yang aman dan terenkripsi AWS, menganalisis basis kode Anda, dan menentukan pembaruan yang diperlukan untuk mem-port aplikasi Anda

Selama analisis ini, AWS Transform membagi solusi atau proyek .NET Anda ke dalam kelompok kode. Grup kode adalah proyek dan semua dependensinya yang bersama-sama menghasilkan unit kode yang dapat dibangun seperti pustaka tautan dinamis (DLL) atau yang dapat dieksekusi. Bahkan jika Anda tidak memilih semua dependensi proyek untuk diubah, AWS Transform menentukan dependensi yang diperlukan untuk membangun proyek yang Anda pilih dan mengubahnya juga, sehingga aplikasi Anda yang diubah akan dapat dibangun dan siap digunakan.

Setelah menganalisis kode Anda, AWS Transform menghasilkan rencana transformasi yang menguraikan perubahan yang diusulkan yang akan dibuat, termasuk daftar grup kode dan dependensi mereka yang akan diubah.

Mengubah aplikasi Anda

Untuk memulai transformasi, AWS Transform membuat kode Anda lagi di lingkungan build yang aman untuk memverifikasi apakah kode tersebut dapat dibangun dari jarak jauh. AWS Transform kemudian mulai porting aplikasi Anda. Ini bekerja dari bawah ke atas, dimulai dengan ketergantungan tingkat terendah. Jika AWS Transform mengalami masalah dengan porting dependensi, Transform menghentikan transformasi dan memberikan informasi tentang apa yang menyebabkan kesalahan.

Transformasi mencakup pembaruan berikut untuk aplikasi Anda:

- Mengganti versi kode C # yang sudah ketinggalan zaman dengan Linux-compatible versi C#
- Meningkatkan .NET Framework ke .NET lintas platform, termasuk:
  - Mengidentifikasi dan mengganti paket, pustaka, dan API secara berulang
  - Memutakhirkan dan mengganti NuGet paket dan API
  - Transisi ke runtime lintas platform
  - Menyiapkan middleware dan memperbarui konfigurasi runtime
  - Mengganti paket pribadi atau pihak ketiga
  - Menangani komponen IIS dan WCF
  - Kesalahan pembuatan debug
- Menulis ulang kode untuk kompatibilitas Linux, termasuk refactoring dan penulisan ulang kode usang dan tidak efisien untuk mem-port kode yang ada

Meninjau laporan transformasi dan menerima perubahan

Setelah transformasi selesai, AWS Transform menyediakan laporan transformasi dengan informasi tentang usulan pembaruan yang dibuat untuk aplikasi Anda, termasuk jumlah file yang diubah, paket diperbarui, dan API diubah. Ini menandai transformasi yang gagal, termasuk file atau bagian file yang terpengaruh dan kesalahan yang ditemui selama percobaan build. Anda juga dapat melihat ringkasan build dengan log build untuk mempelajari lebih lanjut tentang perubahan yang dibuat.

Laporan transformasi juga menyediakan status porting Linux, yang menunjukkan apakah input pengguna tambahan diperlukan atau tidak untuk membuat aplikasi Linux kompatibel. Jika salah satu item dalam grup kode memerlukan masukan dari Anda, Anda mengunduh laporan kesiapan Linux yang berisi Windows-specific pertimbangan yang tidak dapat ditangani AWS Transform pada waktu pembuatan. Jika input diperlukan untuk grup kode atau file apa pun, tinjau laporan untuk detail tentang jenis perubahan apa yang masih perlu dilakukan dan, jika berlaku, untuk rekomendasi tentang cara memperbarui kode Anda. Perubahan ini harus dilakukan secara manual sebelum aplikasi Anda dapat dijalankan di Linux.

Anda dapat meninjau perubahan yang diusulkan AWS Transform yang dibuat dalam tampilan diff sebelum menerimanya sebagai pembaruan di tempat untuk file Anda. Setelah memperbarui file Anda dan menangani item apa pun dalam laporan kesiapan Linux, aplikasi Anda siap dijalankan di .NET lintas platform.

Anda dapat mengunduh file penurunan harga Langkah Berikutnya dengan petunjuk untuk transformasi lanjutan dengan AWS Transform atau pendamping kode AI. Untuk mengunduh file, pilih tombol Unduh Langkah Berikutnya di kanan atas laporan.

Memecahkan masalah dengan transformasi.NET di IDE

Gunakan bagian berikut untuk memecahkan masalah umum dengan transformasi.NET di IDE dengan Transform. AWS

Bagaimana saya tahu jika suatu pekerjaan sedang berkembang?

Anda akan melihat aktivitas reguler di jendela Worklog. Jika AWS Transform tampaknya menghabiskan waktu lama pada langkah di jendela AWS Transform Job Plan, Anda dapat memeriksa apakah pekerjaan tersebut masih aktif di log keluaran. Jika pesan diagnostik sedang dibuat, pekerjaan masih aktif.

Untuk memeriksa output, pilih tab Output diVisual Studio. Di menu Tampilkan output dari:, pilih Amazon Q Language Client.

Tangkapan layar berikut menunjukkan contoh output yang dihasilkan AWS Transform selama transformasi.

Mengapa beberapa proyek tidak dipilih untuk transformasi?

AWS Transform hanya dapat mengubah jenis proyek.NET yang didukung. Untuk daftar jenis proyek yang didukung dan prasyarat lain untuk mengubah proyek .NET Anda, lihat. Langkah 1: Prasyarat

Bagaimana saya bisa mendapatkan dukungan jika proyek atau solusi saya tidak berubah?

Jika Anda tidak dapat memecahkan masalah sendiri, Anda dapat menghubungi Dukungan atau Akun AWS tim Anda untuk mengirimkan kasus dukungan.

Untuk mendapatkan dukungan, berikan ID pekerjaan transformasi sehingga AWS dapat menyelidiki pekerjaan yang gagal. Untuk menemukan ID pekerjaan transformasi, pilih tab Output diVisual Studio. Di menu Tampilkan output dari:, pilih Amazon Q Language Client.

Bagaimana saya bisa mencegah firewall saya mengganggu pekerjaan transformasi?

Jika organisasi Anda menggunakan firewall, itu mungkin mengganggu transformasi diVisual Studio. Anda dapat menonaktifkan sementara pemeriksaan keamanan Node.js untuk memecahkan masalah atau menguji apa yang mencegah transformasi berjalan.

Variabel lingkungan `NODE_TLS_REJECT_UNAUTHORIZED` mengontrol pemeriksaan keamanan penting. Pengaturan `NODE_TLS_REJECT_UNAUTHORIZED` ke “0" menonaktifkan penolakan Node.js terhadap sertifikat yang tidak TLS/SSL sah. Ini berarti:

- Self-signed sertifikat akan diterima
- Sertifikat kedaluwarsa akan diizinkan
- Sertifikat dengan nama host yang tidak cocok akan diizinkan
- Kesalahan validasi sertifikat lainnya akan diabaikan

Jika proxy Anda menggunakan sertifikat mandiri, Anda dapat mengatur variabel lingkungan berikut alih-alih `NODE_TLS_REJECT_UNAUTHORIZED` menonaktifkan:

```
NODE_OPTIONS = –use-openssl-ca
NODE_EXTRA_CA_CERTS = Path/To/Corporate/Certs
```

Jika tidak, Anda harus menentukan sertifikat CA yang digunakan oleh proxy untuk menonaktifkan`NODE_TLS_REJECT_UNAUTHORIZED`.

Untuk menonaktifkan NODE_TLS_REJECT_UNAUTHORIZED di Windows:

1. Buka menu Start dan cari Variabel Lingkungan.
2. Pilih Edit variabel lingkungan sistem.
3. Di jendela System Properties, pilih Environment Variables.
4. Di bawah Variabel sistem, pilih Baru.
5. Setel nama Variabel ke NODE_TLS_REJECT_UNAUTHORIZED dan nilai Variabel ke 0.
6. Pilih OK untuk menyimpan perubahan.
7. Mulai ulang Visual Studio.

Porting kode lapisan UI

AWS Transform dapat mem-port lapisan UI ASP.NET situs web ke kerangka kerja UI yang kompatibel dengan ASP.NET Core, yang dapat berjalan di Linux. Transformasi berikut tersedia:

- Porting MVC Razor UI
- Formulir Web ke porting Blazor UI

Porting MVC Razor UI

ASP.NET Proyek MVC dengan Razor Views UI dapat di-porting ke MVC Razor Views di ASP.NET Core, tetap pada kerangka UI yang sama. Meskipun tampilan Razor serupa antara ASP.NET dan ASP.NET Core, ada perbedaan yang memerlukan perubahan kode porting:

- ASP.NET MVC Razor Views menggunakan System.Web.Mvc namespace dan terkait erat dengan .NET Framework.
- ASP.NET Core Razor Views bersifat modular, dibangun di atas Microsoft.AspNetCore.Mvc, dengan injeksi ketergantungan yang ditingkatkan.
- ASP.NET Core memiliki fitur baru termasuk Tag Helpers dan View Components.

AWS Transform mendeteksi dan mem-port komponen dan konstruksi umum yang tidak kompatibel di Razor Views.

Formulir Web ke porting Blazor UI

ASP.NET UI Formulir Web tidak memiliki mitra di ASP.NET Core. AWS Transform dapat mem-port kode lapisan UI Formulir Web ke Blazor Server di ASP.NET Core. Kode lapisan UI ditulis ulang untuk kerangka UI target.

Untuk mengaktifkan atau menonaktifkan porting Formulir Web ke Blazor UI, periksa atau hapus pengaturan pekerjaan Transform UI ASP.NET Web Forms to Blazor.

Detail transformasi

Proyek Formulir Web Anda dikonversi menjadi proyek Blazor dengan hosting sisi server. Perubahan proyek dan file berikut dibuat selama transformasi:

Pemetaan transformasi file

Dari	Ke	Deskripsi
*.aspx, *.ascx	*.pisau cukur	halaman.aspx dan kontrol kustom.ascx menjadi file.razor
Web.config	appsettings.json	Web.config pengaturan menjadi pengaturan appsettings.json
Global.asax	Program.cs	Global.asax kode menjadi Program.cs kode
*.master	*layout.razor	File master menjadi tata letak. File razor

Post-transformation struktur proyek

Setelah transformasi, file yang diubah berada di lokasi proyek berikut:

- Folder komponen: Semua komponen pisau cukur (*.razor) ditempatkan di folder Komponen yang dibuat pada tingkat yang sama dengan file.csproj proyek.
- Pemetaan halaman: Halaman Formulir Web (*.aspx) dikonversi ke struktur folder. Components/Pages

- Kontrol pemetaan: Kontrol Formulir Web (*.ascx) ditempatkan langsung di bawah folder Komponen.

Keterbatasan dan fitur yang tidak didukung

Fitur-fitur berikut tidak didukung saat ini. Jika beban kerja Anda menggunakan fitur-fitur ini, kode itu perlu di-porting secara manual atau dengan pendamping kode AI pasca-transformasi.

- .svc file (file layanan)
- .file ashx (file handler generik)
- Referensi layanan & referensi web (referensi.map, reference.cs, file.wsdl)
- Campuran file.aspx (Formulir Web) dan file.cshtl (Razor) dalam proyek yang sama
- Ketergantungan pada tipe proyek yang belum didukung oleh AWS Transform untuk.NET
- Third-party vendor UI libraries/controls
- Proyek Situs Web

Hanya proyek Aplikasi Web (berisi file.csproj) yang didukung. Untuk mengonversi proyek Situs Web ke proyek Aplikasi Web, lihat Panduan Microsoft Mengonversi Proyek Situs Web menjadi Proyek Aplikasi Web.

Laporan Transformasi

Setelah pekerjaan transformasi.NET Anda selesai, laporan transformasi ini tersedia:

- Laporan Ringkasan Transformasi
- Laporan Detail Transformasi

Laporan Ringkasan Transformasi

Setelah pekerjaan transformasi selesai, laporan ringkasan transformasi tersedia di konsol web. Unduh laporan ringkasan transformasi dari tab Dasbor. Pilih Unduh Laporan Ringkasan Transformasi, dan pilih Unduh sebagai HTML atau Unduh sebagai JSON.

Laporan Ringkasan Transformasi memberikan gambaran umum tentang transformasi dan informasi spesifik untuk setiap repositori yang diubah, termasuk:

- Detail Tugas
  - Ikhtisar pekerjaan transformasi
  - Ikhtisar hasil transformasi
  - Ikhtisar file
- Daftar repositori, dan untuk setiap repositori:
  - Ikhtisar Nama Repositori
  - Ringkasan Transformasi Solusi
  - Detail Proyek, termasuk Status, Jenis, Versi.NET, Total Baris Kode, Baris Kode yang Diubah, dan File yang Diubah

Laporan Detail Transformasi

Setelah pekerjaan transformasi selesai, laporan detail transformasi tersedia untuk pengguna konsol web dan IDE.

- Pengguna konsol web: laporan detail tersedia untuk setiap repositori yang diubah. Pada tab Dasbor, gunakan tautan Unduh laporan terperinci di daftar repositori untuk mengunduh laporan detail.
- Pengguna IDE: unduh laporan transformasi dari jendela AWS Transform Job Plan menggunakan ikon unduhan.

Laporan detail transformasi adalah laporan HTML interaktif. Ini berisi informasi mendalam tentang transformasi, termasuk alasan perubahan file dan detail kesalahan yang dapat ditindaklanjuti. Informasi diatur dalam hierarki berikut:

- Informasi Pekerjaan
- Untuk setiap solusi:
  - Ringkasan Transformasi Solusi
  - Ikhtisar Transformasi
  - Proyek
  - Untuk setiap proyek:
    - Ringkasan Proyek
    - Ringkasan Kesalahan Bangun
    - NuGet perubahan paket

    - Perubahan File
    - Membangun Kesalahan
- Hasil Tes Unit
- Laporan Kesiapan Linux

Untuk panduan bagian laporan detail transformasi dengan contoh, lihat posting blog Mengumumkan AWS Transformasi untuk.NET laporan transformasi terperinci.

Praktik terbaik untuk transformasi.NET

Ikuti panduan ini untuk hasil modernisasi .NET yang lebih baik dengan AWS Transform. Praktik terbaik ini berlaku untuk asisten AI percakapan .NET, yang saat ini tersedia untuk Visual Studio IDE.

Topik

Sebelum Anda bertransformasi

Tinjau rekomendasi berikut sebelum Anda memulai pekerjaan transformasi.NET.

Gunakan asisten AI percakapan.NET

Gunakan agen .NET baru untuk hasil transformasi terbaik. Asisten AI percakapan tersedia Visual Studio melalui AWS Toolkit for Visual Studio. Ini juga tersedia untuk Kiroatau IDE lainnya melalui Kiro kekuatan untuk agen AWS Transform dan AWS Transform MCP. Agen .NET baru belum tersedia di konsol web.

Gunakan alat lain jika tujuannya bukan transformasi.NET ke .NET

AWS Transform untuk.NET sesuai untuk .NET Framework ke modernisasi kode.NET dan upgrade .NET ke versi yang lebih baru. Ini tidak cocok untuk transformasi yang mencakup bahasa non-.NET, seperti Java, atau non-.net framework, seperti React. Ini juga tidak menyediakan refactoring arsitektur aplikasi. Gunakan AWS Transform custom atau Kiro untuk modernisasi semacam itu.

Gunakan edisi stabil Studio Visual, sebaiknya VS2026

Gunakan edisi GA Visual Studio daripada pratinjau atau edisi Insiders untuk menghindari komplikasi dari perangkat lunak pratinjau. AWS Toolkit dapat digunakan dengan Visual Studio 2026 atau 2022. VS2026 direkomendasikan karena kinerjanya lebih baik dengan konsumsi memori yang berkurang.

Gunakan yang terbaru AWS Toolkit untuk Studio Visual

Tetap pada versi terbaru AWS Toolkit Visual Studio untuk perbaikan bug dan pembaruan fitur. Anda dapat memeriksa versi toolkit Anda dan meningkatkan di Extensions > Manage Extensions > Installed.

Bekerja di Studio Visual IDE untuk proyek dengan kompleksitas menengah atau lebih tinggi

Pengalaman IDE memberi Anda pandangan transformasi terdalam, di mana Anda dapat bekerja bersama agen.NET saat mengubah proyek. Untuk proyek kompleksitas sedang, tinggi, atau kritis, gunakan IDE daripada konsol web.

Memiliki rencana validasi

Putuskan bagaimana Anda akan memvalidasi aplikasi Anda untuk kebenaran setelah diubah. Post-transformation, Anda harus mengonfirmasi bahwa itu berperilaku dan terlihat dengan benar saat dijalankan, dan bahwa kontrol keamanan, logika bisnis, dan penyimpanan data masih utuh. Apakah validasi akan dilakukan secara manual and/or menggunakan otomatisasi pengujian? Jika solusi Anda berisi pengujian unit, AWS Transform akan mem-portingnya untuk Anda dan menjalankannya, dan hasilnya akan muncul di laporan transformasi.

Menyediakan NuGet dependensi paket pribadi

Jika solusi Anda bergantung pada paket pribadi, pastikan mereka tersedia di mesin lokal Anda sehingga IDE dan AWS Transform dapat menemukannya.

Tentukan penggantian paket atau minta rekomendasi agen

Untuk paket yang tidak memiliki versi.NET modern, Anda dapat menyesuaikan rencana transformasi untuk menentukan substitusi paket yang Anda inginkan, atau meminta rekomendasi agen.

Jangan mengakhiri pekerjaan Anda sampai Anda yakin tidak ingin ada perubahan lagi

Setelah transformasi selesai, jangan segera akhiri pekerjaan Anda jika Anda menginginkan perubahan tambahan. Transformasi tidak lengkap sampai Anda mengatakannya. Anda dapat meminta perubahan sampai Anda menandai pekerjaan selesai.

Setelah berubah, serahkan ke pendamping kode AI untuk menyelesaikan aplikasi Anda

Berharap untuk menggabungkan AWS Transform dengan pendamping kode AI sepertiKiro. Pola yang umum adalah pertama-tama AWS bertransformasi dengan Transform untuk.NET, kemudian menyerahkan ke pendamping kode AI untuk menyelesaikan aplikasi. Pekerjaan finalisasi itu mungkin termasuk mengidentifikasi dan memperbaiki kesalahan runtime, masalah perilaku, dan gaya UX.

Dalam perjalanan modernisasi, pikirkan AWS Transform sebagai perjalanan ekspres yang membawa Anda jauh menuju tujuan Anda, dan pendamping kode AI sebagai perjalanan lokal yang membawa Anda ke tujuan akhir Anda.

Untuk membantu hand-off dari AWS Transform ke pendamping kode AI, AWS Transform menyediakan laporan transformasi HTML dan dokumen penurunan harga Langkah Berikutnya di akhir transformasi.

Penilaian dan perencanaan

Setelah Anda memulai pekerjaan transformasi, AWS Transform menganalisis kode Anda dan menghasilkan laporan penilaian dan rencana transformasi. Gunakan praktik terbaik berikut untuk mendapatkan hasil maksimal dari fase ini.

Tinjau penilaian dan ajukan pertanyaan untuk menetapkan harapan Anda

Tinjau temuan penilaian dengan cermat untuk memahami proyek mana yang akan menantang untuk diubah dan mengapa. Agen menyajikan proyek dengan tingkat kompleksitas (rendah, sedang, tinggi, kritis) dan menjelaskan mengapa mereka memiliki peringkat itu.

Ajukan pertanyaan untuk memahami lebih lanjut tentang area yang menantang. Contoh:

- Tanyakan kepada agen apa tingkat kepercayaannya dalam transformasi proyek atau kasus penggunaan.
- Tanyakan kepada agen tugas pasca-transformasi apa yang mungkin diperlukan dengan pendamping kode AI.

Tinjau rencana dan diskusikan opsi dengan agen

Agen menyediakan rencana transformasi untuk ulasan Anda. Anda dapat mendiskusikan dan memodifikasi rencana ini. Mulailah dengan memahami pilihan Anda. Tanyakan kepada agen alternatif apa yang ada untuk bagian mana pun dari rencana tersebut.

Misalnya, rencana transformasi mungkin mengusulkan transformasi layanan WCF ke ASP.NET API Web. Namun, organisasi Anda lebih memilih untuk tetap dengan WCF. Anda meminta agen untuk opsi, dan belajar bahwa alternatifnya adalah mengubahnyaCoreWCF.

Unggah dokumen kemudi untuk menyesuaikan rencana organisasi Anda

Rencana transformasi asli dibuat tanpa pemahaman tentang persyaratan dan preferensi organisasi Anda. Anda dapat memberi tahu agen konvensi organisasi, spesifikasi aplikasi, tumpukan teknologi pilihan, dan contoh pola pengkodean yang disukai.

Setelah mengunggah dokumen kemudi, jelaskan untuk memberikan konteks agen. Dengan informasi tambahan dari dokumen pengarah, agen dapat menyiapkan rencana yang disesuaikan dengan bagaimana organisasi Anda suka bekerja.

Sesuaikan rencana secara bebas

Agen .NET fleksibel dan Anda dapat dengan bebas menyesuaikan rencana transformasi, dalam domain transformasi.NET ke .NET. Anda dapat meminta perubahan rencana dalam obrolan, atau jika Anda mau, Anda dapat bekerja dengan file penurunan harga. Tinjau dan perbaiki rencana sampai puas.

Jika rencana perlu ditinjau oleh beberapa pemangku kepentingan, Anda dapat mengunduh paket sebagai file penurunan harga, membagikannya, dan kemudian mengunggah paket yang disesuaikan.

Misalnya, Anda meninjau rencana transformasi, dan memutuskan Anda ingin proyek perpustakaan kelas diubah menjadi .NET Standard, bukan .NET 10. Dalam obrolan Anda mengatakan, “Merevisi rencana untuk mengubah semua pustaka kelas menjadi .NET Standard 2.0" dan agen memberi Anda rencana yang direvisi.

Transformasi secara interaktif atau mandiri

Pengalaman Visual Studio IDE menawarkan mode otonom dan interaktif untuk mengontrol tingkat interaktivitas selama transformasi.

Gunakan mode interaktif dan otonom untuk mengontrol tingkat interaktivitas

Anda memilih mode saat memulai transformasi, dan Anda dapat mengubahnya kapan saja melalui dropdown di bagian atas jendela obrolan.

- Gunakan mode otonom untuk transformasi tanpa pengawasan, cocok untuk pekerjaan semalam atau jangka panjang. Anda masih memiliki kesempatan untuk meninjau hasil dan meminta perubahan di akhir pekerjaan.

- Gunakan mode interaktif untuk bekerja bersama agen dalam obrolan selama proses transformasi dengan rencana transformasi yang dapat disesuaikan, visibilitas tinggi ke detail transformasi, dan tinjauan langkah demi langkah dari setiap proyek. Anda dapat mengulangi untuk menyempurnakan hasil atau mengubah arah hingga puas.

Anda dapat menggabungkan mode. Misalnya, mulai dalam mode interaktif untuk meninjau dan menyesuaikan rencana transformasi, lalu beralih ke mode otonom untuk mengubah proyek secara otomatis.

Minta agen untuk menjelaskan perubahan kode

Jika Anda melihat perubahan kode yang tidak Anda pahami, mintalah agen untuk menjelaskannya.

Gunakan pos pemeriksaan untuk berhenti di proyek untuk ditinjau atau diubah

Dalam mode interaktif, setelah menyetujui rencana Anda, Anda dapat mengatur atau menghapus pos pemeriksaan untuk setiap proyek. Ketika sebuah proyek dengan pos pemeriksaan telah diubah, kode yang diubah diterapkan ke proyek di IDE dan agen berhenti untuk peninjauan Anda. Anda dapat meninjau hasilnya, dan jika Anda tidak menyukainya, Anda dapat meminta lebih banyak perubahan.

Saat berhenti di pos pemeriksaan, Anda dapat:

- Tinjau hasil seperti yang disajikan dalam obrolan dan ajukan pertanyaan dalam obrolan. Misalnya: “Mengapa Anda mengubah API kriptografi di file X?”
- Lihat perubahan kode, dengan memilih tombol Lihat Hasil untuk proyek di jendela AWS Transform Job Plan.
- Perbaiki transformasi, dengan meminta perubahan tambahan. Misalnya: “Ubah ini untuk menggunakan logger X” atau “Ubah ini untuk menggunakan paket X alih-alih paket Y”.
- Coba lagi transformasi, lakukan dengan instruksi yang berbeda. Misalnya: “Ubah pustaka kelas ini untuk menargetkan .NET Standard alih-alih .NET 10" atau “Coba lagi transformasi ini dan ubah target dari Blazor ke MVC Razor Pages”.
- Buat perubahan kode Anda sendiri di IDE.
- Beritahu agen untuk melanjutkan ke pos pemeriksaan proyek berikutnya.

Iterasi sampai puas

Modernisasi secara inheren berulang, dan Anda dapat berharap untuk mengulangi dengan agen. Anda dapat membuat perubahan di pos pemeriksaan proyek, dan juga di akhir transformasi. Ulangi sesuai kebutuhan sampai Anda puas. Transformasi tidak lengkap sampai Anda mengatakannya.

Kode bersama agen ketika berhenti di pos pemeriksaan

Anda dapat membuat perubahan kode Anda sendiri saat Anda bekerja dengan agen. Ketika berhenti di pos pemeriksaan, Anda dapat mengedit file kode diVisual Studio. Perubahan Anda akan disinkronkan saat Anda mengarahkan agen untuk melanjutkan.

Validasi dan finalisasi

Setelah transformasi selesai, validasi aplikasi Anda yang diubah dan selesaikan untuk penggunaan produksi.

Tinjau laporan transformasi dan Langkah Selanjutnya untuk memahami apa yang dilakukan dan apa yang tersisa

Tinjau laporan transformasi HTML untuk memahami apa yang telah dilakukan. Tinjau file penurunan harga Langkah Berikutnya untuk memahami tugas apa yang masih tersisa. Mintalah agen untuk mengklarifikasi apa pun yang tidak jelas.

Gunakan pendamping kode AI untuk memvalidasi, men-debug, dan menyelesaikan aplikasi Anda

Tinjau aplikasi Anda yang diubah untuk kebenaran. Gunakan pendamping kode AI seperti Kirountuk menyelesaikan masalah.

Validasi pengujian unit

Jika solusi Anda berisi proyek pengujian unit (NUnit,xUnit, atauMSTest), AWS Transform akan memindahkannya ke .NET modern dan menjalankan pengujian untuk Anda. Periksa laporan transformasi HTML untuk hasil pengujian unit. Perbaiki pengujian yang gagal dengan pendamping kode AI.

Validasi fungsionalitas aplikasi

Jalankan aplikasi Anda dan konfirmasikan fungsinya cocok dengan aslinya. Perhatikan masalah fungsional dan kesalahan runtime. Benar dengan pendamping kode AI.

Validasi UI aplikasi

Jalankan aplikasi Anda dan konfirmasikan tampilannya dan berperilaku seperti aslinya termasuk navigasi dan perutean. Jika aplikasi Anda tidak memiliki gaya, mungkin file.css tidak berada di lokasi yang benar. Benar dengan pendamping kode AI.

Modernisasi SQL Server

AWS Transform for SQL Server Modernization adalah AI-powered layanan yang mengotomatiskan modernisasi full-stack database Microsoft SQL Server dan aplikasi.NET terkait dengan Amazon Aurora PostgreSQL. Layanan mengatur seluruh perjalanan migrasi dari konversi skema, migrasi data, dan memodifikasi kode aplikasi agar sesuai dengan PostgreSQL target baru, membuat tim Anda lebih produktif dengan mengotomatiskan tugas yang kompleks dan padat karya.

Wilayah yang didukung

AWS Transform untuk SQL Server tersedia di AS Timur (Virginia N.) - us-east-1

Cross-Region Penggunaan: Untuk database di wilayah yang tidak didukung, Anda dapat mengkloning database ke wilayah yang didukung untuk transformasi, lalu menerapkan hasilnya kembali ke wilayah target Anda.

Kemampuan dan fitur utama

Transformasi basis data

- Konversi skema: Secara otomatis mengonversi skema SQL Server ke Aurora PostgreSQL, termasuk tabel, tampilan, indeks, kendala, dan hubungan
- Transformasi prosedur tersimpan: Mengonversi prosedur yang T-SQL disimpan PL/pgSQL dengan AI-enhanced akurasi
- Migrasi data: Migrasi data dengan validasi integritas menggunakan AWS Database Migration Service (DMS)
- Objek database: Mendukung pemicu, fungsi, tampilan, kolom yang dihitung, dan kolom identitas
- Validasi: Verifikasi integritas data otomatis dan pemeriksaan integritas referensial

Transformasi aplikasi

- Transformasi Entity Framework: Memperbarui konfigurasi Entity Framework 6.3-6.5 dan EF Core 1.0-8.0 untuk PostgreSQL
- ADO.NET transformasi: Mengkonversi kode akses ADO.NET data dari SQL Server ke penyedia PostgreSQL.
- Pembaruan string koneksi: Secara otomatis memperbarui semua string koneksi database ke database PostgreSQL target baru
- Perubahan penyedia database: Mengganti penyedia SQL Server dengan Npgsql (penyedia PostgreSQL)
- Pembaruan konfigurasi ORM: Memodifikasi pemetaan tipe data, kolom identitas, dan konfigurasi khusus database

Validasi & orkestrasi

- Wave-based modernisasi: Mengatur perkebunan besar ke dalam fase migrasi logis
- Pemetaan ketergantungan: Mengidentifikasi hubungan antara aplikasi dan database
- Human-in-the-loop (HITL) pos pemeriksaan: Menyediakan gerbang peninjauan dan persetujuan pada tahap kritis
- Validasi otomatis: Menguji kompatibilitas skema, integritas data, dan fungsionalitas aplikasi
- CI/CD integrasi: Terintegrasi dengan pipa pengembangan yang ada

Deployment

- Penyebaran Amazon ECS dan Amazon EC2: Penyebaran kontainer otomatis dengan dukungan auto-scaling
- Infrastructure-as-code generasi: Membuat CloudFormation atau templat AWS CDK
- Validasi penerapan otomatis: Memverifikasi penerapan yang berhasil dengan pemeriksaan kesehatan
- Kemampuan rollback: Mendukung prosedur rollback jika masalah muncul

Versi dan jenis proyek yang didukung

Versi SQL Server

AWS Transform mendukung versi SQL Server berikut:

Versi SQL Server	Status Support
SQL Server 2022	Didukung
SQL Server 2019	Didukung
SQL Server 2017	Didukung
SQL Server 2016	Didukung
SQL Server 2014	Didukung
SQL Server 2012	Didukung
SQL Server 2008 R2	Didukung

> **Note**
>
> Semua edisi SQL Server didukung (Express, Standard, Enterprise). SQL Server dapat di-host di AWS (Amazon RDS untuk SQL Server atau SQL Server di Amazon EC2) atau dihosting di luar. AWS

VERSI.NET

Versi .NET	Status Support
.NET 10	Didukung
.NET 8	Didukung
.NET 7	Didukung

Versi .NET	Status Support
.NET 6 (Inti)	Didukung
.NET Framework 4.x dan sebelumnya	Tidak Didukung

> **Important**
>
> Legacy .NET Framework 4.x dan versi sebelumnya tidak didukung. Jika aplikasi Anda menggunakan .NET Framework, Anda harus terlebih dahulu meng-upgrade ke .NET Core 6+ menggunakan AWS Transform for .NET modernisasi sebelum menggunakan kemampuan transformasi SQL Server.

Versi Kerangka Entitas

Kerangka Kerja	Versi yang Didukung
Kerangka Entitas 6	6.3, 6.4, 6.5
Inti Kerangka Entitas	1.0 hingga 8.0
ADO.NET	Semua versi (GA)

Repositori kode sumber

AWS Transform mendukung platform kode sumber berikut:

- GitHub dan Server GitHub Perusahaan
- GitLab.com dan GitLab Self-Managed
- Bitbucket Cloud dan Pusat Data Bitbucket
- Azure DevOps dan DevOps Azure Server
- Amazon S3

Basis data target

AWS Transform target Amazon Aurora PostgreSQL (PostgreSQL 15+ kompatibel) dengan dukungan untuk fitur dan pengoptimalan Aurora terbaru.

Persyaratan Teknis

Persyaratan basis data

- Microsoft SQL Server versi 2008 R2 hingga 2022
- SQL Server di-host di AWS (Amazon RDS untuk SQL Server atau SQL Server di Amazon EC2) atau dihosting di luar AWS
- Untuk database AWS-host, database dan AWS Transform harus berada di Region yang sama AWS .
- Untuk database yang dihosting di luar AWS, konektivitas jaringan ke layanan AWS Transform diperlukan.
- Pengguna database dengan izin VIEW DEFINITION dan VIEW DATABASE STATE
- Kata sandi basis data hanya menggunakan karakter ASCII yang dapat dicetak (tidak termasuk '/', '@', "", dan spasi)
- VPC yang berisi sumber SQL Server harus memiliki subnet di setidaknya 2 Availability Zone yang berbeda (diperlukan untuk grup subnet replikasi DMS)

Persyaratan aplikasi

- .NET Core 6, 7, atau 8 aplikasi
- Entity Framework 6.3-6.5 atau Entity Framework Core 1.0-8.0, atau ADO.NET
- Koneksi database dapat ditemukan dalam kode sumber
- Aplikasi berhasil membangun dan menjalankan
- Kode sumber di platform repositori yang didukung

AWS persyaratan akun

- AWS akun dengan akses administrator
- Pusat Identitas IAM diaktifkan
- Peran layanan yang diperlukan dibuat (lihat petunjuk penyiapan di bawah)

- VPC dengan konfigurasi jaringan yang sesuai

Pemrosesan dan penyimpanan data

Lokasi pemrosesan

- Pemrosesan skema terjadi dalam instance DMS dalam VPC Anda
- Migrasi data bersifat opsional dan dapat dikecualikan jika diperlukan
- Artefak transformasi disimpan di wilayah layanan AWS Transform

Artefak yang disimpan

Item berikut disimpan di wilayah layanan:

- Log agen
- Hasil penilaian
- File skema SQL
- Artefak keluaran DMS

> **Important**
>
> Penting untuk Residensi Data: Bahkan ketika migrasi data dipilih keluar, metadata dan artefak pemrosesan disimpan di wilayah layanan. Ini penting untuk organisasi dengan persyaratan residensi data yang ketat.

Manajemen Artifak

- Opsi pelanggan untuk enkripsi menggunakan kunci KMS Anda sendiri
- Periode TTL (time-to-live) yang ditentukan untuk semua artefak
- Artefak dapat diunduh untuk penyimpanan offline

Persyaratan SQL Server

Persyaratan basis data

Database yang dihosting secara eksternal

Database yang dihosting di luar AWS didukung. Untuk menggunakan database yang dihosting secara eksternal, Anda harus memastikan konektivitas jaringan antara database dan layanan AWS Transform.

Beban kerja SSIS

- Batasan: Transformasi SQL Server Integration Services (SSIS) tidak didukung.
- Solusi: Gunakan AWS Glue atau layanan ETL lainnya untuk migrasi SSIS.

Server tertaut

- Batasan: Koneksi database eksternal melalui server tertaut memerlukan konfigurasi manusia.
- Solusi: Konfigurasikan ulang server tertaut setelah migrasi atau gunakan pendekatan alternatif seperti tautan basis data atau integrasi tingkat aplikasi.

T-SQL Pola yang kompleks

- Batasan: Beberapa T-SQL pola lanjutan mungkin memerlukan campur tangan manusia.
- Solusi: Gunakan Amazon Kiro CLI untuk konversi SQL yang kompleks. AWS Transform menandai ini untuk ditinjau selama proses transformasi.

Persyaratan aplikasi

Kerangka .NET Warisan

- Batasan: .NET Framework 4.x dan versi sebelumnya tidak didukung.
- Solusi: Gunakan AWS Transform untuk.NET untuk meningkatkan ke.NET Core 6+ terlebih dahulu, lalu gunakan transformasi SQL Server.

Versi Kerangka Entitas

- Batasan: Hanya Entity Framework 6.3-6.5 dan EF Core 1.0-8.0 yang didukung.
- Solusi: Tingkatkan ke versi Entity Framework yang didukung sebelum transformasi.

VB.NET aplikasi

- Batasan: VB.NET tidak didukung.
- Solusi: Konversi ke C # atau gunakan AWS Transform custom untuk mengonversi dari ke C #. VB.NET

Cross-database dependensi

- Batasan: Tantangan ketika skema database berinteraksi di beberapa database.
- Solusi: Meninjau dan memfaktorkan ulang kueri lintas basis data sebelum migrasi. Pertimbangkan untuk mengkonsolidasikan database atau menggunakan skema PostgreSQL.
- Dampak: Mungkin memerlukan intervensi manusia untuk skenario lintas basis data yang kompleks.

Repository-database kopling

- Batasan: Tantangan ketika satu repositori melayani beberapa database.
- Solusi: Pertimbangkan restrukturisasi repositori atau pendekatan migrasi bertahap.
- Dampak: Mungkin memerlukan perencanaan tambahan untuk migrasi berbasis gelombang.

Persyaratan infrastruktur

Tunggal account/region per pekerjaan

- Batasan: Setiap pekerjaan transformasi menargetkan satu AWS akun dan wilayah.
- Solusi: Buat beberapa pekerjaan transformasi untuk penerapan multi-akun atau multi-wilayah.

Target deployment

- Batasan: Penerapan Amazon ECS dan Amazon EC2 didukung.

Persyaratan repositori

NuGet Paket pribadi

- Batasan: NuGet Paket pribadi memerlukan konfigurasi tambahan.
- Solusi: Konfigurasikan NuGet feed pribadi dalam pengaturan transformasi sebelum memulai pekerjaan.

Penyiapan SQL Server

Lakukan langkah-langkah ini di lingkungan SQL Server Anda untuk mengaktifkan modernisasi AWS Transform.

Pengaturan dan konfigurasi basis data

Langkah 1: Buat pengguna database dengan izin yang diperlukan

Buat pengguna database khusus untuk AWS Transform dengan izin yang diperlukan. Jika Anda sudah memiliki pengguna Konversi Skema DMS, Anda dapat menggunakannya kembali.

Connect ke instance SQL Server Anda dan jalankan perintah berikut:

```
-- Create the login in master database
USE master;
CREATE LOGIN [atx_user] WITH PASSWORD = 'YourStrongPassword123!';

-- Switch to your application database
USE [YourDatabaseName];
CREATE USER [atx_user] FOR LOGIN [atx_user];

-- Grant required permissions
GRANT VIEW DEFINITION TO [atx_user];
GRANT VIEW DATABASE STATE TO [atx_user];
ALTER ROLE [db_datareader] ADD MEMBER [atx_user];

-- Grant master database permissions
USE master;
GRANT VIEW SERVER STATE TO [atx_user];
GRANT VIEW ANY DEFINITION TO [atx_user];
```

> **Note**
>
> Ulangi perintah khusus database (USE, CREATE USER, GRANT) untuk setiap database yang ingin Anda modernisasi.

Peran db_datareader hanya diperlukan untuk migrasi data, bukan untuk konversi skema saja.

- Peran db_datareader memberikan akses baca ke semua tabel dalam database
- Peran ini hanya diperlukan saat melakukan migrasi data.
- Untuk konversi skema saja (tanpa migrasi data), peran db_datareader TIDAK diperlukan
- Izin lainnya (VIEW DEFINITION, VIEW DATABASE STATE, dll.) Cukup untuk konversi skema

Langkah 2: Simpan kredensil di AWS Secrets Manager

Simpan kredensyal database Anda dengan aman di Secrets Manager AWS . Lewati langkah ini jika Anda sudah memiliki rahasia yang dibuat untuk DMS.

1. Arahkan ke AWS Secrets Manager di konsol
2. Pilih Simpan rahasia baru
3. Konfigurasikan rahasianya:
   - Tipe rahasia: Kredensyal untuk database lain
   - Basis Data: Microsoft SQL Server
   - Nama pengguna: atx_user (atau nama pengguna pilihan Anda)
   - Kata sandi: Kata sandi yang Anda buat
   - Nama server: Titik akhir SQL Server Anda
   - Nama database: Nama database Anda
   - Port: 1433 (atau port khusus Anda)
4. Pilih Berikutnya
5. Masukkan nama rahasia: atx-db-modernnization-sqlserver
6. Tambahkan tag yang diperlukan (tag ini wajib):
   - Kunci: Proyek, Nilai: atx-db-modernisasi
   - Kunci: Pemilik, Nilai: konektor database
7. Pilih Berikutnya melalui layar yang tersisa

8. Pilih Toko
9. Perhatikan Rahasia ARN untuk digunakan pada langkah berikutnya

> **Important**
>
> Kata sandi database harus menggunakan karakter ASCII yang dapat dicetak saja, tidak termasuk '/', '@', "", dan spasi. Rahasia yang dijadwalkan untuk dihapus dapat menyebabkan kegagalan transformasi.

Langkah 3: Buat peran DMS yang diperlukan

AWS Transform membutuhkan peran IAM khusus untuk operasi DMS. Terapkan peran ini menggunakan CloudFormation template di bawah ini.

> **Note**
>
> Jika AWS akun Anda sudah memiliki DMS-related peran yang ada, ubah templat ini untuk menggunakan kembali sumber daya tersebut daripada membuat duplikat.

Buat file bernama dms-roles.yaml dengan konten berikut:

```
AWSTemplateFormatVersion: '2010-09-09'
Description: 'DMS Service Roles for AWS Transform SQL Server Modernization'

Resources:
  DMSCloudWatchLogsRole:
    Type: AWS::IAM::Role
    Properties:
      RoleName: dms-cloudwatch-logs-role
      AssumeRolePolicyDocument:
        Version: '2012-10-17'
        Statement:
          - Effect: Allow
            Principal:
              Service:
                - dms.amazonaws.com
                - schema-conversion.dms.amazonaws.com
            Action: sts:AssumeRole
```

```
      ManagedPolicyArns:
        - arn:aws:iam::aws:policy/service-role/AmazonDMSCloudWatchLogsRole

  DMSS3AccessRole:
    Type: AWS::IAM::Role
    Properties:
      RoleName: dms-s3-access-role
      AssumeRolePolicyDocument:
        Version: '2012-10-17'
        Statement:
          - Effect: Allow
            Principal:
              Service:
                - dms.amazonaws.com
                - schema-conversion.dms.amazonaws.com
            Action: sts:AssumeRole
      Policies:
        - PolicyName: S3TaggedAccess
          PolicyDocument:
            Version: '2012-10-17'
            Statement:
              - Effect: Allow
                Action:
                  - s3:GetBucketLocation
                  - s3:GetBucketVersioning
                  - s3:PutObject
                  - s3:PutBucketVersioning
                  - s3:GetObject
                  - s3:GetObjectVersion
                  - s3:ListBucket
                  - s3:DeleteObject
                Resource: arn:aws:s3:::atx-db-modernization-*
                Condition:
                  StringEquals:
                    aws:ResourceAccount: !Ref AWS::AccountId

  DMSSecretsManagerRole:
    Type: AWS::IAM::Role
    Properties:
      RoleName: dms-secrets-manager-role
      AssumeRolePolicyDocument:
        Version: '2012-10-17'
        Statement:
          - Effect: Allow
```

```
            Principal:
              Service:
                - dms.amazonaws.com
                - schema-conversion.dms.amazonaws.com
            Action: sts:AssumeRole
      Policies:
        - PolicyName: SecretsManagerTaggedAccess
          PolicyDocument:
            Version: '2012-10-17'
            Statement:
              - Effect: Allow
                Action:
                  - secretsmanager:GetSecretValue
                  - secretsmanager:DescribeSecret
                Resource: '*'
                Condition:
                  StringEquals:
                    secretsmanager:ResourceTag/Project: atx-db-modernization
                    secretsmanager:ResourceTag/Owner: database-connector

  DMSVPCRole:
    Type: AWS::IAM::Role
    Properties:
      RoleName: dms-vpc-role
      AssumeRolePolicyDocument:
        Version: '2012-10-17'
        Statement:
          - Effect: Allow
            Principal:
              Service:
                - dms.amazonaws.com
                - schema-conversion.dms.amazonaws.com
            Action: sts:AssumeRole
      ManagedPolicyArns:
        - arn:aws:iam::aws:policy/service-role/AmazonDMSVPCManagementRole

  DMSServerlessRole:
    Type: AWS::IAM::ServiceLinkedRole
    Properties:
      AWSServiceName: dms.amazonaws.com
      Description: 'Service Linked Role for AWS DMS Serverless'

Outputs:
  DMSCloudWatchLogsRoleArn:
```

```
    Description: ARN of the DMS CloudWatch Logs Role
    Value: !GetAtt DMSCloudWatchLogsRole.Arn
  DMSS3AccessRoleArn:
    Description: ARN of the DMS S3 Access Role
    Value: !GetAtt DMSS3AccessRole.Arn
  DMSSecretsManagerRoleArn:
    Description: ARN of the DMS Secrets Manager Role
    Value: !GetAtt DMSSecretsManagerRole.Arn
  DMSVPCRoleArn:
    Description: ARN of the DMS VPC Role
    Value: !GetAtt DMSVPCRole.Arn
    Export:
      Name: !Sub ${AWS::StackName}-VPCRole

  DMSServerlessRoleArn:
    Description: ARN of the DMS Serverless Role
    Value: !Sub 'arn:aws:iam::${AWS::AccountId}:role/aws-service-role/
dms.amazonaws.com/AWSServiceRoleForDMSServerless'
    Export:
      Name: !Sub ${AWS::StackName}-ServerlessRole
```

Terapkan CloudFormation tumpukan menggunakan AWS CLI:

```
aws cloudformation create-stack \
  --stack-name dms-roles \
  --template-body file://dms-roles.yaml \
  --capabilities CAPABILITY_NAMED_IAM \
  --region us-east-1
```

Atau terapkan menggunakan AWS Konsol:

1. Arahkan ke CloudFormation di AWS Konsol
2. Pilih Buat tumpukan
3. Pilih Unggah file templat
4. Unggah file dms-roles.yaml
5. Masukkan nama tumpukan: dms-role
6. Akui kemampuan IAM
7. Pilih Buat tumpukan

Langkah 4: Konfigurasikan keamanan jaringan

Pastikan konektivitas jaringan yang tepat antara AWS Transform, database SQL Server Anda, dan AWS layanan lainnya.

Konfigurasi grup keamanan (pendekatan yang disarankan)

Pendekatan yang Disarankan: Gunakan kontrol akses berbasis grup keamanan daripada IP-based aturan. Ini memberikan keamanan yang lebih baik, manajemen yang lebih mudah, dan bekerja dengan mulus dengan arsitektur AWS Transform.

Mengapa Kontrol Group-Based Akses Keamanan?

- Konversi Skema DMS menciptakan Antarmuka Jaringan Elastis (ENI) dalam VPC Anda
- Database Anda tidak perlu diakses publik
- AWS Transform tidak mengekspos alamat IP pribadi, membuat IP-based aturan menjadi rumit
- Referensi grup keamanan menyediakan pembaruan dinamis dan otomatis sebagai skala sumber daya

Konfigurasikan grup keamanan SQL Server Anda

Saat mengonfigurasi Profil Instans Konversi Skema DMS di AWS Transform, Anda menentukan Grup Keamanan untuk instance DMS SC. Grup keamanan database Anda harus mengizinkan lalu lintas masuk dari grup keamanan DMS SC ini.

Step-by-step konfigurasi:

1. Identifikasi Grup Keamanan Konversi Skema DMS:
   - Ini ditentukan saat membuat Profil Instance di AWS Transform
   - Perhatikan ID Grup Keamanan (mis., sg-0123456789abcdef0)
2. Perbarui aturan masuk Grup Keamanan SQL Server Anda:
   - Jenis: TCP Kustom
   - Port: 1433 (atau port SQL Server kustom Anda)
   - Sumber: ID Grup Keamanan Konversi Skema DMS
   - Deskripsi: “Izinkan akses Konversi Skema DMS”
3. Untuk target Aurora PostgreSQL (setelah pembuatan):

- Jenis: PostgreSQL
- Port: 5432 (atau port PostgreSQL kustom Anda)
- Sumber: ID Grup Keamanan Konversi Skema DMS
- Deskripsi: “Izinkan akses Konversi Skema DMS”

> **Important**
>
> Penting untuk Model Keamanan dengan Keistimewaan Terkecil: Jika organisasi Anda menggunakan model keamanan “paling tidak memiliki hak istimewa” yang memblokir semua lalu lintas secara default, Anda harus secara eksplisit mengizinkan lalu lintas masuk dari Grup Keamanan Konversi Skema DMS ke port database Anda. Jangan buka port 1433 ke semua sumber atau rentang IP.

Diperlukan AWS konektivitas layanan

Pastikan VPC Anda dapat berkomunikasi dengan:

- AWS Mengubah titik akhir layanan
- AWS DMS titik akhir
- Titik akhir Aurora PostgreSQL
- Titik akhir S3 untuk penyimpanan artefak
- AWS Secrets Manager titik akhir
- AWS CodeConnections titik akhir

Titik Akhir VPC: Untuk jaringan pribadi, konfigurasikan titik akhir VPC untuk AWS layanan yang diperlukan guna menghindari dependensi gateway internet.

Persyaratan untuk database yang dihosting secara eksternal

Jika database SQL Server Anda di-host di luar AWS, pastikan prasyarat berikut terpenuhi dan kemudian selesaikan langkah-langkah penyiapan sebelum Anda memulai modernisasi.

Prasyarat

- AWS Akun dengan VPC

- Konektivitas jaringan antara VPC dan database eksternal. Untuk informasi tentang mengonfigurasi konektivitas jaringan, lihat Mengonfigurasi konektivitas jaringan di AWS DMS Panduan Pengguna.

Langkah-langkah pengaturan

1. Buat rahasia AWS Secrets Manager dengan detail koneksi untuk database eksternal. Untuk informasi selengkapnya, lihat the section called “Langkah 2: Simpan kredensil di AWS Secrets Manager”.
2. Saat diminta, berikan ID VPC dan ID grup keamanan untuk menghubungkan ke database eksternal. AWS Transform meminta Anda untuk informasi ini karena nama host database dalam rahasia tidak dapat diselesaikan dalam akun. AWS

AWS Ubah pengaturan

Mulailah perjalanan modernisasi Anda dengan mengakses konsol AWS Transform dan memulai proyek modernisasi baru. Langkah ini menetapkan ruang kerja Anda dan menyiapkan konfigurasi dasar untuk transformasi full-stack Anda.

Tindakan untuk Menyelesaikan:

- Arahkan ke layanan AWS Transform di AWS Management Console
- Pilih Modernisasi Microsoft dari opsi layanan
- Pilih Buat proyek modernisasi baru
- Berikan nama dan deskripsi proyek
- Pilih AWS wilayah target Anda (harus sesuai dengan lokasi SQL Server Anda)
- Tinjau dan konfirmasi pengaturan proyek

> **Note**
>
> Praktik Terbaik: Gunakan nama proyek deskriptif yang menyertakan nama aplikasi dan lingkungan target (misalnya, 'CustomerPortal-Production-Modernization') untuk dengan mudah mengidentifikasi proyek dalam skenario multi-gelombang.

Buat pekerjaan modernisasi SQL Server

Connect database dan repo kode sumber

Konfigurasikan konektor basis data

Membangun konektivitas aman ke database SQL Server Anda dengan mengkonfigurasi konektor database. Konektor melakukan analisis lingkungan, penemuan ketergantungan, dan memungkinkan AWS Transform mengakses skema database dan metadata Anda untuk penilaian dan konversi.

Tindakan untuk Menyelesaikan:

- Pilih Konfigurasi Konektor Database dari wizard penyiapan
- Pilih metode koneksi: Koneksi Baru atau Koneksi yang Ada
- Berikan detail koneksi SQL Server (titik akhir, port, nama database)
- Konfigurasikan otentikasi dan uji konektivitas
- Tinjau database yang ditemukan dan konfirmasi pemilihan

> **Note**
>
> Konektivitas Jaringan: Pastikan grup keamanan SQL Server dan NACL mengizinkan koneksi masuk dari AWS titik akhir layanan Transform.

Connect repositori kode sumber

Aktifkan AWS Transform untuk mengakses kode sumber aplikasi.NET Anda. AWS Transform mendukung tiga metode untuk menyediakan kode sumber: konektor Personal Access Token (PAT) (disarankan) AWS CodeConnections, atau Amazon S3. Integrasi ini memungkinkan layanan untuk menganalisis kode aplikasi Anda, mengidentifikasi dependensi database, dan melakukan transformasi kode otomatis untuk kompatibilitas PostgreSQL.

Tindakan untuk Menyelesaikan:

- Arahkan ke bagian Connect Source Code Repository
- Pilih metode otentikasi Anda: Konektor PAT (disarankan) AWS CodeConnections, atau Amazon S3
- Pilih repositori yang berisi aplikasi.NET
- Tentukan cabang untuk analisis (biasanya main/master atau pengembangan)
- Validasi akses repositori dan struktur kode

> **Note**
>
> Repository Discovery: AWS Transform secara otomatis memindai repositori Anda untuk mengidentifikasi proyek.NET, konfigurasi Entity Framework, string koneksi database, dan dependensi SQL.

> **Note**
>
> Catatan Keamanan: AWS Transform hanya memerlukan akses baca ke repositori Anda dan membuat cabang fitur baru untuk kode yang diubah. Cabang utama Anda tetap tak tersentuh.

Manusia dalam lingkaran

AWS Transform menggunakan mekanisme human-in-the-loop (HITL) untuk memastikan kualitas dan memungkinkan Anda untuk meninjau dan menyetujui keputusan transformasi kritis. Pos pemeriksaan berikut membutuhkan perhatian Anda:

Tinjauan dan persetujuan rencana gelombang

Apa yang Anda ulas: Gelombang migrasi yang diusulkan, termasuk database dan aplikasi mana yang dikelompokkan bersama dan urutan gelombang.

Anda dapat:

- Menyetujui rencana gelombang
- Sesuaikan gelombang dengan memindahkan database antar gelombang
- Ubah urutan gelombang
- Membagi atau menggabungkan gelombang

Setelah persetujuan, AWS Transform melanjutkan dengan konversi skema.

Ulasan konversi skema

Apa yang Anda tinjau: Objek database yang dikonversi termasuk tabel, prosedur tersimpan, fungsi, dan pemicu. Item aksi menyoroti objek yang membutuhkan perhatian.

Anda dapat:

- Terima kode yang dikonversi
- Ubah kode yang dikonversi
- Bendera untuk tinjauan manusia setelah transformasi
- Lihat perbandingan berdampingan dari kode asli dan yang dikonversi

Apa yang terjadi setelah persetujuan: AWS Transform menerapkan skema ke Aurora PostgreSQL dan melanjutkan dengan migrasi data (jika dikonfigurasi).

Ulasan kode aplikasi

Apa yang Anda tinjau: Semua perubahan kode aplikasi termasuk konfigurasi Entity Framework, string koneksi, kode akses data, dan panggilan prosedur tersimpan.

Anda dapat:

- Terima perubahan untuk setiap file
- Memodifikasi kode yang diubah
- Tolak perubahan (tidak disarankan)
- Tambahkan komentar untuk tim Anda
- Unduh kode yang diubah untuk tinjauan lokal

Apa yang terjadi setelah persetujuan: AWS Transform melakukan perubahan ke cabang baru di repositori Anda dan melanjutkan dengan validasi.

Ulasan hasil validasi

Apa yang Anda tinjau: Hasil validasi otomatis termasuk kompatibilitas skema, pemeriksaan integritas data, dan status pembuatan aplikasi.

Anda dapat:

- Tinjau tes yang lulus
- Selidiki tes yang gagal
- Peringatan alamat
- Lanjutkan ke penerapan atau kembali untuk memperbaiki masalah

Apa yang terjadi setelah persetujuan: AWS Transform mempersiapkan penerapan ke lingkungan target Anda.

Persetujuan penyebaran

Apa yang Anda tinjau: Konfigurasi penerapan termasuk templat infrastruktur-sebagai-kode, konfigurasi layanan ECS, dan pengaturan penerapan.

Anda dapat:

- Tinjau dan sesuaikan pengaturan penerapan
- Menyetujui penerapan untuk melanjutkan
- Keterlambatan penyebaran untuk pengujian tambahan
- Unduh kode infrastruktur untuk ditinjau

Setelah disetujui, AWS Transform menyebarkan aplikasi dan database modern Anda ke lingkungan target.

Penilaian dan perencanaan gelombang

Mengatur landing zone

Konfigurasikan lingkungan infrastruktur target tempat aplikasi modern Anda akan digunakan. Pengaturan landing zone mencakup penyediaan Aurora PostgreSQL, konfigurasi jaringan, dan konfigurasi penerapan referensi.

Tindakan untuk Menyelesaikan:

- Pilih Siapkan Zona Pendaratan dari menu konfigurasi
- Konfigurasikan target PostgreSQL Aurora (versi, kelas instance,) Multi-AZ
- Konfigurasikan pengaturan jaringan dan keamanan (VPC, subnet, grup keamanan)

> **Note**
>
> Pertimbangan Biaya: Instans PostgreSQL Aurora menimbulkan biaya berkelanjutan. Pertimbangkan untuk menggunakan Aurora Serverless v2 untuk lingkungan. development/ testing

Penilaian

Lakukan analisis komprehensif skema database, kode aplikasi, dan dependensi Anda. Fase penilaian memberikan peringkat kompleksitas transformasi, mengidentifikasi tantangan potensial, dan menghasilkan laporan terperinci.

Tindakan untuk Menyelesaikan:

- Pilih Cabang Repositori untuk analisis kode
- Pilih Repositori dan Database untuk transformasi
- Tinjauan Hasil Penilaian dan peringkat kompleksitas
- Menghasilkan Laporan Penilaian dengan proyeksi usaha
- Laporan ekspor untuk tinjauan dan persetujuan pemangku kepentingan

> **Note**
>
> Wawasan Penilaian: Penilaian memberikan analisis rinci objek skema, prosedur tersimpan, dependensi aplikasi, dan perkiraan tingkat intervensi manusia yang diperlukan.

Perencanaan gelombang

Atur modernisasi Anda menjadi gelombang yang dapat dikelola berdasarkan kompleksitas transformasi, prioritas bisnis, dan dependensi aplikasi. AWS Transform menghasilkan rekomendasi gelombang cerdas yang dapat Anda sesuaikan.

Tindakan untuk Menyelesaikan:

- Tinjau Rencana AI-Generated Gelombang dan rekomendasi
- Sesuaikan Konfigurasi Gelombang berdasarkan prioritas bisnis
- Validasi Dependensi antara aplikasi dan database
- Selesaikan Rencana Gelombang menggunakan antarmuka percakapan
- Tetapkan prioritas eksekusi gelombang dan tetapkan tanggung jawab

> **Note**
>
> AI-Powered Rekomendasi: Anda dapat berinteraksi dengan AI melalui bahasa alami untuk menyesuaikan rekomendasi: 'Pindahkan CustomerDB ke Gelombang 1' atau 'Gabungkan kedua aplikasi ini ke gelombang yang sama. '

> **Note**
>
> Praktik Terbaik: Mulailah dengan gelombang pilot yang berisi 1-2 database kompleksitas rendah untuk membangun proses dan membangun kepercayaan tim.

Migrasi data untuk setiap gelombang

Konversi skema untuk setiap gelombang

Ubah skema database SQL Server menjadi PostgreSQL-compatible setara menggunakan kemampuan konversi. AI-enhanced Proses ini mengkonversi tabel, prosedur tersimpan, fungsi, pemicu, dan objek database lainnya.

Tindakan untuk Menyelesaikan:

- Berikan Target Konversi Skema dan konfigurasikan preferensi
- Jalankan Konversi Otomatis menggunakan layanan Konversi Skema DMS
- Tinjau Hasil Konversi dan identifikasi item intervensi manusia
- Menangani Intervensi Manual melalui petunjuk HITL
- Validasi koreksi manusia dan jalankan kembali konversi jika diperlukan

> **Note**
>
> Intervensi Manual Umum: Server tertaut, tipe yang ditentukan pengguna, T-SQL pola lanjutan, dan fitur SQL khusus vendor biasanya memerlukan tinjauan dan koreksi manusia.

Migrasi data untuk setiap gelombang

Transfer data dari SQL Server ke Aurora PostgreSQL AWS Database Migration Service menggunakan integrasi ().AWS DMS Pilih antara migrasi data produksi atau pembuatan data sintetis untuk tujuan pengujian.

Tindakan untuk Menyelesaikan:

- Pilih Migrasi Data Produksi atau Pembuatan Data Sintetis
- Konfigurasikan contoh replikasi DMS untuk transfer data
- Memantau kemajuan migrasi dan menangani inkonsistensi data
- Validasi integritas data dan kendala referensial
- Konfirmasikan penyelesaian yang berhasil sebelum melanjutkan

> **Note**
>
> Integrasi DMS Otomatis: AWS Transform mengabstraksi kompleksitas konfigurasi DMS, secara otomatis menangani pembuatan endpoint, konfigurasi tugas, dan pemetaan tipe data.

> **Note**
>
> Validasi Data: Layanan melakukan pengujian empiris dengan menjalankan kueri identik terhadap basis data sumber dan target untuk memastikan konsistensi data.

Migrasi kode untuk setiap gelombang

Ubah kode aplikasi.NET untuk bekerja dengan Aurora PostgreSQL, termasuk pembaruan konfigurasi Entity Framework, modifikasi string koneksi, adaptasi kueri SQL, dan penyesuaian khusus kerangka kerja.

Tindakan untuk Menyelesaikan:

- Jalankan Analisis Kode Otomatis untuk dependensi SQL Server
- Lakukan Transformasi Kerangka Kerja (pembaruan penyedia Kerangka Entitas)

- Menyesuaikan sintaks SQL Query ke SQL PostgreSQL-compatible
- Konfigurasikan Manajemen Cabang Target untuk kode yang diubah
- Tangani Intervensi Manual untuk pola yang kompleks

Alur kerja modernisasi SQL Server

Bagian ini memberikan panduan langkah demi langkah dari proses modernisasi SQL Server lengkap menggunakan Transform. AWS

Langkah 1: Buat pekerjaan modernisasi SQL Server

Mulailah perjalanan modernisasi Anda dengan membuat pekerjaan transformasi baru di konsol AWS Transform.

1. Masuk ke konsol AWS Transform
2. Pilih Buat pekerjaan modernisasi
3. Pilih pekerjaan modernisasi Windows dan kemudian pilih modernisasi SQL Server
4. Masukkan rincian pekerjaan:
   - Nama pekerjaan: Nama deskriptif untuk proyek Anda
   - Deskripsi: Deskripsi opsional
   - Wilayah target: AWS wilayah untuk penerapan
5. Pilih Buat pekerjaan

> **Important**
>
> Jangan sertakan informasi identitas pribadi (PII) dalam nama pekerjaan Anda.

Langkah 2: Connect ke database SQL Server

Connect AWS Transform ke database SQL Server Anda untuk mengaktifkan analisis skema dan konversi.

Buat konektor database

1. Dalam pekerjaan modernisasi SQL Server Anda, navigasikan ke Connect to resources

2. Pilih Connect to SQL Server database
3. Pilih Buat konektor baru
4. Masukkan informasi konektor:
   - Nama C onnector: Nama deskriptif
   - AWS ID akun: Akun tempat SQL Server di-host
5. Setelah dikonfirmasi, Anda akan menerima tautan untuk persetujuan. Salin tautan persetujuan untuk mendapatkan persetujuan dari AWS admin Anda untuk akun tersebut. Setelah mereka menyetujui, Anda dapat melanjutkan ke langkah berikutnya.
6. Setelah administrator Anda menyetujui permintaan konektor, klik Kirim untuk melanjutkan ke pengaturan koneksi kode sumber.

Langkah 3: Connect repositori kode sumber

AWS Transform membutuhkan akses ke kode sumber aplikasi.NET Anda untuk menganalisis dan mengubah kode yang berinteraksi dengan database SQL Server Anda. AWS Transform mendukung tiga metode untuk menyediakan kode sumber.

Pilih metode otentikasi Anda

Konektor Personal Access Token (PAT) (disarankan)

Terbaik untuk tim yang memerlukan cakupan izin khusus, dukungan penyedia yang dihosting sendiri, atau akses ke API khusus penyedia seperti Rahasia. GitHub Anda membuat PAT di penyedia kode sumber Anda dengan izin khusus, menyimpannya AWS Secrets Manager, dan AWS Transform mengambilnya saat diperlukan. Anda bertanggung jawab untuk mengelola rotasi dan kedaluwarsa token.

AWS CodeConnections

Terbaik untuk tim yang menginginkan manajemen kredensyal otomatis. AWS CodeConnections menggunakan integrasi penyedia terkelola yang menangani otentikasi melalui alur otorisasi OAuth 2.0. AWS mengelola seluruh siklus hidup kredenal, termasuk penyegaran dan rotasi token otomatis. Tidak diperlukan manajemen kredensyal manual.

Amazon S3

Unggah kode sumber Anda langsung ke bucket Amazon S3. AWS Transform mengakses kode dari bucket selama pekerjaan transformasi.

Fitur	Konektor PAT (disarankan)	AWS CodeConnections
Manajemen kredensial	Manual (dikelola pelanggan)	Otomatis (AWS-dikelola)
Siklus hidup Token	Diperlukan rotasi manual	Penyegaran otomatis
Fleksibilitas izin	Cakupan yang sepenuhnya dapat disesuaikan	Izin tetap
Self-hosted dukungan penyedia	Didukung	Tidak tersedia
Kompleksitas pengaturan	Sedang (pembuatan dan penyimpanan token manual)	Rendah (otorisasi satu kali)
Penyimpanan token	Pelanggan AWS Secrets Manager	AWS-dikelola

Siapkan konektor PAT (disarankan)

Dengan konektor PAT, Anda membuat Token Akses Pribadi di penyedia kode sumber Anda dengan izin khusus, menyimpannya dengan aman AWS Secrets Manager, dan AWS Transform mengambilnya saat diperlukan. Anda bertanggung jawab untuk mengelola siklus hidup token termasuk rotasi dan kedaluwarsa. AWS Transform secara otomatis membuat peran IAM yang diperlukan dengan izin untuk mengakses rahasia Anda.

Konektor PAT mendukung penyedia berikut, termasuk DNS/URL versi yang dihosting sendiri dan khusus:

- GitHub dan Server GitHub Perusahaan
- GitLab.com dan GitLab Self-Managed
- Bitbucket Cloud dan Pusat Data Bitbucket
- Azure DevOps dan DevOps Azure Server

Buat Token Akses Pribadi

Buat PAT di penyedia kode sumber Anda. Izin yang diperlukan bervariasi menurut penyedia. Pilih tab untuk penyedia Anda.

Important

Salin token segera setelah pembuatan. Anda tidak dapat melihatnya lagi. Atur kedaluwarsa untuk durasi pekerjaan transformasi. Jangan atur kedaluwarsa agar tidak pernah kedaluwarsa.

Warning

Jangan pernah memasukkan token PAT ke repositori kode atau membagikannya melalui saluran yang tidak aman. Selalu simpan di dalamnya AWS Secrets Manager.

GitHub

Arahkan ke Pengaturan, Pengaturan pengembang, Token akses pribadi, Fine-grained token. Pilih repositori yang akan diubah dan berikan izin berikut.

Izin repositori

Izin	Akses	Tujuan
Daftar Isi	Membaca dan menulis	Membaca kode sumber dan menulis kode yang diubah kembali ke repositori
Metadata	Read-only	Mengakses informasi repositori dasar

Izin organisasi (diperlukan untuk repositori organisasi)

Izin	Akses	Tujuan
Anggota	Read-only	Daftar organisasi yang dapat diakses token untuk penemuan repositori

GitLab

Arahkan ke Edit Profil, Akses Token. Pilih cakupan berikut.

Lingkup	Tujuan
`read_api`	Membaca metadata repositori, informasi proyek, detail pengguna, dan daftar grup dan cabang
`read_repository`	Membaca file kode sumber dan struktur repositori untuk analisis
`write_repository`	Menulis kode yang diubah kembali ke repositori

Bitbucket

Arahkan ke Pengaturan akun, Keamanan, Buat dan kelola Token API. Cakupan yang diperlukan tergantung pada jenis token Anda.

Workspace/Repository Token (ATCT — Autentikasi pembawa, tidak diperlukan nama pengguna)

Izin	Akses	Tujuan
Repositori	Membaca dan menulis	Daftar repo, membaca cabang, dan menulis kode yang diubah melalui git push

Token API Akun (ATAT — Autentikasi dasar dengan email) atau Kata Sandi Aplikasi (ATBB — Autentikasi dasar dengan nama pengguna)

Lingkup	Tujuan
`read:account`	Mengidentifikasi pengguna yang diautentikasi untuk menyelesaikan keanggotaan repositori
`read:workspace:bitbucket`	Daftar ruang kerja yang dapat diakses token sehingga AWS Transform dapat menghitung repositori mereka. Tidak diperlukan jika Anda menentukan daftar ruang kerja secara rahasia.
`read:repository:bitbucket`	Daftar repositori dan membaca metadata dan informasi cabang

Lingkup	Tujuan
`write:repository:bitbucket`	Menulis kode yang diubah kembali ke repositori melalui git push

Azure DevOps

Arahkan ke Pengaturan pengguna, Token akses pribadi. Pilih Cakupan yang ditentukan khusus. Untuk lingkup organisasi, pilih Semua organisasi yang dapat diakses (disarankan) atau tentukan satu organisasi.

Lingkup	Akses	Tujuan
Kode	Membaca & menulis	Membaca kode sumber, daftar repositori dan cabang, dan menulis kode yang diubah kembali
Profil Pengguna	Baca	Memvalidasi akses token dan menemukan identitas pengguna untuk pencarian organisasi
Manajemen Hak Anggota	Baca	Daftar organisasi yang dapat diakses token untuk penemuan repositori

Simpan PAT di AWS Secrets Manager

1. Buka AWS Secrets Manager konsol.
2. Pilih Simpan rahasia baru.
3. Untuk Tipe rahasia, pilih Tipe rahasia lainnya.
4. Tambahkan pasangan nilai kunci berdasarkan penyedia dan jenis hosting Anda:
   - Cloud-hosted penyedia — Tambahkan kunci bernama `token` PAT Anda sebagai nilainya.
     - Untuk Azure DevOps dengan organisasi tertentu, tambahkan juga kunci bernama `organization` dengan nama organisasi Anda.

    - Untuk kata sandi aplikasi Bitbucket (ATBB), tambahkan juga kunci bernama `username` dengan nama pengguna Bitbucket Anda. Untuk token API akun Bitbucket (ATAT), tambahkan kunci yang diberi nama `email` dengan alamat email Bitbucket Anda.
- Self-hosted dan DNS/URL penyedia kustom — Tambahkan kunci berikut: `host` (URL server Anda, misalnya`https://github.mycompany.com`), `provider_type` (`github`,`gitlab`,`bitbucket`, atau`auado`), dan `token` (PAT Anda).
    - Untuk Azure DevOps dengan organisasi tertentu, tambahkan juga kunci bernama `organization` dengan nama organisasi Anda.
    - Untuk kata sandi aplikasi Bitbucket (ATBB), tambahkan juga kunci bernama `username` dengan nama pengguna Bitbucket Anda. Untuk token API akun Bitbucket (ATAT), tambahkan kunci yang diberi nama `email` dengan alamat email Bitbucket Anda.

Contoh berikut menunjukkan cara rahasia mencari penyedia yang GitHub dihosting di AWS Secrets Manager cloud:

```
{
  "token": "your-github-personal-access-token"
}
```

Contoh berikut menunjukkan kata sandi aplikasi Bitbucket (ATBB):

```
{
  "token": "your-bitbucket-app-password",
  "username": "my-bitbucket-username"
}
```

Contoh berikut menunjukkan GitLab instance yang dihosting sendiri:

```
{
  "host": "https://gitlab.mycompany.com",
  "provider_type": "gitlab",
  "token": "your-gitlab-personal-access-token"
}
```

5. Pilih Berikutnya.
6. Masukkan nama rahasia, misalnya`github-pat-myproject`.
7. (Opsional) Pilih kunci KMS yang dikelola pelanggan untuk enkripsi.

8. Lengkapi wizard dan pilih Store.
9. Salin ARN Rahasia. Anda memerlukan nilai ini saat mengonfigurasi pekerjaan AWS Transform.

Jika Anda menggunakan kunci KMS yang dikelola pelanggan untuk mengenkripsi rahasia Anda (bukan kunci AWS-managed default), Anda harus memperbarui kebijakan kunci KMS untuk memungkinkan AWS Transform mendekripsi rahasia. Tambahkan pernyataan berikut ke kebijakan kunci KMS yang dikelola pelanggan Anda:

```
{
  "Sid": "Allow AWS Transform to decrypt secrets",
  "Effect": "Allow",
  "Principal": {
    "Service": "transform.amazonaws.com"
  },
  "Action": [
    "kms:Decrypt",
    "kms:DescribeKey"
  ],
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "kms:ViaService": "secretsmanager.REGION.amazonaws.com",
      "kms:EncryptionContext:SecretARN": "YOUR-SECRET-ARN"
    }
  }
}
```

Ganti *REGION* dengan AWS Wilayah Anda (misalnya,`us-east-1`) dan *YOUR-SECRET-ARN* dengan ARN rahasia Anda. `kms:ViaService`Kondisi ini memastikan kunci KMS hanya dapat digunakan melalui AWS Secrets Manager layanan. `kms:EncryptionContext:SecretARN`Kondisi ini membatasi dekripsi ke rahasia spesifik Anda.

Untuk memperbarui kebijakan kunci KMS Anda:

1. Buka konsol AWS KMS di`https://console.aws.amazon.com/kms`.
2. Di panel navigasi, pilih Kunci yang dikelola pelanggan.
3. Pilih kunci KMS Anda.
4. Di tab Kebijakan kunci, pilih Edit.
5. Tambahkan pernyataan kebijakan ke kebijakan yang ada.

6. Pilih Simpan perubahan.

> **Note**
>
> Jika Anda menggunakan kunci AWS-managed default (`aws/secretsmanager`), Anda tidak perlu mengubah kebijakan kunci KMS apa pun.

Konfigurasikan AWS Transformasi pekerjaan

1. Dalam tugas AWS Transform Anda, navigasikan ke Connect to resources.
2. Pilih Connect source code repository.
3. Pilih Konektor PAT sebagai metode otentikasi.
4. Masukkan ARN Rahasia dari Langkah 2.
5. (Opsional) Masukkan ARN Kunci KMS jika Anda menggunakan kunci KMS yang dikelola pelanggan.
6. Pilih repositori dan cabang Anda.
7. Pilih Lanjutkan.

AWS Transform secara otomatis membuat peran IAM dengan izin yang diperlukan untuk mengakses rahasia Anda.

Rotasi dan pemeliharaan token

Anda bertanggung jawab untuk memutar token PAT sebelum kedaluwarsa. Untuk memutar token:

1. Buat PAT baru di penyedia kode sumber Anda dengan izin yang sama.
2. Perbarui nilai rahasia di AWS Secrets Manager.
3. Verifikasi bahwa tugas AWS Transform Anda dapat mengakses repositori dengan token baru.
4. Cabut PAT lama di penyedia kode sumber Anda.

Memecahkan masalah konektor PAT

Akses Ditolak — PAT Tidak Valid

Verifikasi bahwa PAT belum kedaluwarsa. Konfirmasikan bahwa PAT memiliki cakupan yang diperlukan untuk penyedia Anda. Periksa apakah PAT disimpan dengan benar AWS Secrets Manager.

Tidak dapat mengambil rahasia

Verifikasi bahwa ARN Rahasia sudah benar. Periksa log pekerjaan untuk mengonfirmasi bahwa AWS Transform membuat peran IAM. Jika Anda menggunakan kunci KMS yang dikelola pelanggan, verifikasi kebijakan kunci.

Izin tidak cukup

PAT mungkin tidak memiliki cakupan yang diperlukan untuk operasi. Regenerasi PAT dengan cakupan yang diperlukan dan perbarui nilai rahasia di. AWS Secrets Manager

Penyiapan AWS CodeConnections

AWS CodeConnections menggunakan integrasi penyedia terkelola yang secara otomatis mengambil kredensyal OAuth sementara melalui alur otorisasi OAuth 2.0. Izin dikonfigurasi di aplikasi penyedia dan dikelola sepenuhnya oleh AWS. Anda mengotorisasi aplikasi sekali, dan AWS menangani semua manajemen kredensyal.

1. Dalam pekerjaan modernisasi SQL Server Anda, navigasikan ke Connect to resources.
2. Pilih Connect source code repository.
3. Jika Anda tidak memiliki koneksi yang ada, pilih Buat koneksi.
4. Pilih penyedia repositori Anda:
   - GitHub / GitHub Perusahaan
   - GitLab.com
   - Bitbucket Cloud
   - Repositori Azure
5. Ikuti alur otorisasi untuk penyedia Anda.
6. Setelah otorisasi, pilih Connect.

Pilih repositori dan cabang Anda

1. Pilih repositori Anda dari daftar.
2. Pilih cabang yang ingin Anda ubah (biasanya utama, master, atau kembangkan).
3. (Opsional) Tentukan subdirektori jika aplikasi.NET Anda tidak ada di root repositori.
4. Pilih Lanjutkan.

> **Note**
>
> AWS Transform membuat cabang baru untuk kode yang diubah. Anda dapat meninjau dan menggabungkan perubahan melalui proses peninjauan kode normal Anda.

Persetujuan akses repositori

Untuk GitHub dan beberapa platform lainnya, administrator repositori harus menyetujui permintaan koneksi:

1. AWS Transform menampilkan tautan verifikasi.
2. Bagikan tautan ini dengan administrator repositori Anda.
3. Administrator meninjau dan menyetujui permintaan dalam pengaturan repositori mereka.
4. Setelah administrator menyetujui permintaan, status koneksi berubah menjadi Disetujui.

> **Important**
>
> Proses persetujuan dapat memakan waktu tergantung pada kebijakan organisasi Anda. Rencanakan sesuai.

Langkah 4: Buat konektor penyebaran (opsional)

Jika Anda ingin menyebarkan aplikasi yang diubah ke AWS akun Anda, Anda memiliki opsi untuk memilih konektor penyebaran.

Siapkan konektor penyebaran

1. Pilih Ya jika Anda ingin menerapkan aplikasi Anda. Memilih Tidak akan melewati langkah ini.

2. Tambahkan AWS akun Anda di mana Anda ingin menyebarkan aplikasi yang diubah.
3. Tambahkan nama yang membantu Anda mengingat konektor dengan mudah
4. Kirimkan permintaan konektor untuk persetujuan.

Persetujuan konektor penyebaran

Administrator AWS akun Anda harus menyetujui permintaan koneksi untuk konektor penyebaran.

1. AWS Transform menampilkan tautan verifikasi
2. Bagikan tautan ini dengan administrator AWS akun Anda
3. Administrator meninjau dan menyetujui permintaan dalam pengaturan repositori mereka
4. Setelah disetujui, status koneksi berubah menjadi Disetujui

> **Important**
>
> Proses persetujuan dapat memakan waktu tergantung pada kebijakan organisasi Anda. Rencanakan sesuai.

Langkah 5: Konfirmasikan sumber daya Anda

Setelah terhubung ke database dan repositori Anda, AWS Transform memverifikasi bahwa semua sumber daya yang diperlukan dapat diakses dan siap untuk transformasi.

Apa AWS Transform memverifikasi

- Konektivitas basis data: Koneksi aktif, pengguna memiliki izin yang diperlukan, database dapat diakses, versi didukung
- Akses repositori: Repositori dapat diakses, cabang ada, file proyek.NET terdeteksi, koneksi database dapat ditemukan
- Kesiapan lingkungan: Konfigurasi VPC mendukung DMS, peran layanan yang AWS diperlukan ada, konektivitas jaringan dibuat, kompatibilitas wilayah dikonfirmasi

Tinjau daftar periksa pra-penerbangan

1. Arahkan ke Konfirmasi sumber daya Anda dalam rencana pekerjaan

2. Tinjau item daftar periksa:
   - # Koneksi database diverifikasi
   - # Akses repositori dikonfirmasi
   - # Versi .NET didukung
   - # Kerangka Entitas atau ADO.NET terdeteksi
   - # Konfigurasi jaringan valid
   - # Izin yang diperlukan diberikan
3. Jika semua item ditampilkan sebagai lengkap, pilih Lanjutkan
4. Jika ada item yang menunjukkan peringatan atau kesalahan, atasi sebelum melanjutkan

Langkah 6: Penemuan dan penilaian

AWS Transform menganalisis database SQL Server dan aplikasi.NET Anda untuk memahami ruang lingkup dan kompleksitas modernisasi.

Apa yang ditemukan

- Objek database: Tabel, tampilan, indeks, prosedur tersimpan, fungsi, pemicu, kendala, tipe data, kolom yang dihitung, kolom identitas, hubungan kunci asing
- Kode aplikasi: .NET struktur proyek, model dan konfigurasi Entity Framework, kode akses ADO.NET data, string koneksi database, panggilan prosedur tersimpan, kueri SQL dalam kode
- Dependensi: Aplikasi mana yang menggunakan database mana, dependensi lintas basis data, prosedur tersimpan bersama, pola akses data umum

Proses penemuan

- AWS Transform memulai penemuan secara otomatis setelah konfirmasi sumber daya
- Penemuan biasanya memakan waktu 5-15 menit tergantung pada ukuran database dan kompleksitas aplikasi
- Pantau kemajuan di worklog
- AWS Transform menampilkan pembaruan waktu nyata saat objek ditemukan

Tinjau hasil penemuan

Setelah penemuan selesai, navigasikan ke Discovery dan penilaian untuk meninjau:

Analisis Database:

- Jumlah objek: Jumlah tabel, tampilan, prosedur tersimpan, fungsi, pemicu
- Skor kompleksitas: Penilaian kompleksitas transformasi (Rendah, Sedang, Tinggi)
- Item tindakan: Objek yang mungkin membutuhkan perhatian manusia
- Fitur yang didukung: Fitur database yang akan dikonversi secara otomatis
- Fitur yang tidak didukung: Fitur yang memerlukan solusi

Analisis Aplikasi:

- Jenis proyek: ASP.NET Inti, Aplikasi Konsol, Perpustakaan Kelas, dll.
- Versi .NET: Terdeteksi versi.NET Core
- Kerangka akses data: Versi Kerangka Entitas atau ADO.NET
- Koneksi basis data: Jumlah string koneksi yang ditemukan
- Kompleksitas kode: Penilaian kompleksitas transformasi

Peta Ketergantungan:

- Representasi visual dari hubungan aplikasi-ke-database
- Cross-database dependensi
- Komponen bersama

Memahami penilaian kompleksitas

AWS Transform mengklasifikasikan modernisasi Anda menjadi tiga kategori:

Kompleksitas	Karakteristik	Hasil yang Diharapkan
Rendah (Kelas A)	Pola SQL standar (ANSI SQL), prosedur tersimpan sederhana, tipe data dasar, Kerangka Entitas dengan konfigurasi standar	Intervensi manusia minimal diharapkan, tingkat keberhasilan otomatisasi tinggi

Kompleksitas	Karakteristik	Hasil yang Diharapkan
Sedang (Kelas B)	T-SQL Pola lanjutan, prosedur tersimpan kompleks dengan logika bisnis, fungsi yang ditentukan pengguna, kolom yang dihitung	Beberapa intervensi manusia diperlukan, ulasan ahli direkomen dasikan
Tinggi (Kelas C)	Rakitan CLR, server tertaut, Pialang Layanan, pencarian teks lengkap yang kompleks	Diperlukan refactoring manusia yang signifikan, pertimbangkan pendekata n bertahap

Laporan penilaian

AWS Transform menghasilkan laporan penilaian terperinci yang mencakup:

- Ringkasan eksekutif dengan ikhtisar tingkat tinggi
- Inventaris database lengkap
- Inventaris aplikasi
- Persentase kesiapan transformasi
- Estimasi upaya
- Penilaian risiko dan strategi mitigasi
- Pendekatan yang direkomendasikan

Anda dapat mengunduh laporan penilaian untuk tinjauan offline dan berbagi dengan pemangku kepentingan.

Langkah 7: Buat dan tinjau rencana gelombang

Untuk perkebunan besar dengan banyak database dan aplikasi, AWS Transform menghasilkan rencana gelombang yang mengurutkan modernisasi dalam kelompok logis.

Apa itu rencana gelombang?

Rencana gelombang mengatur modernisasi Anda ke dalam fase (gelombang) berdasarkan:

- Ketergantungan antara database dan aplikasi
- Prioritas bisnis

- Toleransi risiko
- Ketersediaan sumber daya
- Kompleksitas teknis

Setiap gelombang berisi sekelompok database dan aplikasi yang dapat dimodernisasi bersama tanpa merusak dependensi.

Tinjau rencana gelombang

1. Arahkan ke perencanaan Gelombang dalam rencana kerja
2. Tinjau gelombang yang diusulkan
3. Untuk setiap gelombang, tinjau:
   - Database disertakan
   - Aplikasi disertakan
   - Ketergantungan pada gelombang lain
   - Perkiraan waktu transformasi
   - Tingkat kompleksitas
   - Aplikasi yang dapat digunakan

Sesuaikan rencana gelombang

Anda dapat menyesuaikan rencana gelombang agar sesuai dengan kebutuhan bisnis Anda dengan 2 cara:

Menggunakan JSON:

1. Pilih Unduh semua gelombang untuk mendapatkan file JSON dengan semua gelombang
2. Ubah gelombang di JSON dengan:
   - Memindahkan database antar gelombang
   - Memisahkan gelombang menjadi kelompok-kelompok yang lebih kecil
   - Menggabungkan gelombang bersama
   - Mengubah urutan gelombang
   - Menambahkan atau menghapus database dari ruang lingkup
3. Unggah file JSON kembali ke konsol dengan memilih Unggah paket gelombang

4. AWS Transform memvalidasi perubahan Anda dan memperingatkan jika dependensi dilanggar
5. Pilih Konfirmasi gelombang untuk memperbarui rencana gelombang

Menggunakan Obrolan:

Anda dapat memodifikasi rencana gelombang dengan mengobrol dengan agen dan memintanya untuk memindahkan repositori dan database ke gelombang tertentu. Pendekatan ini bekerja dengan baik jika Anda perlu melakukan pengeditan kecil pada ombak.

> **Important**
>
> Pastikan dependensi dihormati saat menyesuaikan gelombang. Mengubah aplikasi dependen sebelum database-nya dapat menyebabkan masalah.

Modernisasi basis data tunggal

Jika Anda memodernisasi satu database dan aplikasi, AWS Transform membuat rencana sederhana dengan satu gelombang. Anda dapat melanjutkan langsung ke transformasi tanpa perencanaan gelombang.

Menyetujui rencana gelombang

1. Setelah meninjau dan menyesuaikan (jika diperlukan), pilih Menyetujui rencana gelombang
2. AWS Transform mengunci rencana gelombang dan melanjutkan ke transformasi
3. Anda masih dapat memodifikasi rencana nanti dengan memilih Edit rencana gelombang

Langkah 8: Konversi skema

AWS Transform mengonversi skema database SQL Server Anda ke Aurora PostgreSQL, termasuk tabel, tampilan, prosedur tersimpan, fungsi, dan pemicu.

Bagaimana konversi skema bekerja

AWS Transform menggunakan Konversi Skema AWS DMS yang ditingkatkan dengan AI generatif untuk:

- Menganalisis skema dan hubungan SQL Server
- Memetakan tipe data dari SQL Server ke PostgreSQL setara

- Transformasi T-SQL ke PL/pgSQL
- Menangani kolom identitas, kolom yang dihitung, dan kendala
- Validasi konversi dan integritas referensial
- Hasilkan item tindakan untuk objek yang membutuhkan tinjauan manusia

Konversi yang didukung

Secara otomatis dikonversi:

- Tabel, tampilan, dan indeks
- Kunci utama dan kunci asing
- Periksa batasan dan nilai default
- Tipe data yang paling umum
- Prosedur tersimpan sederhana
- Fungsi dan pemicu dasar
- Kolom identitas (dikonversi ke SERIAL atau DIHASILKAN)
- Kolom yang paling dihitung

Mungkin memerlukan tinjauan manusia:

- Prosedur tersimpan yang kompleks dengan tingkat lanjut T-SQL
- Server-specific Fungsi SQL (GETUTCDATE, SUSER_SNAME, dll.)
- Kolom yang dihitung dengan ekspresi kompleks
- Full-text indeks pencarian
- Operasi tipe data XML
- Tipe data HIERARCHYID (membutuhkan ekstensi ltree)

Tidak otomatis dikonversi:

- Majelis CLR
- Server tertaut
- Pialang Layanan
- Lowongan kerja SQL Server Agent

Mulai konversi skema

1. Arahkan ke konversi Skema dalam rencana kerja
2. Tinjau pengaturan konversi:
   - Target versi PostgreSQL
   - Opsi ekstensi (ltree, PostGIS, dll.)
   - Konvensi penamaan
3. Pilih Mulai konversi
4. Pantau kemajuan di worklog
5. Konversi biasanya memakan waktu 10-30 menit tergantung pada jumlah objek database

Tinjau hasil konversi

Setelah konversi selesai, navigasikan ke konversi skema Tinjau:

Ringkasan Konversi:

- Objek dikonversi: Hitungan objek yang berhasil dikonversi
- Item tindakan: Objek yang membutuhkan perhatian manusia
- Peringatan: Potensi masalah untuk ditinjau
- Kesalahan: Objek yang tidak dapat dikonversi

Ulasan oleh Object Type:

- Tabel: Pemetaan tipe data, kendala, indeks
- Prosedur tersimpan: T-SQL ke PL/pgSQL konversi
- Fungsi: Tanda tangan fungsi dan perubahan logika
- Pemicu: Memicu perubahan sintaks dan waktu

Tinjau item tindakan

1. Pilih Lihat item tindakan
2. Untuk setiap item tindakan, tinjau:
   - Nama objek: Objek database

- Jenis masalah: Apa yang membutuhkan perhatian
- Keparahan: Kritis, Peringatan, atau Info
- Rekomendasi: Resolusi yang disarankan
- Kode asli: versi SQL Server
- Kode yang dikonversi: versi PostgreSQL

3. Untuk setiap item tindakan, Anda dapat:
   - Terima: Gunakan kode yang dikonversi
   - Ubah: Edit kode yang dikonversi
   - Bendera untuk nanti: Tandai untuk tinjauan manusia setelah transformasi

Contoh: Konversi prosedur tersimpan

Server SQL: T-SQL

```
CREATE PROCEDURE GetProductsByCategory
    @CategoryId INT,
    @PageSize INT = 10
AS
BEGIN
    SET NOCOUNT ON;

    SELECT TOP (@PageSize)
        ProductId,
        Name,
        Price,
        DATEDIFF(DAY, CreatedDate, GETUTCDATE()) AS DaysOld
    FROM Products
    WHERE CategoryId = @CategoryId
    ORDER BY Name
END
```

PostgreSQL PL/pgSQL yang dikonversi:

```
CREATE OR REPLACE FUNCTION get_products_by_category(
    p_category_id INTEGER,
    p_page_size INTEGER DEFAULT 10
)
RETURNS TABLE (
    product_id INTEGER,
```

```
    name VARCHAR(255),
    price NUMERIC(18,2),
    days_old INTEGER
) AS $$
BEGIN
    RETURN QUERY
    SELECT
        p.product_id,
        p.name,
        p.price,
        EXTRACT(DAY FROM (NOW() - p.created_date))::INTEGER AS days_old
    FROM products p
    WHERE p.category_id = p_category_id
    ORDER BY p.name
    LIMIT p_page_size;
END;
$$ LANGUAGE plpgsql;
```

Perubahan yang dilakukan:

- Prosedur dikonversi ke fungsi mengembalikan TABEL
- Nama parameter diawali dengan p_
- TOP dikonversi ke LIMIT
- DATEDIFF dikonversi ke EXTRACT
- GETUTCDATE () dikonversi ke NOW ()
- Nama kolom dikonversi ke huruf kecil (konvensi PostgreSQL)

Menyetujui konversi skema

1. Setelah meninjau semua item tindakan dan membuat modifikasi yang diperlukan
2. Pilih Menyetujui konversi skema
3. AWS Transform menyiapkan skema yang dikonversi untuk penerapan ke Aurora PostgreSQL

> **Note**
>
> Anda dapat mengunduh skema yang dikonversi sebagai skrip SQL untuk tinjauan offline atau kontrol versi.

Langkah 9: Migrasi data (opsional)

AWS Transform menyediakan opsi untuk memigrasikan data dari SQL Server ke Aurora PostgreSQL. Migrasi data bersifat opsional dan dapat dilewati jika Anda hanya memerlukan skema dan transformasi kode.

Opsi migrasi data

Opsi 1: Migrasi Data Produksi

Migrasikan data produksi aktual Anda menggunakan AWS DMS:

- Beban awal penuh dari semua data
- Replikasi berkelanjutan selama pengujian (CDC)
- Pemotongan waktu henti minimal
- Validasi data dan pemeriksaan integritas

Opsi 2: Lewati Migrasi Data

Transformasi skema dan kode saja:

- Berguna untuk development/testing lingkungan
- Kapan data akan dimigrasikan secara terpisah
- Untuk proyek pembuktian konsep

Konfigurasikan migrasi data

1. Arahkan ke Migrasi data dalam rencana pekerjaan
2. Pilih opsi migrasi Anda:
   - Migrasikan data produksi
   - Lewati migrasi data
3. Jika memigrasikan data produksi, konfigurasikan:
   - Jenis migrasi: Beban penuh, atau Beban penuh+CDC
   - Validasi: Aktifkan validasi data
   - Kinerja: Ukuran instans DMS

- Pilih >Mulai migrasi

Proses migrasi data produksi

Jika Anda memilih untuk memigrasikan data produksi:

1. Sinkronisasi awal: AWS DMS melakukan beban penuh dari semua tabel
2. Replikasi berkelanjutan: (Jika CDC diaktifkan) Menjaga data tetap disinkronkan
3. Validasi: Memverifikasi jumlah baris dan integritas data
4. Persiapan cutover: Mempersiapkan sinkronisasi akhir

Garis Waktu Migrasi:

- Database kecil (<10 GB): 30 menit - 2 jam
- Database menengah (10-100 GB): 2-8 jam
- Database besar (> 100 GB): 8+ jam

Validasi data

AWS Transform memvalidasi data yang dimigrasi dengan pemeriksaan berikut:

- Perbandingan jumlah baris (sumber vs target)
- Integritas kunci utama
- Hubungan kunci asing
- Kompatibilitas tipe data
- Hasil kolom yang dihitung
- Penanganan nilai nol

Langkah 10: Transformasi kode aplikasi

AWS Transform mengubah kode aplikasi.NET Anda untuk bekerja dengan Aurora PostgreSQL alih-alih SQL Server. Ini meminta nama cabang target di repositori Anda untuk melakukan kode sumber yang diubah. Setelah Anda memasukkan nama cabang, AWS Transform akan membuat cabang baru dan memulai transformasi untuk mencocokkan database PostgreSQL.

Apa yang akan ditransformasikan

Perubahan Kerangka Entitas:

- Penyedia basis data: UseSqlServer () → UseNpgsql ()
- String koneksi: Format SQL Server → Format PostgreSQL
- Pemetaan tipe data: tipe SQL Server → tipe PostgreSQL
- DbContext konfigurasi: SQL → Server-specific PostgreSQL-specific
- File migrasi: Diperbarui untuk kompatibilitas PostgreSQL

ADO.NET Perubahan:

- Kelas koneksi: SqlConnection → NpgsqlConnection
- Kelas perintah: SqlCommand → NpgsqlCommand
- Pembaca data: SqlDataReader → NpgsqlDataReader
- Parameter: SqlParameter → NpgsqlParameter
- S QL sintaks: → T-SQL PostgreSQL SQL

Perubahan Konfigurasi:

- String koneksi di appsettings.json
- NuGet Paket penyedia database
- Konfigurasi injeksi ketergantungan
- Startup/Program.cs konfigurasi

Mulai transformasi kode

1. Arahkan ke Transformasi aplikasi dalam rencana kerja
2. Tinjau pengaturan transformasi:
   - Targetkan versi.NET (jika memutakhirkan)
   - Versi penyedia PostgreSQL
   - Preferensi gaya kode
3. Pilih Mulai transformasi
4. Pantau kemajuan di worklog

5. Transformasi biasanya memakan waktu 15-45 menit tergantung pada ukuran basis kode

Langkah 11: Tinjau hasil transformasi

Sebelum melanjutkan ke penerapan, tinjau hasil transformasi lengkap untuk memastikan semuanya siap untuk pengujian.

Anda dapat mengunduh kode yang diubah dari cabang repositori untuk:

- Pengujian dan validasi lokal
- Ulasan kode di IDE Anda
- Integrasi dengan CI/CD pipeline Anda
- Komit kontrol versi

Anda juga dapat mengunduh ringkasan transformasi untuk meninjau perubahan bahasa alami yang dibuat oleh AWS Transform sebagai bagian dari transformasi.

Ringkasan transformasi

1. Arahkan ke ringkasan Transformasi dalam rencana kerja
2. Tinjau hasil keseluruhan:
   - Konversi skema: Objek dikonversi, item tindakan, peringatan
   - Migrasi data: Tabel bermigrasi, baris ditransfer, status validasi
   - Transformasi kode: File diubah, baris dimodifikasi, masalah diselesaikan
   - Skor kesiapan: Kesiapan keseluruhan untuk penerapan

Hasilkan laporan transformasi

AWS Transform menghasilkan laporan transformasi yang komprehensif:

1. Pilih Hasilkan laporan
2. Pilih jenis laporan:
   - Ringkasan eksekutif: High-level ikhtisar untuk pemangku kepentingan
   - Detail teknis: Dokumentasi transformasi lengkap
   - Item tindakan: Daftar tugas manusia yang diperlukan
3. Pilih Unduh laporan

Laporan tersebut meliputi:

- Ruang lingkup dan tujuan transformasi
- Objek dan kode diubah
- Masalah yang dihadapi dan resolusi
- Hasil validasi
- Penilaian kesiapan penerapan
- Rekomendasi untuk pengujian

Langkah 12: Validasi dan pengujian

Sebelum menerapkan ke produksi, validasi bahwa aplikasi yang diubah berfungsi dengan benar dengan Aurora PostgreSQL.

Jenis validasi

Validasi Otomatis: AWS Transform melakukan pemeriksaan otomatis:

- Validasi skema terhadap basis data sumber
- Verifikasi integritas data
- Pengujian kesetaraan kueri
- Validasi string koneksi
- Validasi konfigurasi

Validasi Manusia: Anda harus melakukan pengujian tambahan:

- Pengujian fungsional fitur aplikasi
- Pengujian integrasi dengan sistem lain
- Pengujian kinerja dan pembandingan
- Pengujian penerimaan pengguna
- Pengujian keamanan

Jalankan validasi otomatis

1. Arahkan ke Validasi dalam rencana kerja

2. Pilih Jalankan validasi
3. AWS Transform mengeksekusi tes validasi:
   - Konektivitas basis data
   - Kompatibilitas skema
   - Integritas data
   - Membangun aplikasi
   - Fungsionalitas dasar
4. Tinjau hasil validasi:
   - Lulus: Tes yang berhasil
   - Gagal: Tes yang perlu diperhatikan
   - Peringatan: Potensi masalah untuk ditinjau

Daftar periksa pengujian

Fungsi Database:

- Semua tabel dapat diakses
- Prosedur yang disimpan dijalankan dengan benar
- Fungsi mengembalikan hasil yang diharapkan
- Memicu api dengan tepat
- Kendala ditegakkan dengan benar
- Indeks meningkatkan kinerja kueri

Fungsionalitas Aplikasi:

- Aplikasi dimulai dengan sukses
- Koneksi basis data dibuat
- Operasi CRUD bekerja dengan benar
- Panggilan prosedur tersimpan berhasil
- Transaksi commit/rollback dengan benar
- Penanganan kesalahan berfungsi seperti yang diharapkan

Integritas Data:

- Jumlah baris yang cocok dengan sumber
- Kunci utama unik
- Kunci asing valid
- Kolom yang dihitung benar
- Penanganan nol sesuai
- Tipe data yang kompatibel

Kinerja:

- Waktu respons kueri dapat diterima
- Pengumpulan koneksi dikonfigurasi
- Indeks dioptimalkan
- Tidak ada masalah kueri N+1
- Operasi Batch efisien
- Pemanfaatan sumber daya wajar

Langkah 13: Penerapan

Setelah validasi berhasil, gunakan aplikasi dan database modern Anda ke produksi.

Opsi deployment

- Amazon ECS dan Amazon EC2 Linux

Pre-deployment daftar periksa

Sebelum menyebarkan ke produksi:

- Semua tes validasi lulus
- Pengujian kinerja selesai
- Tinjauan keamanan selesai
- Backup dan rencana rollback didokumentasikan
- Pemantauan dan peringatan dikonfigurasi
- Tim dilatih pada lingkungan baru

- Pemangku kepentingan diinformasikan tentang penyebaran
- Jendela pemeliharaan dijadwalkan

Terapkan ke Amazon ECS

1. Arahkan ke Deployment dalam rencana pekerjaan
2. Pilih Deploy ke ECS
3. Konfigurasikan pengaturan penerapan:
   - Cluster: Pilih atau buat cluster ECS
   - Layanan: Konfigurasikan layanan ECS
   - Definisi tugas: Tinjau definisi tugas yang dihasilkan
   - Load balancer: Konfigurasikan ALB/NLB
   - Auto-scaling: Tetapkan kebijakan penskalaan
4. Tinjau infrastruktur-sebagai-kode (CloudFormation templat atau kode CDK) AWS
5. Pilih Deploy

Pantau penyebaran

AWS Transform menyebarkan aplikasi Anda:

1. Menciptakan klaster Aurora PostgreSQL
2. Menerapkan skema basis data
3. Memuat data (jika ada)
4. Menyebarkan wadah aplikasi
5. Mengkonfigurasi penyeimbang beban
6. Menyiapkan auto-scaling

Pantau kemajuan penerapan dan verifikasi:

- Penyediaan infrastruktur
- Inisialisasi basis data
- Deployment aplikasi
- Pemeriksaan kesehatan lewat

- Aplikasi dapat diakses
- Koneksi database bekerja
- Log menunjukkan operasi normal

Post-deployment validasi

Setelah penyebaran:

Pengujian Asap:

- Verifikasi fungsionalitas kritis
- Uji alur kerja pengguna kunci
- Periksa poin integrasi
- Pantau tingkat kesalahan

Pemantauan Kinerja:

- Lacak waktu respons
- Memantau kueri basis data
- Periksa pemanfaatan sumber daya
- Tinjau log aplikasi

Validasi Pengguna:

- Lakukan pengujian penerimaan pengguna
- Kumpulkan umpan balik
- Mengatasi masalah apa pun
- Pelajaran dokumen yang dipetik

Prosedur rollback

Jika masalah muncul setelah penerapan:

Rollback Segera:

- Kembali ke versi aplikasi sebelumnya

- Beralih kembali ke SQL Server (jika masih tersedia)
- Pulihkan dari cadangan jika diperlukan

Rollback Sebagian:

- Gulung kembali komponen tertentu
- Menyimpan perubahan basis data
- Kembalikan kode aplikasi saja

Perbaiki Maju:

- Terapkan perbaikan terbaru ke versi Aurora PostgreSQL
- Menyebarkan kode aplikasi yang diperbarui
- Monitor untuk resolusi

> **Important**
>
> Simpan database SQL Server Anda untuk periode setelah cutover untuk mengaktifkan rollback jika diperlukan.

Post-deployment optimasi

Setelah penerapan berhasil:

Tuning Kinerja:

- Optimalkan kueri lambat
- Sesuaikan pengaturan kolam koneksi
- Fine-tune Parameter Aurora PostgreSQL
- Tinjau dan optimalkan indeks

Optimalisasi Biaya:

- Right-size Contoh Aurora
- Konfigurasikan auto-scaling dengan tepat

- Tinjau pengaturan penyimpanan
- Optimalkan retensi cadangan

Pengaturan pemantauan:

- Konfigurasikan CloudWatch dasbor
- Mengatur peringatan
- Aktifkan Pemantauan yang Ditingkatkan
- Konfigurasikan Performance Insights

Dokumentasi:

- Perbarui runbook
- Perubahan arsitektur dokumen
- Melatih tim operasi
- Buat panduan pemecahan masalah

Uji dan gunakan setelah transformasi

Terapkan aplikasi Anda yang telah diubah ke landing zone dan validasi konektivitas antara aplikasi Anda dan database yang diubah, dan secara opsional mengatur pipeline. CI/CD Contoh konfigurasi CI/CD pipeline dibuat sebagai referensi untuk membantu Anda menerapkan otomatisasi penerapan. Fase terakhir ini memvalidasi modernisasi lengkap.

Konfigurasikan sumber daya

Anda perlu mengonfigurasi sumber daya yang digunakan dalam penerapan seperti peran IAM, profil instance, dan bucket S3. AWS Transform tidak memiliki izin untuk membuat bucket S3 atau membuat perubahan di IAM Anda. Tindakan ini diperlukan untuk dilakukan oleh pengguna yang memiliki izin untuk membuat peran baru. Biasanya itu adalah administrator akun.

Untuk menyederhanakan proses, kami menyediakan skrip pengaturan yang secara otomatis:

- Mengeksekusi template yang diperlukan CloudFormation
- Menciptakan Peran ECS Service-Linked
- Menyiapkan semua sumber daya IAM yang diperlukan

> **Important**
>
> Jalankan skrip penyiapan yang disediakan apa adanya - jangan mengubah nama sumber daya apa pun di templat atau skrip karena ini dapat memengaruhi proses penerapan.

Untuk Linux:

```
./setup.sh --region <YOUR-Region>
```

Untuk Windows:

```
./setup.ps1 -Region <YOUR-Region>
```

Periksa grup keamanan di VPC Anda

AWS Transform menyebarkan aplikasi ke subnet yang sama dan grup keamanan yang sama dengan instance Aurora PostgreSQL.

Agar aplikasi dapat terhubung ke instance Aurora, pastikan bahwa grup keamanan ini memiliki aturan keluar yang memungkinkan koneksi ke grup keamanan yang sama dan aturan masuk yang memungkinkan koneksi dari grup keamanan yang sama.

Pilih dan konfigurasikan aplikasi

AWS Transform UI menampilkan daftar aplikasi yang dapat digunakan. Pilih aplikasi dan klik Konfigurasi dan Komit untuk membuka jendela konfigurasi tempat Anda dapat menentukan:

- Infrastruktur target untuk aplikasi: pilih ECS atau salah satu instans EC2
- Parameter aplikasi atau contoh

Setelah Anda menyelesaikan dan menutup jendela konfigurasi, AWS Transform:

- Menghasilkan artefak penerapan berdasarkan konfigurasi Anda
- Komit artefak ini ke repositori Anda, termasuk:
  - Skrip penyebaran
  - CloudFormation template

- Contoh konfigurasi CI/CD pipeline (yang dapat Anda gunakan sebagai referensi untuk menyiapkan otomatisasi penerapan)

Setelah komit selesai, status aplikasi berubah menjadi Dikonfigurasi dan berkomitmen. Anda kemudian dapat memilih aplikasi dan menggunakan tombol Deploy di kanan atas untuk memulai penyebaran.

Penyebaran ECS

Semua aplikasi dengan target ECS dikerahkan ke dalam cluster ECS baru yang dibuat di akun Anda. Setiap aplikasi digunakan dalam wadah khusus sendiri - beberapa aplikasi per kontainer tidak didukung. Anda mengonfigurasi parameter kontainer untuk setiap aplikasi secara individual, seperti CPU dan Memori yang diperlukan.

Penyebaran EC2

Anda dapat memilih salah satu instans EC2 yang diusulkan dan dapat mengonfigurasi parameternya seperti jenis instans. Beberapa aplikasi dapat digunakan ke instans EC2 yang sama:

- Untuk menerapkan beberapa aplikasi ke satu instans, pilih instans EC2 target yang sama untuk masing-masing aplikasi ini selama konfigurasi.
- Semua aplikasi yang memiliki instance yang sama dipilih sebagai target dikerahkan ke instance itu dan diberi port acak yang berbeda.
- Perhatikan bahwa jika Anda mengedit parameter instance untuk satu aplikasi, dan kemudian mengedit aplikasi lain yang menargetkan instance yang sama, perubahan akan memengaruhi semua aplikasi yang diterapkan ke instance tersebut.

Keberhasilan penyebaran

Setelah penerapan berhasil, aplikasi Anda berjalan di Aurora PostgreSQL dengan pengurangan biaya lisensi, peningkatan kinerja, dan skalabilitas cloud-native. Pekerjaan tetap terbuka kecuali secara eksplisit dihentikan. Ini memungkinkan opsi untuk kemungkinan penyebaran ulang dengan parameter penerapan yang diperbarui.

Praktik terbaik dan pemecahan masalah

Praktik terbaik, dan masalah umum dan resolusi mereka selama proses modernisasi.

Log aplikasi ECS

CloudWatch log

- Semua log kontainer ECS secara otomatis dikirim ke CloudWatch Log
- Akses log di CloudWatch konsol di bawah Grup Log
- Format penamaan grup log:/aws/ecs/{application-name}
- Setiap instance kontainer membuat aliran log baru di dalam grup

Melihat log

Melalui AWS Konsol:

- Arahkan CloudWatch ke> Grup Log
- Pilih grup log aplikasi Anda
- Pilih aliran log yang relevan untuk melihat log kontainer

Menggunakan AWS CLI:

```
aws logs get-log-events --log-group-name /aws/ecs/your-app-name --log-stream-name your-
stream-name
```

Lokasi log umum

- Log aplikasi: CloudWatch Log
- Acara Layanan ECS: Konsol ECS > Cluster > Layanan > Tab Acara
- Kontainer health/status: Konsol ECS > Cluster > Layanan> Tab Tugas

Manajemen koneksi database

Aplikasi menggunakan variabel lingkungan untuk pengaturan koneksi database

Jika Anda mengalami masalah konektivitas:

- Verifikasi pengaturan koneksi saat ini di variabel lingkungan Anda
- Perbarui variabel lingkungan untuk memodifikasi string koneksi database sesuai kebutuhan

- Perubahan string koneksi dapat dilakukan melalui pembaruan variabel lingkungan tanpa pemindahan aplikasi

Masalah koneksi database

Masalah: Tidak dapat menghubungkan AWS Transform ke SQL Server

Solusi:

- Verifikasi konektivitas jaringan antara AWS Transform dan SQL Server
- Periksa aturan grup keamanan untuk akses port yang tepat (1433)
- Konfirmasikan kredensyal database di Secrets Manager
- Uji izin database dengan pengguna yang dibuat
- Pastikan SQL Server dikonfigurasi untuk otentikasi mode campuran
- Verifikasi rahasia memiliki tag yang diperlukan (Proyek: atx-db-modernisasi, Pemilik: database-connector)

Masalah firewall dan grup keamanan

Masalah: Batas waktu koneksi atau kesalahan “tidak dapat mencapai basis data”

Akar Penyebab: Grup keamanan atau ACL jaringan memblokir lalu lintas

Solusi:

1. Verifikasi Konfigurasi Grup Keamanan:
   - Konfirmasikan grup keamanan SQL Server Anda memiliki aturan masuk yang memungkinkan port 1433 dari grup keamanan Konversi Skema DMS
   - Periksa apakah sumbernya adalah ID grup keamanan (misalnya, sg-0123456789abcdef0), bukan alamat IP
   - Verifikasi grup keamanan Konversi Skema DMS ditentukan dengan benar di Profil Instance
   - Pastikan tidak ada aturan penolakan yang bertentangan
2. Periksa ACL Jaringan:
   - Verifikasi ACL Jaringan tingkat subnet memungkinkan lalu lintas masuk pada port 1433
   - Pastikan ACL Jaringan memungkinkan port fana keluar untuk lalu lintas kembali
   - Periksa subnet database dan ACL Jaringan subnet DMS

3. Verifikasi Konfigurasi VPC:
   - Konfirmasikan instance Konversi Skema DMS dan SQL Server berada di VPC yang sama atau memiliki peering VPC yang tepat
   - Periksa tabel rute memungkinkan lalu lintas antar subnet
   - Pastikan tidak ada peralatan firewall yang memblokir lalu lintas
4. Uji Konektivitas:
   - Luncurkan instance EC2 pengujian di subnet yang sama dengan Konversi Skema DMS
   - Lampirkan grup keamanan yang sama dengan Konversi Skema DMS
   - Uji koneksi ke SQL Server menggunakan telnet atau SQL Server Management Studio
   - Jika pengujian berhasil, masalahnya adalah dengan konfigurasi AWS Transform; jika gagal, masalahnya adalah network/firewall

Kesalahan Umum: Membuka port 1433 hingga 0.0.0. 0/0 (semua sumber) adalah risiko keamanan. Selalu gunakan kontrol akses berbasis grup keamanan untuk membatasi akses hanya ke grup keamanan Konversi Skema DMS.

Masalah konversi skema

Masalah: Konversi skema menunjukkan banyak item tindakan

Solusi:

- Tinjau item tindakan dalam laporan konversi
- Prioritaskan berdasarkan dampak
- Gunakan Pengembang Amazon Q untuk konversi SQL yang kompleks
- Konsultasikan AWS Support untuk panduan
- Pertimbangkan pendekatan bertahap untuk database yang kompleks

Masalah transformasi aplikasi

Masalah: Transformasi aplikasi gagal dibangun

Solusi:

- Tinjau kesalahan build dalam laporan transformasi
- Konfigurasikan NuGet feed pribadi jika diperlukan

- Perbarui referensi paket jika diperlukan
- Periksa Windows-specific dependensi
- Tinjau log transformasi untuk kesalahan terperinci

Masalah migrasi data

Masalah: Validasi migrasi data gagal

Solusi:

- Tinjau laporan validasi untuk kegagalan tertentu
- Periksa pemetaan tipe data
- Verifikasi konfigurasi kolom identitas (DIHASILKAN OLEH DEFAULT vs GENERATED ALWAYS)
- Tinjau ekspresi kolom yang dihitung
- Hubungi AWS Support untuk masalah data yang kompleks

Masalah pembersihan sumber daya

Masalah: Pekerjaan transformasi gagal dengan kesalahan sumber daya

Solusi:

- Periksa sumber daya DMS yang ada (proyek migrasi, penyedia data, profil instance)
- Membersihkan sumber daya yang gagal atau tidak lengkap dari upaya sebelumnya
- Verifikasi rahasia tidak dijadwalkan untuk dihapus
- Periksa kuota layanan untuk DMS dan Aurora PostgreSQL
- Hubungi AWS Support jika pembersihan tidak menyelesaikan masalah

Masalah deployment

Masalah: Aplikasi yang diubah tidak dapat terhubung ke Aurora PostgreSQL

Solusi:

- Verifikasi format string koneksi untuk PostgreSQL
- Periksa aturan grup keamanan

- Verifikasi kredensyal database di Secrets Manager
- Pastikan SSL/TLS dikonfigurasi dengan benar
- Uji koneksi menggunakan psql atau pgAdmin

Mendapatkan bantuan tambahan

Saat menghubungi AWS Support, harap berikan:

- ID pekerjaan transformasi
- AWS ID akun
- Region
- Pesan kesalahan dan tangkapan layar
- Log transformasi (tersedia di konsol AWS Transform)

Keamanan di AWS Transformasi

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 adalah tanggung jawab bersama antara Anda AWS dan Anda. Model tanggung jawab bersama menjelaskan hal ini sebagai keamanan dari cloud dan keamanan dalam cloud:

- Keamanan cloud — AWS bertanggung jawab untuk melindungi infrastruktur yang menjalankan AWS layanan di AWS Cloud. AWS juga memberi Anda layanan yang dapat Anda gunakan dengan aman. Third-party auditor secara teratur menguji dan memverifikasi efektivitas keamanan kami sebagai bagian dari Program Kepatuhan Program AWS Kepatuhan Program AWS . Untuk mempelajari tentang program kepatuhan yang berlaku untuk AWS Transform, lihat AWS Layanan dalam Lingkup oleh AWS Layanan Program Kepatuhan .
- Keamanan di cloud — Tanggung jawab Anda ditentukan oleh AWS layanan yang Anda gunakan. Anda juga bertanggung jawab atas faktor lain, yang mencakup sensitivitas data Anda, persyaratan perusahaan Anda, serta undang-undang dan peraturan yang berlaku.

Dokumentasi ini membantu Anda memahami cara menerapkan model tanggung jawab bersama saat menggunakan AWS Transform. Topik berikut menunjukkan cara mengonfigurasi AWS Transform untuk memenuhi tujuan keamanan dan kepatuhan Anda. Anda juga mempelajari cara menggunakan AWS layanan lain yang membantu Anda memantau dan mengamankan sumber daya AWS Transform Anda.

Topik

Perlindungan data di AWS Transform

Model tanggung jawab AWS bersama berlaku untuk perlindungan data di AWS Transform. Seperti yang dijelaskan dalam model AWS ini, bertanggung jawab untuk melindungi infrastruktur global yang menjalankan semua AWS Cloud. Anda bertanggung jawab untuk mempertahankan kendali atas konten yang di-host pada infrastruktur ini. Anda juga bertanggung jawab atas konfigurasi keamanan dan tugas manajemen untuk Layanan AWS yang Anda gunakan. Untuk informasi selengkapnya tentang privasi data, silakan lihat Pertanyaan Umum Privasi Data. Untuk informasi tentang perlindungan data di Eropa, lihat AWS postingan blog Model Tanggung Jawab Bersama dan GDPR AWS di Blog Keamanan .

Untuk tujuan perlindungan data, kami menyarankan Anda melindungi Akun AWS kredensyal dan mengatur pengguna individu dengan AWS Identity and Access Management ()IAM. Dengan cara ini, setiap pengguna hanya diberi izin yang diperlukan untuk memenuhi tanggung jawab tugas mereka. Kami juga merekomendasikan agar Anda mengamankan data Anda dengan cara-cara berikut ini:

- Gunakan autentikasi multi-faktor (MFA) pada setiap akun.
- Gunakan SSL/TLS untuk berkomunikasi dengan AWS sumber daya. Kami merekomendasikan TLS 1.2 atau versi yang lebih baru.
- Siapkan API dan logging aktivitas pengguna dengan AWS CloudTrail.
- Gunakan solusi AWS enkripsi, bersama dengan semua kontrol keamanan default di dalamnya Layanan AWS.
- Gunakan layanan keamanan terkelola tingkat lanjut seperti Amazon Macie, yang membantu dalam menemukan dan mengamankan data sensitif yang disimpan di dalamnya. 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 menyarankan agar Anda tidak pernah memasukkan informasi rahasia atau sensitif, seperti alamat email pelanggan Anda, ke dalam tag atau bidang teks bentuk bebas seperti bidang Nama. Ini termasuk saat Anda bekerja dengan AWS Transform atau lainnya Layanan AWS menggunakan Konsol Manajemen AWS, API, AWS Command Line Interface (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. Untuk informasi selengkapnya tentang cara AWS Transform menggunakan konten, lihatAWS Transform peningkatan layanan.

AWS Transform menyimpan pertanyaan Anda, tanggapannya, dan konteks tambahan, seperti metadata konsol dan kode di IDE Anda, untuk menghasilkan tanggapan atas pertanyaan Anda. Untuk informasi tentang cara data dienkripsi, lihat. Enkripsi data di AWS Transform Untuk informasi tentang cara AWS menggunakan beberapa pertanyaan yang Anda ajukan AWS Transform dan tanggapannya untuk meningkatkan layanan kami, lihatAWS Transform peningkatan layanan.

Di AWS Transform, data Anda disimpan di AWS Wilayah tempat AWS Transform profil Anda dibuat.

Dengan inferensi silang, permintaan Anda untuk AWS Transform dapat diproses di Wilayah yang berbeda dalam geografi tempat data Anda disimpan. Untuk informasi lebih lanjut, lihat Cross-Region inferensi.

Topik

Cross-region pengolahan di AWS Transform

Bagian berikut menjelaskan bagaimana inferensi lintas wilayah dan panggilan lintas wilayah digunakan untuk menyediakan layanan. AWS Transform

Cross-region inferensi

AWS Transform didukung oleh Amazon Bedrock, dan menggunakan inferensi lintas wilayah untuk mendistribusikan lalu lintas di berbagai tempat Wilayah AWS untuk meningkatkan kinerja dan keandalan inferensi model bahasa besar (LLM). Dengan inferensi lintas wilayah, Anda mendapatkan:

- Peningkatan throughput dan ketahanan selama periode permintaan tinggi
- Peningkatan kinerja
- Akses ke AWS Transform kemampuan dan fitur yang baru diluncurkan yang mengandalkan LLM paling kuat yang dihosting di Amazon Bedrock

Cross-region permintaan inferensi disimpan dalam Wilayah AWS yang merupakan bagian dari geografi tempat data awalnya berada. Misalnya, permintaan yang dibuat dari AWS Transform konfigurasi di AS disimpan Wilayah AWS di AS. Meskipun inferensi lintas wilayah tidak mengubah tempat data Anda disimpan, permintaan dan hasil keluaran Anda dapat berpindah ke luar Wilayah

tempat data awalnya berada. Semua data akan dienkripsi saat dikirim melalui jaringan aman Amazon. Tidak ada biaya tambahan untuk menggunakan inferensi lintas wilayah.

Inferensi lintas wilayah tidak memengaruhi tempat penyimpanan data Anda. Untuk informasi tentang tempat data disimpan saat Anda menggunakan AWS Transform, lihatPerlindungan data di AWS Transform.

Wilayah yang didukung untuk AWS Transform inferensi lintas wilayah

Permintaan tertentu yang Anda buat AWS Transform mungkin memerlukan panggilan lintas wilayah. Tabel berikut menjelaskan Wilayah mana permintaan Anda dapat diarahkan tergantung pada geografi tempat permintaan tersebut berasal.

Wilayah Sumber	Wilayah Tujuan
AS Timur (Virginia Utara) (us-east-1)	AS Timur (Virginia Utara) (us-east-1) US East (Ohio) (us-east-2) AS Barat (Oregon) (us-west-2)
Eropa (Frankfurt) (eu-central-1)	Eropa (Frankfurt) (eu-central-1) Eropa (Stockholm) (eu-north-1) Europe (Milan) (eu-south-1) Eropa (Spanyol) (eu-south-2) Eropa (Irlandia) (eu-west-1) Eropa (Paris) (eu-west-3)
Asia Pasifik (Mumbai) (ap-south-1)	Asia Pasifik (Tokyo) (ap-northeast-1) Asia Pasifik (Seoul) (ap-northeast-2) Asia Pasifik (Osaka) (ap-northeast-3) Asia Pasifik (Mumbai) (ap-south-1) Asia Pasifik (Hyderabad) (ap-south-2)

Wilayah Sumber	Wilayah Tujuan
	Asia Pasifik (Singapura) (ap-southeast-1) Asia Pasifik (Sydney) (ap-southeast-2) Asia Pasifik (Melbourne) (ap-southeast-4)
Asia Pasifik (Tokyo) (ap-northeast-1)	Asia Pasifik (Tokyo) (ap-northeast-1) Asia Pasifik (Osaka) (ap-northeast-3)
Asia Pasifik (Seoul) (ap-northeast-2)	Asia Pasifik (Tokyo) (ap-northeast-1) Asia Pasifik (Seoul) (ap-northeast-2) Asia Pasifik (Osaka) (ap-northeast-3) Asia Pasifik (Mumbai) (ap-south-1) Asia Pasifik (Hyderabad) (ap-south-2) Asia Pasifik (Singapura) (ap-southeast-1) Asia Pasifik (Sydney) (ap-southeast-2) Asia Pasifik (Melbourne) (ap-southeast-4)
Asia Pasifik (Sydney) (ap-southeast-2)	Asia Pasifik (Sydney) (ap-southeast-2) Asia Pasifik (Melbourne) (ap-southeast-4)

Wilayah Sumber	Wilayah Tujuan
Eropa (London) (eu-west-2)	Eropa (Frankfurt) (eu-central-1) Eropa (Stockholm) (eu-north-1) Europe (Milan) (eu-south-1) Eropa (Spanyol) (eu-south-2) Eropa (Irlandia) (eu-west-1) Eropa (London) (eu-west-2) Eropa (Paris) (eu-west-3)
Kanada (Pusat) (ca-central-1)	Wilayah AWS Komersial+Kanada (Tengah) (ca-central-1)
Amerika Selatan (Sao Paulo) (sa-east-1)	Wilayah AWS Komersial+Amerika Selatan (São Paulo) (sa-east-1)

Untuk daftar lengkap Wilayah tempat Anda dapat menggunakan AWS Transform, lihatWilayah yang Didukung untuk AWS Transform.

Cross-Region pengetahuan

Ketika Anda mengajukan pertanyaan umum tentang AWS Transform layanan, alur kerja transformasi, atau AWS dokumentasi terkait, AWS Transform mungkin mengajukan permintaan lintas wilayah ke US East (Virginia) (us-east-1) untuk wilayah AS atau Eropa (Frankfurt) (eu-central-1) untuk semua wilayah lain untuk mengambil dokumentasi dan memenuhi permintaan Anda. Misalnya, ketika Anda mengajukan pertanyaan tentang cara menggunakan AWS layanan lain seperti Lambda, AWS Transform mungkin membuat panggilan lintas wilayah untuk mengambil AWS dokumentasi yang relevan untuk menanggapi pertanyaan Anda. Tabel berikut menjelaskan Wilayah mana permintaan Anda dapat diarahkan tergantung pada geografi tempat permintaan tersebut berasal.

Wilayah Sumber	Wilayah Tujuan
AS Timur (Virginia Utara) (us-east-1)	AS Timur (Virginia Utara) (us-east-1)

Wilayah Sumber	Wilayah Tujuan
Eropa (Frankfurt) (eu-central-1)	Eropa (Frankfurt) (eu-central-1)
Eropa (London) (eu-west-2)	Eropa (Frankfurt) (eu-central-1)
Asia Pasifik (Tokyo) (ap-northeast-1)	Eropa (Frankfurt) (eu-central-1)
Asia Pasifik (Sydney) (ap-southeast-2)	Eropa (Frankfurt) (eu-central-1)
Asia Pasifik (Seoul) (ap-northeast-2)	Eropa (Frankfurt) (eu-central-1)
Asia Pasifik (Mumbai) (ap-south-1)	Eropa (Frankfurt) (eu-central-1)
Kanada (Pusat) (ca-central-1)	Eropa (Frankfurt) (eu-central-1)
Amerika Selatan (Sao Paulo) (sa-east-1)	Eropa (Frankfurt) (eu-central-1)

Pengaturan ini diaktifkan secara default. Administrator akun dapat mengubah pengaturan ini. Menonaktifkan fitur ini mengakibatkan hilangnya akses ke fitur yang diperlukan AWS Transform untuk mengambil pengetahuan dari wilayah lain. Ini mungkin menghasilkan tanggapan yang kurang akurat.

Untuk menonaktifkan panggilan lintas wilayah yang dilakukan oleh AWS Transform:

1. Saat pertama kali mengatur AWS Transform, navigasikan ke halaman Memulai dan selesaikan konfigurasi. Untuk AWS Transform konfigurasi yang ada, navigasikan ke halaman Pengaturan.
2. Alihkan Aktifkan panggilan lintas wilayah untuk menjawab pertanyaan umum AWS terkait ke posisi mati.

AWS Transform untuk inferensi lintas wilayah Windows

Tabel berikut menunjukkan sumber Wilayah tempat Anda dapat memanggil profil inferensi dan Wilayah tujuan tempat permintaan dapat dirutekan:

Wilayah Sumber	Wilayah Tujuan Inferensi
AS Timur (Virginia Utara) (us-east-1)	AS Timur (Virginia Utara) (us-east-1) US East (Ohio) (us-east-2)

Wilayah Sumber	Wilayah Tujuan Inferensi
	AS Barat (Oregon) (us-west-2)
Eropa (Frankfurt) (eu-central-1)	Eropa (Frankfurt) (eu-central-1) Eropa (Stockholm) (eu-north-1) Europe (Milan) (eu-south-1) Eropa (Spanyol) (eu-south-2) Eropa (Irlandia) (eu-west-1) Eropa (Paris) (eu-west-3)
Asia Pasifik (Tokyo) (ap-northeast-1)	Asia Pasifik (Tokyo) (ap-northeast-1) Asia Pasifik (Osaka) (ap-northeast-3)
Asia Pasifik (Sydney) (ap-southeast-2)	Asia Pasifik (Sydney) (ap-southeast-2) Asia Pasifik (Melbourne) (ap-southeast-4)
Asia Pasifik (Seoul) (ap-northeast-2)	Semua wilayah komersial
Asia Pasifik (Mumbai) (ap-south-1)	Semua wilayah komersial
Kanada (Pusat) (ca-central-1)	Semua wilayah komersial

AWS Transform untuk Migrasi inferensi lintas wilayah

Tabel berikut menunjukkan sumber Wilayah tempat Anda dapat memanggil profil inferensi dan Wilayah tujuan tempat permintaan dapat dirutekan:

Wilayah Sumber	Wilayah Tujuan Inferensi
AS Timur (Virginia Utara) (us-east-1)	Semua wilayah komersial
Eropa (Frankfurt) (eu-central-1)	Semua wilayah komersial

Wilayah Sumber	Wilayah Tujuan Inferensi
Eropa (London) (eu-west-2)	Semua wilayah komersial
Asia Pasifik (Tokyo) (ap-northeast-1)	Asia Pasifik (Tokyo) (ap-northeast-1) Asia Pasifik (Osaka) (ap-northeast-3)
Asia Pasifik (Sydney) (ap-southeast-2)	Asia Pasifik (Sydney) (ap-southeast-2) Asia Pasifik (Melbourne) (ap-southeast-4)
Asia Pasifik (Seoul) (ap-northeast-2)	Semua wilayah komersial
Asia Pasifik (Mumbai) (ap-south-1)	Semua wilayah komersial
Kanada (Pusat) (ca-central-1)	Semua wilayah komersial

AWS Transform untuk inferensi lintas wilayah mainframe

AWS Transform untuk mainframe menggunakan inferensi lintas wilayah geografis Amazon Bedrock. Ini berarti bahwa beberapa panggilan Anda mungkin diarahkan ke Wilayah AWS luar wilayah sumber Anda. Tabel berikut menunjukkan sumber Wilayah tempat Anda dapat memanggil profil inferensi dan Wilayah tujuan tempat permintaan dapat dirutekan:

Wilayah Sumber	Wilayah Tujuan Inferensi
AS Timur (Virginia Utara) (us-east-1)	AS Timur (Virginia Utara) (us-east-1) US East (Ohio) (us-east-2) AS Barat (Oregon) (us-west-2)
Eropa (Frankfurt) (eu-central-1)	Eropa (Frankfurt) (eu-central-1) Eropa (Stockholm) (eu-north-1) Europe (Milan) (eu-south-1) Eropa (Spanyol) (eu-south-2)

Wilayah Sumber	Wilayah Tujuan Inferensi
	Eropa (Irlandia) (eu-west-1) Eropa (Paris) (eu-west-3)
Eropa (London) (eu-west-2)	Eropa (Frankfurt) (eu-central-1) Eropa (Stockholm) (eu-north-1) Europe (Milan) (eu-south-1) Eropa (Spanyol) (eu-south-2) Eropa (Irlandia) (eu-west-1) Eropa (London) (eu-west-2) Eropa (Paris) (eu-west-3)
Asia Pasifik (Tokyo) (ap-northeast-1)	Asia Pasifik (Tokyo) (ap-northeast-1) Asia Pasifik (Osaka) (ap-northeast-3)
Asia Pasifik (Sydney) (ap-southeast-2)	Asia Pasifik (Sydney) (ap-southeast-2) Asia Pasifik (Melbourne) (ap-southeast-4)
Asia Pasifik (Seoul) (ap-northeast-2)	Semua wilayah komersial
Asia Pasifik (Mumbai) (ap-south-1)	Semua wilayah komersial
Kanada (Pusat) (ca-central-1)	Kanada (Pusat) (ca-central-1) AS Timur (Virginia Utara) (us-east-1) US East (Ohio) (us-east-2) AS Barat (Oregon) (us-west-2)
Amerika Selatan (Sao Paulo) (sa-east-1)	Semua wilayah komersial

AWS Transform Khusus

AWS Transform kustom menggunakan inferensi lintas wilayah geografis Amazon Bedrock. Ini berarti bahwa beberapa panggilan Anda mungkin diarahkan ke Wilayah AWS luar wilayah sumber Anda. Tabel berikut menunjukkan sumber Wilayah tempat Anda dapat memanggil profil inferensi dan Wilayah tujuan tempat permintaan dapat dirutekan:

Wilayah Sumber	Wilayah Tujuan Inferensi
AS Timur (Virginia Utara) (us-east-1)	AS Timur (Virginia Utara) (us-east-1) US East (Ohio) (us-east-2) AS Barat (Oregon) (us-west-2)
Eropa (Frankfurt) (eu-central-1)	Eropa (Frankfurt) (eu-central-1) Eropa (Stockholm) (eu-north-1) Europe (Milan) (eu-south-1) Eropa (Spanyol) (eu-south-2) Eropa (Irlandia) (eu-west-1) Eropa (Paris) (eu-west-3)
Eropa (London) (eu-west-2)	Eropa (Frankfurt) (eu-central-1) Eropa (Stockholm) (eu-north-1) Europe (Milan) (eu-south-1) Eropa (Spanyol) (eu-south-2) Eropa (Irlandia) (eu-west-1) Eropa (London) (eu-west-2) Eropa (Paris) (eu-west-3)
Asia Pasifik (Tokyo) (ap-northeast-1)	Semua wilayah komersial

Wilayah Sumber	Wilayah Tujuan Inferensi
Asia Pasifik (Sydney) (ap-southeast-2)	Asia Pasifik (Sydney) (ap-southeast-2) Asia Pasifik (Melbourne) (ap-southeast-4)
Asia Pasifik (Seoul) (ap-northeast-2)	Semua wilayah komersial
Asia Pasifik (Mumbai) (ap-south-1)	Semua wilayah komersial
Kanada (Pusat) (ca-central-1)	Kanada (Pusat) (ca-central-1) AS Timur (Virginia Utara) (us-east-1) US East (Ohio) (us-east-2) AS Barat (Oregon) (us-west-2)

Enkripsi data di AWS Transform

Topik ini memberikan informasi khusus AWS Transform tentang enkripsi dalam perjalanan dan enkripsi saat istirahat.

AWS Transform menyediakan enkripsi secara default untuk melindungi data pelanggan sensitif saat istirahat dengan enkripsi menggunakan kunci yang AWS dimiliki.

Jenis enkripsi

Enkripsi saat bergerak

Komunikasi antara pelanggan dan dan antara AWS Transform AWS Transform dan dependensi hilirnya dilindungi menggunakan TLS 1.2 atau koneksi yang lebih tinggi.

> **Important**
>
> Saat Anda Menghubungkan akun penemuan, AWS Transform akan membuat bucket Amazon S3 atas nama Anda di akun tersebut. Bucket itu tidak `SecureTransport` diaktifkan secara default. Jika Anda ingin kebijakan bucket menyertakan transportasi aman, Anda harus memperbarui kebijakan tersebut. Untuk informasi selengkapnya, lihat Praktik terbaik keamanan untuk Amazon S3.

Enkripsi saat diam

AWS Transform menyimpan data saat istirahat menggunakan Amazon DynamoDB dan Amazon Simple Storage Service (Amazon S3). Data saat istirahat dienkripsi menggunakan solusi AWS enkripsi secara default. AWS Transform mengenkripsi data Anda dengan enkripsi menggunakan kunci AWS milik dari AWS Key Management Service (AWS KMS). Anda tidak perlu mengambil tindakan apa pun untuk melindungi kunci AWS terkelola yang mengenkripsi data Anda. Untuk informasi selengkapnya, lihat kunci yang AWS dimiliki di Panduan AWS Key Management Service Pengembang.

Jenis data	AWS enkripsi kunci yang dimiliki	Enkripsi kunci yang dikelola pelanggan (Opsional)
Data bucket pelanggan Masukan dan output pelanggan seperti kode dan dokumentasi yang disimpan dalam bucket Amazon S3	Diaktifkan	Diaktifkan
Toko Artifak Artefak menengah sebagai bagian dari transformasi kode yang disimpan dalam ember S3	Diaktifkan	Diaktifkan
Tujuan Job Maksud pelanggan untuk pekerjaan yang disimpan dalam bucket Amazon S3	Diaktifkan	Diaktifkan
Pesan obrolan Pesan antara pelanggan dan AWS Transform disimpan dalam bucket Amazon S3	Diaktifkan	Diaktifkan
Basis Pengetahuan Obrolan	Diaktifkan	Diaktifkan

Jenis data	AWS enkripsi kunci yang dimiliki	Enkripsi kunci yang dikelola pelanggan (Opsional)
Data terindeks yang relevan dengan AWS Transform dan obrolan pelanggan disimpan di Amazon OpenSearch dan diproses melalui Bedrock AWS		

Catatan: Pelanggan dapat mendaftarkan Customer Managed Key (CMK) mereka sendiri untuk digunakan untuk mengenkripsi semua tipe data di atas.

Kunci yang dikelola pelanggan adalah kunci KMS di AWS akun Anda yang Anda buat, miliki, dan kelola untuk secara langsung mengontrol akses ke data Anda dengan mengontrol akses ke kunci KMS. Hanya kunci simetris yang didukung. Untuk informasi tentang membuat kunci KMS Anda sendiri, lihat Membuat kunci di Panduan AWS Key Management Service Pengembang.

Saat Anda menggunakan kunci yang dikelola pelanggan, AWS Transform gunakan hibah KMS, yang memungkinkan pengguna, peran, atau aplikasi yang berwenang untuk menggunakan kunci KMS. Ketika AWS Transform administrator memilih untuk menggunakan kunci terkelola pelanggan untuk enkripsi selama konfigurasi, hibah dibuat untuk mereka. Hibah inilah yang memungkinkan pengguna akhir untuk menggunakan kunci enkripsi untuk enkripsi data saat istirahat. Untuk informasi lebih lanjut tentang hibah, lihat Hibah di. AWS KMS

AWS Kunci yang Dimiliki (Default)

> **Note**
>
> AWS kunci yang dimiliki — AWS Transform menggunakan kunci ini secara default untuk secara otomatis mengenkripsi data yang dapat diidentifikasi secara pribadi. Anda tidak dapat melihat, mengelola, atau menggunakan kunci yang AWS dimiliki, atau mengaudit penggunaannya. Namun, Anda tidak perlu mengambil tindakan apa pun atau mengubah program apa pun untuk melindungi kunci yang mengenkripsi data Anda. Untuk informasi selengkapnya, lihat kunci yang AWS dimiliki di Panduan Pengembang Layanan Manajemen AWS Kunci.

Enkripsi data saat istirahat secara default membantu mengurangi overhead operasional dan kompleksitas yang terlibat dalam melindungi data sensitif. Pada saat yang sama, ini memungkinkan Anda untuk membangun aplikasi aman yang memenuhi kepatuhan enkripsi yang ketat dan persyaratan peraturan.

Meskipun Anda tidak dapat menonaktifkan lapisan enkripsi ini atau memilih jenis enkripsi alternatif, Anda dapat menambahkan lapisan enkripsi kedua di atas kunci enkripsi yang ada AWS dengan memilih kunci yang dikelola pelanggan saat Anda membuat transformasi:

Kunci terkelola pelanggan (Opsional)

Kunci terkelola pelanggan — AWS Transform mendukung penggunaan kunci terkelola pelanggan simetris yang Anda buat, miliki, dan kelola untuk menambahkan lapisan enkripsi kedua di atas enkripsi yang ada menggunakan kunci yang AWS dimiliki. Karena Anda memiliki kontrol penuh atas lapisan enkripsi ini, Anda dapat melakukan tugas-tugas seperti:

- Menetapkan dan memelihara kebijakan utama
- Menetapkan dan memelihara kebijakan dan hibah IAM
- Mengaktifkan dan menonaktifkan kebijakan utama
- Memutar bahan kriptografi kunci
- Menambahkan tanda
- Membuat alias kunci
- Memutar bahan kriptografi kunci
- Menambahkan tanda
- Membuat alias kunci
- Kunci penjadwalan untuk penghapusan
- Untuk informasi selengkapnya, lihat kunci terkelola pelanggan di Panduan Pengembang Layanan Manajemen AWS Kunci.

> **Note**
>
> AWS Transform secara otomatis mengaktifkan enkripsi saat istirahat menggunakan kunci yang AWS dimiliki untuk melindungi data yang dapat diidentifikasi secara pribadi tanpa biaya. Namun, AWS KMS biaya berlaku untuk menggunakan kunci yang dikelola pelanggan. Untuk informasi lebih lanjut tentang harga, lihat AWS Key Management Service harga.

Untuk informasi lebih lanjut tentang AWS KMS, lihat Apa itu AWS Key Management Service?

Bagaimana AWS Transform menggunakan hibah di AWS Key Management Service

AWS Transform membutuhkan hibah untuk menggunakan kunci yang dikelola pelanggan Anda.

Saat Anda membuat [Nama Sumber Daya] yang dienkripsi dengan kunci yang dikelola pelanggan, AWS Transform buat hibah atas nama Anda dengan mengirimkan CreateGrant permintaan ke. AWS Key Management Service Hibah AWS Key Management Service digunakan untuk memberikan AWS Transform akses ke kunci KMS di akun pelanggan.

AWS Transform memerlukan hibah untuk menggunakan kunci yang dikelola pelanggan Anda untuk operasi internal berikut:

- Kirim permintaan [KMS API] AWS KMS untuk memverifikasi bahwa ID kunci KMS terkelola pelanggan simetris yang dimasukkan saat membuat [Nama Sumber Daya] valid.
- Kirim permintaan [KMS API] AWS KMS untuk menghasilkan kunci data yang dienkripsi oleh kunci yang dikelola pelanggan Anda.
- Kirim permintaan [KMS API] AWS KMS untuk mendekripsi kunci data terenkripsi sehingga dapat digunakan untuk mengenkripsi data Anda.

Anda dapat mencabut akses ke hibah, atau menghapus akses layanan ke kunci yang dikelola pelanggan kapan saja. Jika Anda melakukannya, AWS Transform tidak akan dapat mengakses data apa pun yang dienkripsi oleh kunci yang dikelola pelanggan, yang memengaruhi operasi yang bergantung pada data tersebut. Misalnya, jika Anda mencoba mendapatkan [Nama Sumber Daya] dari [Nama Sumber Daya] terenkripsi yang tidak AWS Transform dapat diakses, maka operasi akan mengembalikan kesalahan AccessDeniedException .

Buat kunci yang dikelola pelanggan

Anda dapat membuat kunci terkelola pelanggan simetris dengan menggunakan AWS Management Console, atau AWS Key Management Service API.

Untuk membuat kunci terkelola pelanggan simetris, ikuti langkah-langkah untuk Membuat kunci terkelola pelanggan simetris di Panduan AWS Key Management Service Pengembang.

Untuk menggunakan kunci terkelola pelanggan dengan AWS Transform sumber daya Anda, operasi API berikut harus diizinkan dalam kebijakan kunci:

kms: CreateGrant — Menambahkan hibah ke kunci yang dikelola pelanggan. Memberikan akses kontrol ke kunci KMS tertentu, yang memungkinkan akses ke operasi AWS Transform hibah memerlukan. Untuk informasi selengkapnya tentang Menggunakan Hibah, lihat Panduan AWS Key Management Service Pengembang.

Hal ini memungkinkan AWS Transform untuk melakukan hal berikut:

- Panggil [KMS API (GenerateDataKeyWithoutPlainText/GenerateDatakey) untuk menghasilkan kunci data terenkripsi dan menyimpannya, karena kunci data tidak segera digunakan untuk mengenkripsi.
- Panggil [KMS API (Decrypt)] untuk menggunakan kunci data terenkripsi yang disimpan untuk mengakses data terenkripsi.
- Siapkan kepala sekolah yang pensiun untuk memungkinkan layanan. RetireGrant
- kms: DescribeKey — Memberikan detail kunci yang dikelola pelanggan untuk memungkinkan [Nama Layanan] memvalidasi kunci.

Enkripsi untuk modernisasi SQL

Selama proses modernisasi SQL Server AWS Transform menciptakan sumber daya di akun Anda, seperti instans EC2, cluster ECS, gambar ECR, dan objek S3. Anda memiliki opsi untuk menyediakan kunci terkelola pelanggan untuk mengenkripsi data di akun Anda. Anda harus menandai kunci KMS Anda dengan kunci CreatedFordan nilai AWSTransform. Contoh kebijakan ini mencantumkan izin minimum yang diperlukan dalam kebijakan kunci jika Anda tidak memiliki kebijakan kunci default. Jika Anda memiliki kebijakan kunci default, Anda masih perlu memperbarui kunci KMS secara manual selain kebijakan kunci default Anda untuk mengizinkan ECS menggunakan kunci Anda.

```
"Statement": [
   {
     "Effect": "Allow",
     "Principal": {
       "AWS": "arn:aws:iam::111122223333:root"
     },
     "Action": [
       "kms:Decrypt",
       "kms:GenerateDataKey"
     ],
     "Resource": "*",
     "Condition": {
```

```
        "StringLike": {
          "kms:ViaService": "s3.*.amazonaws.com",
          "kms:EncryptionContext:aws-transform": "*"
        },
        "ArnLike": {
          "aws:PrincipalArn": "arn:aws:iam::*:role/*AWSTransform*"
        }
      }
    },
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::111122223333:root"
      },
      "Action": [
        "kms:Decrypt",
        "kms:GenerateDataKey",
        "kms:GenerateDataKeyWithoutPlaintext",
        "kms:ReEncrypt*"
      ],
      "Resource": "*",
      "Condition": {
        "StringLike": {
          "kms:ViaService": "ec2.*.amazonaws.com"
        },
        "ArnLike": {
          "aws:PrincipalArn": "arn:aws:iam::*:role/*AWSTransform*"
        }
      }
    },
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::111122223333:root"
      },
      "Action": "kms:CreateGrant",
      "Resource": "*",
      "Condition": {
        "StringLike": {
          "kms:ViaService": [
            "ec2.*.amazonaws.com",
            "ecr.*.amazonaws.com"
          ]
        },
```

```
        "Bool": {
          "kms:GrantIsForAWSResource": "true"
        },
        "ArnLike": {
          "aws:PrincipalArn": "arn:aws:iam::*:role/*AWSTransform*"
        }
      }
    },
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "fargate.amazonaws.com"
      },
      "Action": "kms:GenerateDataKeyWithoutPlaintext",
      "Resource": "*",
      "Condition": {
        "StringLike": {
          "kms:EncryptionContext:aws:ecs:clusterName": "AWSTransform*"
        }
      }
    },
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "fargate.amazonaws.com"
      },
      "Action": "kms:CreateGrant",
      "Resource": "*",
      "Condition": {
        "StringLike": {
          "kms:EncryptionContext:aws:ecs:clusterName": "AWSTransform*"
        },
        "ForAllValues:StringEquals": {
          "kms:GrantOperations": "Decrypt"
        }
      }
    }
  ]
```

Enkripsi untuk migrasi database

Selama proses migrasi database, AWS Transform gunakan AWS Database Migration Service (DMS) untuk memigrasikan data database Anda. Anda dapat memberikan kunci terkelola pelanggan secara opsional untuk mengenkripsi data database Anda selama migrasi.

Kebijakan berikut berisi tiga pernyataan. Pernyataan pertama memberikan izin akun root untuk kunci tersebut. AWS KMS Pernyataan kedua memungkinkan peran konektor untuk memvalidasi kunci selama penemuan. Pernyataan ketiga memungkinkan peran konektor untuk menggunakan kunci untuk operasi DMS.

```
"Statement": [
   {
     "Sid": "Enable IAM User Permissions",
     "Effect": "Allow",
     "Principal": { "AWS": "arn:aws:iam::111122223333:root" },
     "Action": "kms:*",
     "Resource": "*"
   },
   {
     "Sid": "Allow connector role to validate key during discovery",
     "Effect": "Allow",
     "Principal": { "AWS": "connector-role-arn" },
     "Action": "kms:DescribeKey",
     "Resource": "*"
   },
   {
     "Sid": "Allow connector role to use key via DMS",
     "Effect": "Allow",
     "Principal": { "AWS": "connector-role-arn" },
     "Action": [
       "kms:CreateGrant",
       "kms:Decrypt",
       "kms:DescribeKey",
       "kms:Encrypt",
       "kms:GenerateDataKey*",
       "kms:ReEncrypt*"
     ],
     "Resource": "*",
     "Condition": {
       "StringEquals": {
         "kms:ViaService": "dms.us-east-1.amazonaws.com"
       }
```

```
      }
    }
  ]
```

Menggunakan kunci KMS yang dikelola pelanggan

Setelah membuat kunci KMS yang dikelola pelanggan, AWS Transform administrator harus menyediakan kunci di AWS Transform konsol untuk menggunakannya untuk mengenkripsi data.

Untuk menyiapkan kunci terkelola pelanggan untuk mengenkripsi data AWS Transform, administrator memerlukan izin untuk digunakan. AWS KMS

Untuk menggunakan fungsionalitas yang dienkripsi dengan kunci yang dikelola pelanggan, pengguna memerlukan izin AWS Transform untuk mengizinkan mengakses kunci yang dikelola pelanggan.

Jika Anda melihat kesalahan yang terkait dengan hibah KMS saat menggunakan AWS Transform, Anda mungkin perlu memperbarui izin Anda untuk memungkinkan AWS Transform untuk membuat hibah. Untuk secara otomatis mengkonfigurasi izin yang diperlukan, buka AWS Transform konsol dan pilih Perbarui izin di spanduk di bagian atas halaman. AWS Transform Untuk memungkinkan membuat hibah, Anda perlu memperbarui izin di halaman AWS Transform konsol.

Izinkan AWS Transform akses ke kunci yang dikelola pelanggan

Contoh pernyataan kebijakan berikut memberikan izin kepada pengguna untuk mengakses fitur yang dienkripsi dengan kunci yang dikelola pelanggan dengan mengizinkan AWS Transform akses ke kunci. Kebijakan ini wajib digunakan AWS Transform jika administrator telah menyiapkan kunci terkelola pelanggan untuk enkripsi.

> **Note**
>
> Informasi ini tidak berlaku untuk alur kerja penerapan modernisasi SQL Server menggunakan CMK.

```
"Statement": [
   {
     "Sid": "QKMSDecryptGenerateDataKeyPermissions",
     "Effect": "Allow",
     "Action": [
       "kms:Decrypt",
       "kms:GenerateDataKey",
```

```
        "kms:GenerateDataKeyWithoutPlaintext",
        "kms:ReEncryptFrom",
        "kms:ReEncryptTo"
      ],
      "Resource": [
        "arn:aws:kms:arn:aws::111122223333:key/[[key_id]]"
      ],
      "Condition": {
        "StringLike": {
          "kms:ViaService": [
            "q.us-east-1.amazonaws.com"
          ]
        }
      }
    }
  ]
```

AWS Transform peningkatan layanan

Untuk membantu AWS Transform memberikan informasi yang paling relevan, kami dapat menggunakan konten tertentu dari AWS Transform, seperti pertanyaan yang Anda ajukan AWS Transform dan tanggapannya, untuk peningkatan layanan. Halaman ini menjelaskan konten apa yang kami gunakan dan cara memilih keluar.

AWS Transform konten yang digunakan untuk peningkatan layanan

Kami dapat menggunakan konten tertentu AWS Transform untuk peningkatan layanan. AWS Transform dapat menggunakan konten ini, misalnya, untuk memberikan tanggapan yang lebih baik terhadap pertanyaan umum, memperbaiki masalah AWS Transform operasional, untuk de-bugging, atau untuk pelatihan model.

Konten yang AWS mungkin digunakan untuk peningkatan layanan mencakup, misalnya, pertanyaan Anda AWS Transform dan tanggapan serta kode yang AWS Transform dihasilkan.

Cara memilih keluar

Cara Anda memilih untuk tidak AWS Transform menggunakan konten untuk peningkatan layanan tergantung pada lingkungan tempat Anda menggunakan AWS Transform.

Untuk AWS Transform mengonfigurasi kebijakan opt-out layanan AI di. AWS Organizations Untuk informasi selengkapnya, silakan lihat kebijakan penolakan layanan AI di AWS Organizations Panduan Pengguna.

Untuk memilih keluar dari berbagi konten Anda di Visual Studio IDE, gunakan prosedur berikut.

Buka Alat -> Opsi -> AWS Toolkit -> Amazon Q

Beralih Bagikan Konten Amazon Q dengan AWS Benar atau Salah.

Untuk memilih keluar dari berbagi data telemetri Anda di AWS Toolkit untukVisual Studio, gunakan prosedur ini:

1. Di bawah Alat, pilih Opsi.
2. Di panel Opsi, pilih AWS Toolkit, lalu pilih Umum.
3. Hapus pilihan Izinkan AWS Toolkit untuk mengumpulkan informasi penggunaan.

Identitas dan manajemen akses untuk AWS Transformasi

AWS Identity and Access Management (IAM) adalah Layanan AWS yang membantu administrator mengontrol akses ke AWS sumber daya dengan aman. Administrator IAM mengontrol siapa yang dapat diautentikasi (masuk) dan diberi wewenang (memiliki izin) untuk menggunakan sumber daya Transform. AWS IAM adalah Layanan AWS yang dapat Anda gunakan tanpa biaya tambahan.

Topik

Audiens

Cara Anda menggunakan AWS Identity and Access Management (IAM) berbeda berdasarkan peran Anda:

- Pengguna layanan - minta izin dari administrator Anda jika Anda tidak dapat mengakses fitur (lihat Pemecahan masalah AWS Mengubah identitas dan akses)
- Administrator layanan - tentukan akses pengguna dan mengirimkan permintaan izin (lihat Bagaimana AWS Transform bekerja dengan IAM)
- Administrator IAM - tulis kebijakan untuk mengelola akses (lihat Identity-based contoh kebijakan untuk AWS Transformasi)

Mengautentikasi dengan identitas

Otentikasi adalah cara Anda masuk AWS menggunakan kredensyal identitas Anda. Anda harus diautentikasi sebagai Pengguna root akun AWS, pengguna IAM, atau dengan mengambil peran IAM.

Anda dapat masuk sebagai identitas federasi menggunakan kredensil dari sumber identitas seperti AWS IAM Identity Center (Pusat Identitas IAM), otentikasi masuk tunggal, atau kredensional. Google/ Facebook Untuk informasi selengkapnya tentang cara masuk, lihat Cara masuk ke Akun AWS Anda dalam Panduan Pengguna AWS Sign-In .

Untuk akses terprogram, AWS sediakan SDK dan CLI untuk menandatangani permintaan secara kriptografis. Untuk informasi selengkapnya, lihat AWS Signature Version 4 untuk permintaan API dalam Panduan Pengguna IAM.

Akun AWS pengguna root

Saat Anda membuat Akun AWS, Anda mulai dengan satu identitas masuk yang disebut pengguna Akun AWS root yang memiliki akses lengkap ke semua Layanan AWS dan sumber daya. Kami sangat menyarankan agar Anda tidak menggunakan pengguna root untuk tugas sehari-hari. Untuk tugas yang memerlukan kredensial pengguna root, lihat Tugas yang memerlukan kredensial pengguna root dalam Panduan Pengguna IAM.

Identitas terfederasi

Sebagai praktik terbaik, mewajibkan pengguna manusia untuk menggunakan federasi dengan penyedia identitas untuk mengakses Layanan AWS menggunakan kredensi sementara.

Identitas federasi adalah pengguna dari direktori perusahaan Anda, penyedia identitas web, atau Directory Service yang mengakses Layanan AWS menggunakan kredensil dari sumber identitas. Identitas terfederasi mengambil peran yang memberikan kredensial sementara.

Untuk manajemen akses terpusat, kami menyarankan AWS IAM Identity Center. Untuk informasi selengkapnya, lihat Apa itu Pusat Identitas IAM? dalam Panduan Pengguna AWS IAM Identity Center .

Pengguna dan grup IAM

Pengguna IAM adalah identitas dengan izin khusus untuk satu orang atau aplikasi. Sebaiknya gunakan kredensial sementara alih-alih pengguna IAM dengan kredensial jangka panjang. Untuk informasi selengkapnya, lihat Mewajibkan pengguna manusia untuk menggunakan federasi dengan penyedia identitas untuk mengakses AWS menggunakan kredensi sementara di Panduan Pengguna IAM.

Grup IAM menentukan kumpulan pengguna IAM dan mempermudah pengelolaan izin untuk pengguna dalam jumlah besar. Untuk mempelajari selengkapnya, lihat Kasus penggunaan untuk pengguna IAM dalam Panduan Pengguna IAM.

Peran IAM

Peran IAM adalah identitas dengan izin khusus yang menyediakan kredensial sementara. Anda dapat mengambil peran dengan beralih dari pengguna ke peran IAM (konsol) atau dengan memanggil operasi AWS CLI atau AWS API. Untuk informasi selengkapnya, lihat Metode untuk mengambil peran dalam Panduan Pengguna IAM.

Peran IAM berguna untuk akses pengguna terfederasi, izin pengguna IAM sementara, akses lintas akun, akses lintas layanan, dan aplikasi yang berjalan di Amazon EC2. Untuk informasi selengkapnya, lihat Akses sumber daya lintas akun di IAM dalam Panduan Pengguna IAM.

Mengelola akses menggunakan kebijakan

Anda mengontrol akses AWS dengan membuat kebijakan dan melampirkannya ke AWS identitas atau sumber daya. Kebijakan menentukan izin saat dikaitkan dengan identitas atau sumber daya. AWS mengevaluasi kebijakan ini ketika kepala sekolah membuat permintaan. Sebagian besar kebijakan disimpan AWS sebagai dokumen JSON. Untuk informasi selengkapnya tentang dokumen kebijakan JSON, lihat Gambaran umum kebijakan JSON dalam Panduan Pengguna IAM.

Menggunakan kebijakan, administrator menentukan siapa yang memiliki akses ke apa dengan mendefinisikan principal mana yang dapat melakukan tindakan pada sumber daya apa, dan dalam kondisi apa.

Secara default, pengguna dan peran tidak memiliki izin. Administrator IAM membuat kebijakan IAM dan menambahkannya ke peran, yang kemudian dapat diambil oleh pengguna. Kebijakan IAM mendefinisikan izin terlepas dari metode yang Anda gunakan untuk melakukan operasinya.

Identity-based kebijakan

Identity-based kebijakan adalah dokumen kebijakan izin JSON yang Anda lampirkan ke identitas (pengguna, grup, atau peran). Kebijakan ini mengontrol tindakan apa yang bisa dilakukan oleh identitas tersebut, terhadap sumber daya yang mana, dan dalam kondisi apa. Untuk mempelajari cara membuat kebijakan berbasis identitas, lihat Tentukan izin IAM kustom dengan kebijakan yang dikelola pelanggan dalam Panduan Pengguna IAM.

Identity-based kebijakan dapat berupa kebijakan inline (disematkan langsung ke dalam satu identitas) atau kebijakan terkelola (kebijakan mandiri yang dilampirkan pada beberapa identitas). Untuk mempelajari cara memilih antara kebijakan terkelola dan kebijakan inline, lihat Pilih antara kebijakan terkelola dan kebijakan inline dalam Panduan Pengguna IAM.

Resource-based kebijakan

Resource-based kebijakan adalah dokumen kebijakan JSON yang Anda lampirkan ke sumber daya. Contohnya termasuk kebijakan kepercayaan peran IAM dan kebijakan bucket Amazon S3. Dalam layanan yang mendukung kebijakan berbasis sumber daya, administrator layanan dapat menggunakannya untuk mengontrol akses ke sumber daya tertentu. Anda harus menentukan principal dalam kebijakan berbasis sumber daya.

Resource-based kebijakan adalah kebijakan inline yang terletak di layanan tersebut. Anda tidak dapat menggunakan kebijakan AWS terkelola dari IAM dalam kebijakan berbasis sumber daya.

Jenis-jenis kebijakan lain

AWS mendukung jenis kebijakan tambahan yang dapat menetapkan izin maksimum yang diberikan oleh jenis kebijakan yang lebih umum:

- Batasan izin – Menetapkan izin maksimum yang dapat diberikan oleh kebijakan berbasis identitas kepada entitas IAM. Untuk informasi selengkapnya, lihat Batasan izin untuk entitas IAM dalam Panduan Pengguna IAM.
- Kebijakan kontrol layanan (SCP) – Menentukan izin maksimum untuk organisasi atau unit organisasi di AWS Organizations. Untuk informasi selengkapnya, lihat Kebijakan kontrol layanan dalam Panduan Pengguna AWS Organizations .

- Kebijakan kontrol sumber daya (RCP) – Menetapkan izin maksimum yang tersedia untuk sumber daya di akun Anda. Untuk informasi selengkapnya, lihat Kebijakan kontrol sumber daya (RCP) dalam Panduan Pengguna AWS Organizations .
- Kebijakan sesi – Kebijakan lanjutan yang diteruskan sebagai parameter saat membuat sesi sementara untuk peran atau pengguna terfederasi. Untuk informasi selengkapnya, lihat Kebijakan sesi dalam Panduan Pengguna IAM.

Berbagai jenis kebijakan

Ketika beberapa jenis kebijakan berlaku pada suatu permintaan, izin yang dihasilkan lebih rumit untuk dipahami. Untuk mempelajari cara AWS menentukan apakah akan mengizinkan permintaan saat beberapa jenis kebijakan terlibat, lihat Logika evaluasi kebijakan di Panduan Pengguna IAM.

Bagaimana AWS Transform bekerja dengan IAM

Sebelum Anda menggunakan IAM untuk mengelola akses ke AWS Transform, pelajari fitur IAM apa yang tersedia untuk digunakan dengan AWS Transform.

Fitur IAM	AWS Transformasi dukungan
Identity-based kebijakan	Ya
Resource-based kebijakan	Tidak
Tindakan kebijakan	Ya
Sumber daya kebijakan	Ya
Kunci kondisi kebijakan	Ya
ACL	Tidak
ABAC (tanda dalam kebijakan)	Parsial
Kredensial sementara	Ya
Izin principal	Ya

Fitur IAM	AWS Transformasi dukungan
Peran layanan	Ya
Service-linked peran	Ya

Untuk mendapatkan tampilan tingkat tinggi tentang cara AWS Transform dan AWS layanan lainnya bekerja dengan sebagian besar fitur IAM, lihat AWS layanan yang bekerja dengan IAM di Panduan Pengguna IAM.

Identity-based kebijakan untuk AWS Transformasi

Mendukung kebijakan berbasis identitas: Ya

Identity-based kebijakan adalah dokumen kebijakan izin JSON yang dapat Anda lampirkan ke identitas, seperti pengguna IAM, grup pengguna, atau peran. Kebijakan ini mengontrol jenis tindakan yang dapat dilakukan oleh pengguna dan peran, di sumber daya mana, dan berdasarkan kondisi seperti apa. Untuk mempelajari cara membuat kebijakan berbasis identitas, lihat Tentukan izin IAM kustom dengan kebijakan terkelola pelanggan dalam Panduan Pengguna IAM.

Dengan kebijakan berbasis identitas IAM, Anda dapat menentukan secara spesifik apakah tindakan dan sumber daya diizinkan atau ditolak, serta kondisi yang menjadi dasar dikabulkan atau ditolaknya tindakan tersebut. Untuk mempelajari semua elemen yang dapat Anda gunakan dalam kebijakan JSON, lihat Referensi elemen kebijakan JSON IAM dalam Panduan Pengguna IAM.

Identity-based contoh kebijakan untuk AWS Transformasi

Untuk melihat contoh kebijakan berbasis identitas AWS Transform, lihat. Identity-based contoh kebijakan untuk AWS Transformasi

Resource-based kebijakan dalam AWS Transformasi

Mendukung kebijakan berbasis sumber daya: Tidak

Resource-based kebijakan adalah dokumen kebijakan JSON yang Anda lampirkan ke sumber daya. Contoh kebijakan berbasis sumber daya adalah kebijakan kepercayaan peran IAM dan kebijakan bucket Amazon S3. Dalam layanan yang mendukung kebijakan berbasis sumber daya, administrator layanan dapat menggunakannya untuk mengontrol akses ke sumber daya tertentu. Untuk sumber daya tempat kebijakan dilampirkan, kebijakan menentukan tindakan apa yang dapat dilakukan

oleh principal tertentu pada sumber daya tersebut dan dalam kondisi apa. Anda harus menentukan principal dalam kebijakan berbasis sumber daya. Prinsipal dapat mencakup akun, pengguna, peran, pengguna federasi, atau. Layanan AWS

Untuk mengaktifkan akses lintas akun, Anda dapat menentukan secara spesifik seluruh akun atau entitas IAM di akun lain sebagai principal dalam kebijakan berbasis sumber daya. Untuk informasi selengkapnya, lihat Akses sumber daya lintas akun di IAM dalam Panduan Pengguna IAM.

Tindakan kebijakan untuk AWS Transformasi

Mendukung tindakan kebijakan: Ya

Administrator dapat menggunakan kebijakan AWS JSON untuk menentukan siapa yang memiliki akses ke apa. Yaitu, di mana utama dapat melakukan tindakan pada sumber daya, dan dalam kondisi apa.

Elemen `Action` dari kebijakan JSON menjelaskan tindakan yang dapat Anda gunakan untuk mengizinkan atau menolak akses dalam sebuah kebijakan. Sertakan tindakan dalam kebijakan untuk memberikan izin untuk melakukan operasi terkait.

Untuk melihat daftar tindakan AWS Transform, lihat Tindakan, sumber daya, dan kunci kondisi untuk AWS Transform dalam Referensi Otorisasi Layanan.

Untuk melihat daftar tindakan untuk AWS mengubah kustom, lihat Tindakan, sumber daya, dan kunci kondisi untuk AWS mengubah kustom di Referensi Otorisasi Layanan.

Tindakan kebijakan di AWS Transform menggunakan awalan berikut sebelum tindakan:

```
transform
```

Untuk menetapkan secara spesifik beberapa tindakan dalam satu pernyataan, pisahkan tindakan tersebut dengan koma.

```
"Action": [
      "transform:action1",
      "transform:action2"
         ]
```

Untuk melihat contoh kebijakan berbasis identitas AWS Transform, lihat. Identity-based contoh kebijakan untuk AWS Transformasi

Sumber daya kebijakan untuk AWS Transformasi

Mendukung sumber daya kebijakan: Ya

Administrator dapat menggunakan kebijakan AWS JSON untuk menentukan siapa yang memiliki akses ke apa. Yaitu, di mana utama dapat melakukan tindakan pada sumber daya, dan dalam kondisi apa.

Elemen kebijakan JSON `Resource` menentukan objek yang menjadi target penerapan tindakan. Praktik terbaiknya, tentukan sumber daya menggunakan Amazon Resource Name (ARN). Untuk tindakan yang tidak mendukung izin di tingkat sumber daya, gunakan wildcard (*) untuk menunjukkan bahwa pernyataan tersebut berlaku untuk semua sumber daya.

```
"Resource": "*"
```

Untuk melihat daftar jenis sumber daya AWS Transform dan ARNnya, dan dengan tindakan mana Anda dapat menentukan ARN dari setiap sumber daya, lihat Tindakan, sumber daya, dan kunci kondisi AWS untuk Transform dalam Referensi Otorisasi Layanan.

Untuk melihat daftar jenis sumber daya dan ARNnya, dan dengan tindakan mana Anda dapat menentukan ARN dari setiap sumber daya AWS untuk mengubah kustom, lihat Tindakan, sumber daya, dan kunci kondisi AWS untuk mengubah kustom dalam Referensi Otorisasi Layanan.

Untuk melihat contoh kebijakan berbasis identitas AWS Transform, lihat. Identity-based contoh kebijakan untuk AWS Transformasi

Kunci kondisi kebijakan untuk AWS Transformasi

Mendukung kunci kondisi kebijakan khusus layanan: Yes

Administrator dapat menggunakan kebijakan AWS JSON untuk menentukan siapa yang memiliki akses ke apa. Yaitu, principal dapat melakukan tindakan pada suatu sumber daya, dan dalam suatu syarat.

Elemen `Condition` menentukan ketika pernyataan dieksekusi berdasarkan kriteria yang ditetapkan. Anda dapat membuat ekspresi bersyarat yang menggunakan operator kondisi, misalnya sama dengan atau kurang dari, untuk mencocokkan kondisi dalam kebijakan dengan nilai-nilai yang

diminta. Untuk melihat semua kunci kondisi AWS global, lihat kunci konteks kondisi AWS global di Panduan Pengguna IAM.

Untuk melihat daftar kunci kondisi AWS Transform, dan dengan tindakan dan sumber daya yang dapat Anda gunakan kunci kondisi, lihat Tindakan, sumber daya, dan kunci kondisi untuk AWS Transform dalam Referensi Otorisasi Layanan.

Untuk melihat daftar kunci kondisi, serta tindakan dan sumber daya yang dapat Anda gunakan kunci kondisi untuk AWS mengubah kustom, lihat Tindakan, sumber daya, dan kunci kondisi untuk AWS mengubah kustom di Referensi Otorisasi Layanan.

Untuk melihat contoh kebijakan berbasis identitas AWS Transform, lihat. Identity-based contoh kebijakan untuk AWS Transformasi

ACL di AWS Transformasi

Mendukung ACL: Tidak

Daftar kontrol akses (ACL) mengendalikan principal mana (anggota akun, pengguna, atau peran) yang memiliki izin untuk mengakses sumber daya. ACL serupa dengan kebijakan berbasis sumber daya, meskipun kebijakan tersebut tidak menggunakan format dokumen kebijakan JSON.

ABAC dengan AWS Transformasi

Mendukung ABAC (tag dalam kebijakan): Sebagian

Attribute-based Access Control (ABAC) adalah strategi otorisasi yang mendefinisikan izin berdasarkan atribut yang disebut tag. Anda dapat melampirkan tag ke entitas dan AWS sumber daya IAM, lalu merancang kebijakan ABAC untuk mengizinkan operasi saat tag prinsipal cocok dengan tag pada sumber daya.

Untuk mengendalikan akses berdasarkan tanda, berikan informasi tentang tanda di elemen kondisi dari kebijakan menggunakan kunci kondisi `aws:ResourceTag/key-name`, `aws:RequestTag/key-name`, atau `aws:TagKeys`.

Jika sebuah layanan mendukung ketiga kunci kondisi untuk setiap jenis sumber daya, nilainya adalah Ya untuk layanan tersebut. Jika suatu layanan mendukung ketiga kunci kondisi untuk hanya beberapa jenis sumber daya, nilainya adalah Parsial.

Untuk informasi selengkapnya tentang ABAC, lihat Tentukan izin dengan otorisasi ABAC dalam Panduan Pengguna IAM. Untuk melihat tutorial yang menguraikan langkah-langkah pengaturan ABAC, lihat Menggunakan kontrol akses berbasis atribut (ABAC) dalam Panduan Pengguna IAM.

Menggunakan kredensyal sementara dengan AWS Transformasi

Mendukung kredensial sementara: Ya

Kredensi sementara menyediakan akses jangka pendek ke AWS sumber daya dan secara otomatis dibuat saat Anda menggunakan federasi atau beralih peran. AWS merekomendasikan agar Anda secara dinamis menghasilkan kredensi sementara alih-alih menggunakan kunci akses jangka panjang. Untuk informasi selengkapnya, lihat Kredensial keamanan sementara di IAM dan Layanan AWS yang berfungsi dengan IAM dalam Panduan Pengguna IAM.

Cross-service izin utama untuk AWS Transformasi

Mendukung sesi akses terusan (FAS): Ya

Sesi akses teruskan (FAS) menggunakan izin pemanggilan utama Layanan AWS, dikombinasikan dengan permintaan Layanan AWS untuk membuat permintaan ke layanan hilir. Untuk detail kebijakan ketika mengajukan permintaan FAS, lihat Sesi akses terusan.

Peran layanan untuk AWS Transformasi

Mendukung peran layanan: Ya

Peran layanan adalah peran IAM yang diambil oleh sebuah layanan untuk melakukan tindakan atas nama Anda. Administrator IAM dapat membuat, mengubah, dan menghapus peran layanan dari dalam IAM. Untuk informasi selengkapnya, lihat Buat sebuah peran untuk mendelegasikan izin ke Layanan AWS dalam Panduan pengguna IAM.

> **Warning**
>
> Mengubah izin untuk peran layanan dapat merusak fungsionalitas AWS Transform. Edit peran layanan hanya jika AWS Transform memberikan panduan untuk melakukannya.

Service-linked peran untuk AWS Transformasi

Mendukung peran terkait layanan: Ya

Peran terkait layanan adalah jenis peran layanan yang ditautkan ke. Layanan AWS Layanan dapat mengambil peran untuk melakukan tindakan atas nama Anda. Service-linked peran muncul di Anda Akun AWS dan dimiliki oleh layanan. Administrator IAM dapat melihat, tetapi tidak dapat mengedit izin untuk peran terkait layanan.

Untuk detail tentang membuat atau mengelola peran terkait layanan AWS Transform, lihat. Menggunakan peran terkait layanan untuk AWS Transform

Identity-based contoh kebijakan untuk AWS Transformasi

Secara default, pengguna dan peran tidak memiliki izin untuk membuat atau memodifikasi sumber daya AWS Transform. Untuk memberikan izin kepada pengguna untuk melakukan tindakan di sumber daya yang mereka perlukan, administrator IAM dapat membuat kebijakan IAM.

Untuk mempelajari cara membuat kebijakan berbasis identitas IAM menggunakan contoh dokumen kebijakan JSON ini, lihat Membuat kebijakan IAM (konsol) di Panduan Pengguna IAM.

Untuk detail tentang tindakan dan jenis sumber daya yang ditentukan oleh AWS Transform, termasuk format ARN untuk setiap jenis sumber daya, lihat Tindakan, Sumber Daya, dan Kunci Kondisi untuk AWS Transformasi dalam Referensi Otorisasi Layanan.

Topik

- Praktik terbaik kebijakan
- Menggunakan AWS Mengubah konsol
- Mengizinkan pengguna melihat izin mereka sendiri
- Izinkan administrator untuk menerima permintaan konektor dari akun dengan AWS Transformasi
- Izinkan administrator untuk menetapkan pengguna Pusat Identitas IAM yang ada dan membuat pengguna Pusat Identitas IAM baru untuk ditetapkan AWS Transformasi
- Izinkan administrator untuk mengaktifkan AWS Transformasi
- Memungkinkan pengguna untuk mengakses AWS Transformasi dengan kredensyal IAM

Praktik terbaik kebijakan

Identity-based kebijakan menentukan apakah seseorang dapat membuat, mengakses, atau menghapus sumber daya AWS Transform di akun Anda. Tindakan ini membuat Akun AWS Anda dikenai biaya. Ketika Anda membuat atau mengedit kebijakan berbasis identitas, ikuti panduan dan rekomendasi ini:

- Mulailah dengan kebijakan AWS terkelola dan beralih ke izin hak istimewa paling sedikit — Untuk mulai memberikan izin kepada pengguna dan beban kerja Anda, gunakan kebijakan AWS terkelola yang memberikan izin untuk banyak kasus penggunaan umum. Mereka tersedia di Anda Akun AWS. Kami menyarankan Anda mengurangi izin lebih lanjut dengan menentukan kebijakan

yang dikelola AWS pelanggan yang khusus untuk kasus penggunaan Anda. Untuk informasi selengkapnya, lihat Kebijakan yang dikelola AWS atau Kebijakan yang dikelola AWS untuk fungsi tugas dalam Panduan Pengguna IAM.

- Menerapkan izin dengan hak akses paling rendah – Ketika Anda menetapkan izin dengan kebijakan IAM, hanya berikan izin yang diperlukan untuk melakukan tugas. Anda melakukannya dengan mendefinisikan tindakan yang dapat diambil pada sumber daya tertentu dalam kondisi tertentu, yang juga dikenal sebagai izin dengan hak akses paling rendah. Untuk informasi selengkapnya tentang cara menggunakan IAM untuk mengajukan izin, lihat Kebijakan dan izin dalam IAM dalam Panduan Pengguna IAM.
- Gunakan kondisi dalam kebijakan IAM untuk membatasi akses lebih lanjut – Anda dapat menambahkan suatu kondisi ke kebijakan Anda untuk membatasi akses ke tindakan dan sumber daya. Sebagai contoh, Anda dapat menulis kondisi kebijakan untuk menentukan bahwa semua permintaan harus dikirim menggunakan SSL. Anda juga dapat menggunakan ketentuan untuk memberikan akses ke tindakan layanan jika digunakan melalui yang spesifik Layanan AWS, seperti CloudFormation. Untuk informasi selengkapnya, lihat Elemen kebijakan JSON IAM: Kondisi dalam Panduan Pengguna IAM.
- Gunakan IAM Access Analyzer untuk memvalidasi kebijakan IAM Anda untuk memastikan izin yang aman dan fungsional – IAM Access Analyzer memvalidasi kebijakan baru dan yang sudah ada sehingga kebijakan tersebut mematuhi bahasa kebijakan IAM (JSON) dan praktik terbaik IAM. IAM Access Analyzer menyediakan lebih dari 100 pemeriksaan kebijakan dan rekomendasi yang dapat ditindaklanjuti untuk membantu Anda membuat kebijakan yang aman dan fungsional. Untuk informasi selengkapnya, lihat Validasi kebijakan dengan IAM Access Analyzer dalam Panduan Pengguna IAM.
- Memerlukan otentikasi multi-faktor (MFA) - Jika Anda memiliki skenario yang mengharuskan pengguna IAM atau pengguna root di Anda, Akun AWS aktifkan MFA untuk keamanan tambahan. Untuk meminta MFA ketika operasi API dipanggil, tambahkan kondisi MFA pada kebijakan Anda. Untuk informasi selengkapnya, lihat Amankan akses API dengan MFA dalam Panduan Pengguna IAM.

Untuk informasi selengkapnya tentang praktik terbaik dalam IAM, lihat Praktik terbaik keamanan di IAM dalam Panduan Pengguna IAM.

Menggunakan AWS Mengubah konsol

Untuk mengakses konsol AWS Transform, Anda harus memiliki set izin minimum. Izin ini harus memungkinkan Anda untuk membuat daftar dan melihat detail tentang sumber daya AWS Transform

di Anda Akun AWS. Jika Anda membuat kebijakan berbasis identitas yang lebih ketat daripada izin minimum yang diperlukan, konsol tidak akan berfungsi sebagaimana mestinya untuk entitas (pengguna atau peran) dengan kebijakan tersebut.

Anda tidak perlu mengizinkan izin konsol minimum untuk pengguna yang melakukan panggilan hanya ke AWS CLI atau AWS API. Sebagai gantinya, izinkan akses hanya ke tindakan yang sesuai dengan operasi API yang coba mereka lakukan.

Mengizinkan pengguna melihat izin mereka sendiri

Contoh ini menunjukkan cara membuat kebijakan yang mengizinkan pengguna IAM melihat kebijakan inline dan terkelola yang dilampirkan ke identitas pengguna mereka. Kebijakan ini mencakup izin untuk menyelesaikan tindakan ini di konsol atau menggunakan API atau secara terprogram. AWS CLI AWS

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ViewOwnUserInfo",
            "Effect": "Allow",
            "Action": [
                "iam:GetUserPolicy",
                "iam:ListGroupsForUser",
                "iam:ListAttachedUserPolicies",
                "iam:ListUserPolicies",
                "iam:GetUser"
            ],
            "Resource": ["arn:aws:iam::*:user/${aws:username}"]
        },
        {
            "Sid": "NavigateInConsole",
            "Effect": "Allow",
            "Action": [
                "iam:GetGroupPolicy",
                "iam:GetPolicyVersion",
                "iam:GetPolicy",
                "iam:ListAttachedGroupPolicies",
                "iam:ListGroupPolicies",
                "iam:ListPolicyVersions",
                "iam:ListPolicies",
                "iam:ListUsers"
```

```
            ],
            "Resource": "*"
        }
    ]
}
```

Izinkan administrator untuk menerima permintaan konektor dari akun dengan AWS Transformasi

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "transform:GetConnector",
                "transform:AssociateConnectorResource",
                "transform:RejectConnector"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetBucketPublicAccessBlock",
                "s3:GetAccountPublicAccessBlock"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "iam:CreateRole",
                "iam:AttachRolePolicy",
                "iam:PassRole"
            ],
            "Resource": "arn:aws:iam::111122223333:role/service-role/
AWSTransform-*"
        },
        {
```

```
            "Effect": "Allow",
            "Action": [
                "iam:CreatePolicy"
            ],
            "Resource": "arn:aws:iam::111122223333:policy/service-role/
AWSTransform-*"
        }
    ]
}
```

Izinkan administrator untuk menetapkan pengguna Pusat Identitas IAM yang ada dan membuat pengguna Pusat Identitas IAM baru untuk ditetapkan AWS Transformasi

Kebijakan berikut memberikan izin yang diperlukan administrator untuk mengelola pengguna dan grup Pusat Identitas IAM di konsol Transform. AWS Dengan kebijakan ini, administrator dapat menetapkan dan menghapus pengguna dan grup, melihat nama tampilan, membuat pengguna Pusat Identitas IAM baru, mengelola penetapan aplikasi, dan mengonfigurasi setelan sesi aplikasi.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "SSOApplicationAssignments",
      "Effect": "Allow",
      "Action": [
        "sso:ListApplicationAssignments",
        "sso:CreateApplicationAssignment",
        "sso:DeleteApplicationAssignment",
        "sso:ListInstances",
        "sso:DescribeInstance",
        "sso:ListDirectoryAssociations",
        "sso:GetApplicationGrant",
        "sso:GetApplicationAccessScope",
        "sso:GetApplicationSessionConfiguration",
        "sso:PutApplicationSessionConfiguration"
      ],
      "Resource": "*"
    },
    {
      "Sid": "SSODirectoryCreateUser",
      "Effect": "Allow",
```

```
      "Action": [
        "sso-directory:CreateUser"
      ],
      "Resource": "*"
    },
    {
      "Sid": "IdentityStoreAccess",
      "Effect": "Allow",
      "Action": [
        "identitystore:DescribeUser",
        "identitystore:DescribeGroup",
        "identitystore:ListUsers",
        "identitystore:ListGroups"
      ],
      "Resource": "*"
    },
    {
      "Sid": "AllowKmsAccessViaIdentityCenter",
      "Effect": "Allow",
      "Action": [
        "kms:Decrypt"
      ],
      "Resource": "*",
      "Condition": {
        "ArnLike": {
          "kms:EncryptionContext:aws:sso:instance-arn": "arn:*:sso:::instance/*"
        },
        "StringLike": {
          "kms:ViaService": "sso.*.amazonaws.com"
        }
      }
    },
    {
      "Sid": "AllowKmsAccessViaIdentityStore",
      "Effect": "Allow",
      "Action": [
        "kms:Decrypt"
      ],
      "Resource": "*",
      "Condition": {
        "ArnLike": {
          "kms:EncryptionContext:aws:identitystore:identitystore-arn":
"arn:*:identitystore::*:identitystore/*"
        },
```

```
        "StringLike": {
          "kms:ViaService": "identitystore.*.amazonaws.com"
        }
      }
    }
  ]
}
```

Izinkan administrator untuk mengaktifkan AWS Transformasi

Kebijakan berikut memberikan izin yang diperlukan administrator untuk mengaktifkan dan mengelola AWS Transform melalui konsol. AWS Ini termasuk mengelola profil, mengonfigurasi agen, dan mengelola konektor. Kebijakan ini ditujukan hanya untuk administrator dan harus dicakup oleh prinsipal tepercaya di akun Anda.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "SSOEnableTransform",
      "Effect": "Allow",
      "Action": [
        "sso:ListInstances",
        "sso:CreateInstance",
        "sso:CreateApplication",
        "sso:PutApplicationAuthenticationMethod",
        "sso:PutApplicationGrant",
        "sso:PutApplicationAssignmentConfiguration",
        "sso:ListApplications",
        "sso:GetSharedSsoConfiguration",
        "sso:DescribeInstance",
        "sso:PutApplicationAccessScope",
        "sso:DescribeApplication",
        "sso:DeleteApplication",
        "sso:UpdateApplication",
        "sso:DescribeRegisteredRegions",
        "sso:GetSSOStatus"
      ],
      "Resource": "*"
    },
    {
      "Sid": "SSODirectoryActions",
      "Effect": "Allow",
```

```
      "Action": [
        "sso-directory:GetUserPoolInfo",
        "sso-directory:DescribeUsers",
        "sso-directory:DescribeGroups",
        "sso-directory:SearchGroups",
        "sso-directory:SearchUsers",
        "sso-directory:DescribeDirectory"
      ],
      "Resource": "*"
    },
    {
      "Sid": "KMSListAliases",
      "Effect": "Allow",
      "Action": [
        "kms:ListAliases"
      ],
      "Resource": "*"
    },
    {
      "Sid": "KMSActions",
      "Effect": "Allow",
      "Action": [
        "kms:CreateGrant",
        "kms:Encrypt",
        "kms:Decrypt",
        "kms:GenerateDataKey*",
        "kms:RetireGrant",
        "kms:DescribeKey"
      ],
      "Resource": "*",
      "Condition": {
        "StringLike": {
          "kms:ViaService": "transform.*.amazonaws.com"
        }
      }
    },
    {
      "Sid": "IAMServiceLinkedRole",
      "Effect": "Allow",
      "Action": [
        "iam:CreateServiceLinkedRole"
      ],
      "Resource": [
```

```
        "arn:*:iam::*:role/aws-service-role/transform.amazonaws.com/
AWSServiceRoleForAWSTransform"
      ]
    },
    {
      "Sid": "TransformConsoleActions",
      "Effect": "Allow",
      "Action": [
        "transform:GetAccountSettings",
        "transform:UpdateAccountSettings",
        "transform:CreateProfile",
        "transform:UpdateProfile",
        "transform:DeleteProfile",
        "transform:ListProfiles",
        "transform:GetConnector",
        "transform:ListConnectors",
        "transform:DeleteConnector",
        "transform:AssociateConnectorResource",
        "transform:RejectConnector",
        "transform:ListAgents",
        "transform:GetAgent",
        "transform:GetAgentRuntimeConfiguration",
        "transform:PutAgentRuntimeConfiguration",
        "transform:UpdateAgentAccess",
        "transform:TagResource",
        "transform:UntagResource",
        "transform:ListTagsForResource",
        "transform:GetWebAppUrl"
      ],
      "Resource": "*"
    }
  ]
}
```

Memungkinkan pengguna untuk mengakses AWS Transformasi dengan kredensyal IAM

Kebijakan berikut memberikan akses utama IAM ke aplikasi web AWS Transform dan API menggunakan kredensi IAM. Kebijakan ini diperlukan untuk IAM-only akses dan akses IAM diaktifkan bersama penyedia identitas.

Ganti *`regionaccount-id`*,, dan *`profile-id`* dengan nilai-nilai Anda. Anda dapat menemukan ID profil di konsol AWS Transform di bawah Pengaturan.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowTransformAccess",
      "Effect": "Allow",
      "Action": "transform:AccessTransformProfile",
      "Resource": "arn:aws:transform:region:account-id:profile/profile-id"
    }
  ]
}
```

Untuk memberikan akses ke semua profil di akun, gunakan wildcard untuk ID profil:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowTransformAccessAllProfiles",
      "Effect": "Allow",
      "Action": "transform:AccessTransformProfile",
      "Resource": "arn:aws:transform:region:account-id:profile/*"
    }
  ]
}
```

> **Note**
>
> `transform:AccessTransformProfile`Tindakan memberikan akses ke AWS Transform saja. Tindakan dalam ruang kerja dikendalikan oleh peran AWS Transform workspace (Admin, Contributor, Approver, Read-only), bukan oleh kebijakan IAM. Peran ruang kerja menentukan apa yang dapat dilakukan pengguna, seperti membuat pekerjaan, mengelola kolaborator, atau menyetujui tugas.

Pemecahan masalah AWS Mengubah identitas dan akses

Gunakan informasi berikut untuk membantu Anda mendiagnosis dan memperbaiki masalah umum yang mungkin Anda temui saat bekerja dengan AWS Transform dan IAM.

Topik

Saya tidak berwenang untuk melakukan tindakan di AWS Transformasi

Jika Anda menerima pesan kesalahan bahwa Anda tidak memiliki otorisasi untuk melakukan tindakan, kebijakan Anda harus diperbarui agar Anda dapat melakukan tindakan tersebut.

Contoh kesalahan berikut terjadi ketika pengguna IAM `mateojackson` mencoba menggunakan konsol untuk melihat detail tentang suatu sumber daya *`my-example-widget`* rekaan, tetapi tidak memiliki izin `AWS Transform:`*`GetWidget`* rekaan.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform: AWS
 Transform:GetWidget on resource: my-example-widget
```

Dalam hal ini, kebijakan untuk pengguna `mateojackson` harus diperbarui untuk mengizinkan akses ke sumber daya *`my-example-widget`* dengan menggunakan tindakan `AWS Transform:`*`GetWidget`*.

Jika Anda memerlukan bantuan, hubungi AWS administrator Anda. Administrator Anda adalah orang yang memberi Anda kredensial masuk.

Saya tidak berwenang untuk melakukan iam: PassRole

Jika Anda menerima kesalahan yang tidak diizinkan untuk melakukan `iam:PassRole` tindakan, kebijakan Anda harus diperbarui agar Anda dapat meneruskan peran ke AWS Transform.

Beberapa Layanan AWS memungkinkan Anda untuk meneruskan peran yang ada ke layanan tersebut alih-alih membuat peran layanan baru atau peran terkait layanan. Untuk melakukannya, Anda harus memiliki izin untuk meneruskan peran ke layanan.

Contoh kesalahan berikut terjadi ketika pengguna IAM bernama `marymajor` mencoba menggunakan konsol untuk melakukan tindakan di AWS Transform. Namun, tindakan tersebut memerlukan layanan untuk mendapatkan izin yang diberikan oleh peran layanan. Mary tidak memiliki izin untuk meneruskan peran tersebut pada layanan.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
 iam:PassRole
```

Dalam kasus ini, kebijakan Mary harus diperbarui agar dia mendapatkan izin untuk melakukan tindakan `iam:PassRole` tersebut.

Jika Anda memerlukan bantuan, hubungi AWS administrator Anda. Administrator Anda adalah orang yang memberi Anda kredensial masuk.

Saya ingin mengizinkan orang di luar saya Akun AWS untuk mengakses saya AWS Mengubah sumber daya

Anda dapat membuat peran yang dapat digunakan pengguna di akun lain atau orang-orang di luar organisasi Anda untuk mengakses sumber daya Anda. Anda dapat menentukan siapa saja yang dipercaya untuk mengambil peran tersebut. Untuk layanan yang mendukung kebijakan berbasis sumber daya atau daftar kontrol akses (ACL), Anda dapat menggunakan kebijakan tersebut untuk memberi orang akses ke sumber daya Anda.

Untuk mempelajari selengkapnya, periksa referensi berikut:

- Untuk mempelajari apakah AWS Transform mendukung fitur ini, lihatBagaimana AWS Transform bekerja dengan IAM.
- Untuk mempelajari cara menyediakan akses ke sumber daya Anda di seluruh sumber daya Akun AWS yang Anda miliki, lihat Menyediakan akses ke pengguna IAM di pengguna lain Akun AWS yang Anda miliki di Panduan Pengguna IAM.
- Untuk mempelajari cara menyediakan akses ke sumber daya Anda kepada pihak ketiga Akun AWS, lihat Menyediakan akses yang Akun AWS dimiliki oleh pihak ketiga dalam Panduan Pengguna IAM.
- Untuk mempelajari cara memberikan akses melalui federasi identitas, lihat Menyediakan akses ke pengguna terautentikasi eksternal (federasi identitas) dalam Panduan Pengguna IAM.
- Untuk mempelajari perbedaan antara menggunakan peran dan kebijakan berbasis sumber daya untuk akses lintas akun, lihat Akses sumber daya lintas akun di IAM di Panduan Pengguna IAM.

Menggunakan peran terkait layanan untuk AWS Transform

AWS Transform menggunakan AWS Identity and Access Management peran terkait layanan (IAM). Peran terkait layanan adalah jenis unik peran IAM yang ditautkan langsung ke. AWS Transform Service-linked peran telah ditentukan sebelumnya oleh AWS Transform dan mencakup semua izin yang diperlukan layanan untuk memanggil AWS layanan lain atas nama Anda.

Menggunakan peran terkait layanan untuk AWS Transform

AWS Transform menggunakan AWS Identity and Access Management peran terkait layanan (IAM). Peran terkait layanan adalah jenis unik peran IAM yang ditautkan langsung ke. AWS Transform Service-linked peran telah ditentukan sebelumnya oleh AWS Transform dan mencakup semua izin yang diperlukan layanan untuk memanggil AWS layanan lain atas nama Anda.

Peran terkait layanan membuat pengaturan AWS Transform lebih mudah karena Anda tidak perlu menambahkan izin yang diperlukan secara manual. AWS Transform mendefinisikan izin peran terkait layanan, dan kecuali ditentukan lain, hanya AWS Transform dapat mengambil perannya. Izin yang ditentukan mencakup kebijakan kepercayaan dan kebijakan izin, dan kebijakan izin tersebut tidak dapat dilampirkan ke entitas IAM lainnya.

Anda dapat menghapus peran tertaut layanan hanya setelah menghapus sumber daya terkait terlebih dahulu. Ini melindungi AWS Transform sumber daya Anda karena Anda tidak dapat secara tidak sengaja menghapus izin untuk mengakses sumber daya.

Untuk informasi tentang layanan lain yang mendukung peran terkait layanan, lihat AWS layanan yang bekerja dengan IAM dan cari layanan yang memiliki Ya di kolom peran. Service-linked Pilih Ya dengan sebuah tautan untuk melihat dokumentasi peran terkait layanan untuk layanan tersebut.

Service-linked izin peran untuk AWS Transform

AWS Transform menggunakan peran terkait layanan bernama AWSServiceRoleForAWSTransform—Peran ini Service-Linked memberi AWS Transform kemampuan untuk memberikan informasi penggunaan.

Peran AWSServiceRoleForAWSTransform terkait layanan mempercayai layanan berikut untuk mengambil peran:

- `transform.amazonaws.com`

Kebijakan izin peran bernama AWSServiceRoleForAWSTransform memungkinkan AWS Transform untuk menyelesaikan tindakan berikut pada sumber daya yang ditentukan:

- jam tangan awan: PutMetricData
  - Kirim metrik khusus CloudWatch untuk operasi AWS Transform
  - Lacak kemajuan transformasi, tingkat keberhasilan, dan metrik kinerja
  - Aktifkan pemantauan dan peringatan pada alur kerja transformasi
- sso: DescribeApplication
  - Lihat detail tentang aplikasi tertentu di Pusat Identitas
  - Dapatkan metadata aplikasi seperti nama, deskripsi, status, dan konfigurasi
- sso: GetApplicationAssignmentConfiguration
  - Mengambil pengaturan konfigurasi tugas untuk aplikasi
  - Lihat bagaimana users/groups dikonfigurasi untuk ditugaskan ke aplikasi
- sso: ListApplicationAssignmentsForPrincipal
  - Daftar semua aplikasi yang ditugaskan untuk pengguna atau grup tertentu (prinsipal)
  - Lihat aplikasi mana yang dapat diakses oleh identitas tertentu
- Mengaktifkan dekripsi KMS-encrypted data saat diakses melalui IAM Identity Center. Hanya berfungsi ketika konteks enkripsi berisi ARN instance IAM Identity Center yang valid dan harus diakses melalui titik akhir layanan IAM Identity Center.
- Memungkinkan dekripsi KMS-encrypted data saat diakses melalui Identity Store. Hanya berfungsi jika konteks enkripsi berisi ARN Toko Identitas yang valid dan harus diakses melalui titik akhir layanan Identity Store.
- manajer rahasia: GetSecretValue
  - Akses rahasia AWS Transform terkait layanan yang digunakan untuk menyimpan rahasia klien untuk penyedia identitas eksternal
  - Sumber daya: arn:aws:secretsmanager: *:*:secret:transform-preprod! *
  - Kondisi: Harus dimiliki oleh layanan transform-preprod dan diakses dari akun yang sama
- dukungan:CreateCase, dukungan:DescribeCases, dukungan:DescribeCommunications, dukungan:AddCommunicationToCase, dukungan: ResolveCase
  - Membuat dan mengelola kasus dukungan premium dari aplikasi AWS Transform web
  - Lihat detail kasus dan komunikasi
  - Tambahkan komunikasi dan selesaikan kasus dukungan

Anda harus mengonfigurasi izin agar pengguna, grup, atau peran Anda membuat, mengedit, atau menghapus peran terkait layanan. Untuk informasi selengkapnya, lihat izin Service-linked peran di Panduan Pengguna IAM.

Membuat peran terkait layanan untuk AWS Transform

Anda tidak perlu membuat peran terkait layanan secara manual. Saat Anda membuat profil untuk AWS Transform di Konsol Manajemen AWS, AWS Transform buat peran terkait layanan untuk Anda.

Jika Anda menghapus peran terkait layanan ini, dan ingin membuatnya lagi, Anda dapat mengulangi proses yang sama untuk membuat kembali peran tersebut di akun Anda. Saat Anda memperbarui pengaturan, AWS Transform buat peran terkait layanan untuk Anda lagi.

Anda juga dapat menggunakan konsol IAM atau AWS CLI untuk membuat peran terkait layanan dengan nama layanan. `transform.amazonaws.com` Untuk informasi selengkapnya, lihat Membuat peran tertaut layanan dalam Panduan Pengguna IAM. Jika Anda menghapus peran tertaut layanan ini, Anda dapat mengulang proses yang sama untuk membuat peran tersebut lagi.

Menyunting peran terkait layanan untuk AWS Transform

AWS Transform tidak memungkinkan Anda untuk mengedit peran AWSServiceRoleForAWSTransform terkait layanan. Setelah membuat peran terkait layanan, Anda tidak dapat mengubah nama peran karena berbagai entitas mungkin merujuk peran tersebut. Namun, Anda dapat mengedit penjelasan peran menggunakan IAM. Untuk informasi selengkapnya, lihat Mengedit peran terkait layanan dalam Panduan Pengguna IAM.

Menghapus peran terkait layanan untuk AWS Transform

Jika Anda tidak perlu lagi menggunakan fitur atau layanan yang memerlukan peran terkait layanan, kami merekomendasikan Anda menghapus peran tersebut. Dengan begitu, Anda tidak memiliki entitas yang tidak digunakan yang tidak dipantau atau dipelihara secara aktif. Tetapi, Anda harus membersihkan sumber daya peran terkait layanan sebelum menghapusnya secara manual.

> **Note**
>
> Jika AWS Transform layanan menggunakan peran saat Anda mencoba menghapus sumber daya, maka penghapusan mungkin gagal. Jika hal itu terjadi, tunggu beberapa menit dan coba mengoperasikannya lagi.

Untuk menghapus peran tertaut layanan secara manual menggunakan IAM

Gunakan konsol IAM, the AWS CLI, atau AWS API untuk menghapus peran AWSServiceRoleForAWSTransform terkait layanan. Untuk informasi selengkapnya, lihat Menghapus peran terkait layanan dalam Panduan Pengguna IAM.

Wilayah yang Didukung untuk AWS Transform peran terkait layanan

AWS Transform tidak mendukung penggunaan peran terkait layanan di setiap Wilayah tempat layanan tersedia. Anda dapat menggunakan AWSServiceRoleForAWSTransform peran di Wilayah berikut. Untuk informasi selengkapnya, lihat AWS Wilayah dan titik akhir.

Nama wilayah	Identitas wilayah	Support di AWS Transform
US East (Northern Virginia)	us-east-1	Ya
Eropa (Frankfurt)	eu-central-1	Ya

Menggunakan peran terkait layanan untuk AWS Ubah Kustom

AWS Transform Custom menggunakan nama peran terkait layanan AWSServiceRoleForAWSTransformCustomuntuk mempublikasikan metrik dan log ke akun Anda atas nama Anda. Metrik ini memungkinkan Anda memantau jumlah transformasi, latensi, dan kode status secara langsung di dasbor Anda. Log memberikan visibilitas ke masalah yang dihadapi selama transformasi.

Peran terkait layanan ini telah ditentukan sebelumnya oleh AWS Transform Custom dan hanya menyertakan izin yang dibutuhkan layanan. Anda tidak perlu menambahkan izin apa pun secara manual, dan peran tersebut hanya dapat diasumsikan oleh AWS Transform Custom. Untuk informasi umum tentang peran terkait layanan, lihat Menggunakan peran terkait layanan di Panduan Pengguna IAM.

Service-linked izin peran untuk AWS Ubah Kustom

Peran AWSServiceRoleForAWSTransformCustom terkait layanan mempercayai layanan berikut untuk mengambil peran:

- `transform-custom.amazonaws.com`

Kebijakan izin peran bernama AWSServiceRoleForAWSTransformCustom memungkinkan AWS Transform Custom untuk menyelesaikan tindakan berikut pada sumber daya yang ditentukan:

- `cloudwatch:PutMetricData`pada semua AWS sumber daya
  - Publikasikan metrik operasional ke bawah namespace `AWS/TransformCustom`
  - Lacak jumlah transformasi, latensi, dan kode status
  - Cakupan ke `AWS/TransformCustom` namespace melalui tombol kondisi `cloudwatch:namespace`
- `logs:CreateLogGroup`dan `logs:PutRetentionPolicy` pada grup `/aws/TransformCustom` log
  - Buat grup CloudWatch log Log untuk menerbitkan log transformasi
  - Menetapkan kebijakan penyimpanan log pada grup log
  - Tercakup `arn:aws:logs:*:*:log-group:/aws/TransformCustom` dengan `aws:ResourceAccount` kondisi yang memastikan akses hanya dalam akun Anda
- `logs:CreateLogStream`dan `logs:PutLogEvents` pada aliran log dalam grup `/aws/TransformCustom` log
  - Buat aliran log dan publikasikan peristiwa log untuk operasi transformasi
  - Tercakup `arn:aws:logs:*:*:log-group:/aws/TransformCustom:log-stream:*` dengan `aws:ResourceAccount` kondisi yang memastikan akses hanya dalam akun Anda

Anda harus mengonfigurasi izin agar pengguna, grup, atau peran Anda membuat, mengedit, atau menghapus peran terkait layanan. Untuk informasi selengkapnya, lihat izin Service-linked peran di Panduan Pengguna IAM.

Membuat peran terkait layanan untuk AWS Ubah Kustom

Anda tidak perlu membuat peran terkait layanan secara manual. Saat Anda menjalankan transformasi menggunakan AWS Transform Custom CLI, AWS Transform Custom membuat peran terkait layanan untuk Anda.

Jika Anda menghapus peran terkait layanan ini, dan ingin membuatnya lagi, Anda dapat mengulangi proses yang sama untuk membuat kembali peran tersebut di akun Anda. Saat Anda menjalankan transformasi menggunakan AWS Transform Custom CLI, AWS Transform Custom membuat peran terkait layanan untuk Anda lagi.

Anda juga dapat menggunakan konsol IAM untuk membuat peran terkait layanan dengan kasus penggunaan. AWSServiceRoleForAWSTransformCustom Di AWS CLI atau AWS API, buat peran terkait layanan dengan nama `transform-custom.amazonaws.com` layanan. Untuk informasi selengkapnya, lihat Membuat peran tertaut layanan dalam Panduan Pengguna IAM. Jika Anda menghapus peran tertaut layanan ini, Anda dapat mengulang proses yang sama untuk membuat peran tersebut lagi.

Menyunting peran terkait layanan untuk AWS Ubah Kustom

AWS Transform Custom tidak memungkinkan Anda mengedit peran AWSServiceRoleForAWSTransformCustom terkait layanan. Setelah membuat peran terkait layanan, Anda tidak dapat mengubah nama peran karena berbagai entitas mungkin merujuk peran tersebut. Namun, Anda dapat mengedit penjelasan peran menggunakan IAM. Untuk informasi selengkapnya, lihat Mengedit peran terkait layanan dalam Panduan Pengguna IAM.

Menghapus peran terkait layanan untuk AWS Ubah Kustom

Jika Anda tidak perlu lagi menggunakan fitur atau layanan yang memerlukan peran terkait layanan, kami merekomendasikan Anda menghapus peran tersebut. Dengan begitu, Anda tidak memiliki entitas yang tidak digunakan yang tidak dipantau atau dipelihara secara aktif.

Peran AWSServiceRoleForAWSTransformCustom terkait layanan tidak membuat sumber daya persisten di akun Anda. Anda dapat menghapusnya secara langsung tanpa pembersihan sumber daya apa pun.

Jika Anda menghapus peran ini dan kemudian menjalankan transformasi menggunakan AWS Transform Custom CLI, peran yang ditautkan layanan akan dibuat ulang secara otomatis.

Untuk menghapus peran tertaut layanan secara manual menggunakan IAM

Gunakan konsol IAM, the AWS CLI, atau AWS API untuk menghapus peran AWSServiceRoleForAWSTransformCustom terkait layanan. Untuk informasi selengkapnya, lihat Menghapus peran terkait layanan dalam Panduan Pengguna IAM.

Wilayah yang Didukung untuk AWS Mengubah peran terkait layanan kustom

AWS Transform Custom mendukung penggunaan peran terkait layanan di semua Wilayah tempat AWS Transform Custom tersedia. Untuk daftar Wilayah yang didukung, lihat Wilayah yang Didukung.

AWS kebijakan terkelola untuk AWS Transformasi

Kebijakan AWS terkelola adalah kebijakan mandiri yang dibuat dan dikelola oleh AWS. AWS Kebijakan terkelola dirancang untuk memberikan izin bagi banyak kasus penggunaan umum sehingga Anda dapat mulai menetapkan izin kepada pengguna, grup, dan peran.

Perlu diingat bahwa kebijakan AWS terkelola mungkin tidak memberikan izin hak istimewa paling sedikit untuk kasus penggunaan spesifik Anda karena tersedia untuk digunakan semua pelanggan. AWS Kami menyarankan Anda untuk mengurangi izin lebih lanjut dengan menentukan kebijakan yang dikelola pelanggan yang khusus untuk kasus penggunaan Anda.

Anda tidak dapat mengubah izin yang ditentukan dalam kebijakan AWS terkelola. Jika AWS memperbarui izin yang ditentukan dalam kebijakan AWS terkelola, pembaruan akan memengaruhi semua identitas utama (pengguna, grup, dan peran) yang dilampirkan kebijakan tersebut. AWS kemungkinan besar akan memperbarui kebijakan AWS terkelola saat baru Layanan AWS diluncurkan atau operasi API baru tersedia untuk layanan yang ada.

Untuk informasi selengkapnya, lihat Kebijakan terkelola AWS dalam Panduan Pengguna IAM.

AWS Ubah pembaruan untuk AWS kebijakan terkelola

Lihat detail tentang pembaruan kebijakan AWS terkelola untuk AWS Transform sejak 1 Maret 2021.

Ubah	Deskripsi	Date
AWSTransformInfrastructureExecutorAccessBatch – Kebijakan baru	Menambahkan kebijakan AWS terkelola baru yang memberikan izin yang diperlukan untuk menjalankan penilaian dan AWS transform asi Modernisasi Berkelanj utan Transform menggunak an AWS Batch, termasuk mengunggah kode sumber, mengambil hasil, memantau status pekerjaan Batch, membaca log, dan mengelola jadwal transformasi.	Juli 20, 2026

Ubah	Deskripsi	Date
AWSTransformInfrastructureExecutorAccessEC2 – Kebijakan baru	Menambahkan kebijakan AWS terkelola baru yang memberikan izin yang diperlukan untuk menjalankan penilaian dan AWS transformasi Modernisasi Berkelanjutan Transform menggunakan Amazon EC2, termasuk mengunggah kode sumber, mengambil hasil, memantau status pekerjaan, dan mengelola jadwal transformasi.	Juli 20, 2026
AWSTransformSecurityAgentExecutorAccess – Kebijakan baru	Menambahkan kebijakan AWS terkelola baru yang memberikan izin AWS Transform Continuous Modernization yang diperlukan untuk memanggil layanan Agen AWS Keamanan untuk tinjauan dan remediasi keamanan kode otomatis, termasuk mengunggah artefak pemindaian dan mengambil temuan.	Juni 30, 2026
AWSServiceRoleForAWSTransformCustom— Kebijakan yang diperbarui	Menambahkan izin CloudWatch Log untuk memungkinkan AWS Transform custom mempublikasikan log ke grup `/aws/TransformCustom` log di akun Anda.	5 Mei 2026

Ubah	Deskripsi	Date
AWSTransformCustom ExecuteTransformations— Kebijakan yang diperbarui	Menambahkan izin untuk membuat peran (`AWSServiceRoleForAWSTransformCustom`) AWS Transform custom service-linked () untuk mengaktifkan emisi CloudWatch metrik ke akun pelanggan.	April 27, 2026
AWSTransformCustom ManageTransformations— Kebijakan yang diperbarui	Menambahkan izin untuk membuat peran (`AWSServiceRoleForAWSTransformCustom`) AWS Transform custom service-linked () untuk mengaktifkan emisi CloudWatch metrik ke akun pelanggan.	April 27, 2026
AWSTransformCustom FullAccess— Kebijakan yang diperbarui	Menambahkan izin untuk membuat peran (`AWSServiceRoleForAWSTransformCustom`) AWS Transform custom service-linked () untuk mengaktifkan emisi CloudWatch metrik ke akun pelanggan.	April 7, 2026
AWSServiceRoleForAWSTransformCustom – Kebijakan baru	Menambahkan kebijakan AWS terkelola baru untuk peran AWS Transform custom service-linked. Kebijakan ini memungkinkan AWS Transform custom untuk memublikasikan CloudWatch metrik ke akun Anda.	Maret 23, 2026

Ubah	Deskripsi	Date
DBModProvisioningAndMigration – Kebijakan baru	Kebijakan ini memberikan kemampuan penyediaan dan migrasi database.	Maret 24, 2026
DBModDiscoveryAndAssessment – Kebijakan baru	Menambahkan kebijakan AWS terkelola baru yang menyediakan penemuan modernisasi database yang komprehensif dan kemampuan penilaian.	Maret 24, 2026
AWSTransformCustomFullAccess – Kebijakan baru	Menambahkan kebijakan AWS terkelola baru yang menyediakan akses penuh ke kustom AWS Transform.	Desember 5, 2025
AWSTransformCustomExecuteTransformations – Kebijakan baru	Menambahkan kebijakan AWS terkelola baru yang menyediakan akses untuk mengeksekusi transformasi di kustom AWS Transform.	Desember 5, 2025
AWSTransformCustomManageTransformations – Kebijakan baru	Menambahkan kebijakan AWS terkelola baru yang menyediakan akses untuk membuat, memperbarui, membaca, dan menghapus sumber daya AWS transformasi di kustom Transform, serta menjalankan transformasi.	Desember 5, 2025

Ubah	Deskripsi	Date
AWSServiceRoleForAWSTransform— Kebijakan yang diperbarui	Menambahkan izin untuk mengakses rahasia terkait layanan AWS Transform yang digunakan untuk menyimpan rahasia klien untuk penyedia identitas eksternal. Menambahkan izin untuk membuat kasus dukungan premium dari aplikasi web AWS Transform.	Desember 1, 2025
AWSTransformApplicationECSDeploymentPolicy— Kebijakan yang diperbarui	Menambahkan izin inspeksi peran IAM, pembuatan peran terkait layanan ECS, dan izin KMS untuk dukungan enkripsi ECR.	November 22, 2025
AWSTransformApplicationDeploymentPolicy— Kebijakan yang diperbarui	Menambahkan izin jaringan EC2, izin inspeksi peran IAM, izin daftar bucket S3, dan dukungan enkripsi KMS untuk kemampuan penerapan yang ditingkatkan.	November 22, 2025
AWSServiceRoleForAWSTransform— Kebijakan yang diperbarui	Menambahkan dukungan untuk kunci yang dikelola pelanggan di IAM Identity Center.	September 17, 2025

Ubah	Deskripsi	Date
AWSTransformApplicationDeploymentPolicy – Kebijakan baru	Menambahkan kebijakan AWS terkelola baru yang memungkinkan AWS Transform menerapkan aplikasi.NET yang diubah dengan membuat dan mengelola instans CloudFormation , tumpukan, dan sumber daya terkait Amazon EC2.	Agustus 28, 2025
AWSServiceRoleForAWSTransform— Kebijakan yang diperbarui	Menambahkan kebijakan baru.	15 Mei 2025

AWS kebijakan terkelola: AWSServiceRoleForAWSTransform

Kebijakan ini dilampirkan pada peran AWSServiceRoleForAWSTransformterkait layanan (SLR).

Detail izin

Untuk melihat detail izin kebijakan, lihat AWSServiceRoleForAWSTransformdi Panduan Referensi Kebijakan AWS Terkelola.

AWS kebijakan terkelola: AWSServiceRoleForAWSTransformCustom

Kebijakan ini dilampirkan pada peran AWSServiceRoleForAWSTransformCustomterkait layanan (SLR). Peran ini memungkinkan AWS Transform custom untuk mempublikasikan CloudWatch metrik dan log ke akun Anda atas nama Anda.

Deskripsi

Kebijakan ini mencakup izin berikut:

- Amazon CloudWatch — Memungkinkan metrik penerbitan CloudWatch di bawah `AWS/TransformCustom` namespace. Ini memungkinkan pemantauan jumlah transformasi, latensi, dan kode status di dasbor Anda CloudWatch .

- Amazon CloudWatch Logs — Memungkinkan pembuatan grup log, aliran log, menyetel kebijakan penyimpanan, dan memublikasikan peristiwa log ke grup `/aws/TransformCustom` log. Ini memberikan visibilitas ke masalah yang dihadapi selama transformasi.

AWS kebijakan terkelola: AWSTransformApplicationDeploymentPolicy

Kebijakan ini memungkinkan AWS Transform untuk menerapkan aplikasi.NET yang telah diubah dengan membuat dan mengelola instans CloudFormation , tumpukan, dan sumber daya terkait Amazon EC2.

Deskripsi

Kebijakan ini mencakup izin berikut:

- CloudFormation— Memungkinkan membuat, memperbarui, menghapus, dan mendeskripsikan CloudFormation tumpukan dengan nama yang dimulai dengan AWSTransForm. Operasi tumpukan dibatasi untuk sumber daya yang ditandai dengan CreatedBy: AWSTransForm dan terbatas pada akun yang sama. AWS
- Amazon EC2 — Memungkinkan mendeskripsikan VPC, subnet, grup keamanan, gambar, instance, tabel rute, dan gateway internet. Izin menjalankan, memulai, menghentikan, mengakhiri, dan memodifikasi instans EC2, tetapi hanya ketika dipanggil. CloudFormation Pembuatan tag dibatasi untuk kunci tag tertentu yang diizinkan dan hanya selama CloudFormation operasi.
- AWS Identity and Access Management (IAM) - Memungkinkan mendapatkan dan meneruskan peran IAM tertentu untuk instance penerapan AWSTransForm. Termasuk izin untuk memeriksa kebijakan peran dan lampiran. Akses dibatasi untuk AWS akun yang sama.
- Amazon EC2 Systems Manager (SSM) — Memungkinkan pengambilan parameter Amazon Linux AMI dari penyimpanan parameter AWS yang dikelola dan mengirim perintah ke instance. AWSTransform-tagged
- Amazon S3 — Memungkinkan mengelola objek di bucket penerapan AWSTransform, termasuk mencantumkan bucket dan mendapatkan lokasi bucket dalam akun yang sama. AWS
- AWS Key Management Service (KMS) - Memungkinkan operasi enkripsi dan dekripsi menggunakan kunci KMS yang ditandai untuk AWSTransForm, dengan batasan penggunaan layanan S3 dan EC2.

Kebijakan ini menerapkan akses hak istimewa paling sedikit melalui izin tingkat sumber daya, kondisi berbasis tag, penggunaan pembatasan kontrol layanan, pembatasan tingkat akun`aws:CalledVia`,

dan pernyataan penolakan eksplisit untuk mencegah modifikasi tag yang tidak sah di luar operasi. CloudFormation

Detail izin

Untuk melihat detail izin kebijakan, lihat AWSTransformApplicationDeploymentPolicydi Panduan Referensi Kebijakan AWS Terkelola.

AWS kebijakan terkelola: AWSTransformApplicationECSDeploymentPolicy

Kebijakan ini memungkinkan AWS Transform untuk menerapkan aplikasi yang diubah ke Amazon ECS dengan membuat dan mengelola klaster, layanan, tugas, dan sumber daya terkait ECS.

Deskripsi

Kebijakan ini mencakup izin berikut:

- CloudFormation— Memungkinkan membuat, memperbarui, menghapus, dan mendeskripsikan CloudFormation tumpukan dengan nama yang dimulai dengan AWSTransForm. Operasi tumpukan dibatasi untuk sumber daya yang ditandai dengan CreatedBy: AWSTransForm dan terbatas pada akun yang sama. AWS
- Amazon ECS - Memungkinkan membuat, memperbarui, dan menghapus cluster, layanan, dan definisi tugas ECS. Izin menjalankan tugas, daftar tugas, dan menjelaskan status tugas. Semua operasi dibatasi untuk sumber daya dengan nama yang dimulai dengan AWSTransForm dan ditandai dengan: AWSTransForm. CreatedBy
- AWS Identity and Access Management (IAM) - Memungkinkan mendapatkan dan melewati peran IAM tertentu untuk tugas ECS (AWSTransform-Deploy-ECS-Task-Role dan). AWSTransform-Deploy-ECS-Execution-Role Termasuk izin untuk memeriksa kebijakan peran dan membuat peran terkait layanan ECS bila diperlukan.
- Amazon CloudWatch Logs - Memungkinkan membuat, menghapus, dan mengelola grup log dengan nama yang dimulai dengan/aws/ecs/awsTransform. Izin mengambil peristiwa log untuk pemecahan masalah aplikasi yang diterapkan.
- Amazon ECR - Memungkinkan membuat repositori kontainer dengan nama yang dimulai dengan awstransform untuk menyimpan gambar wadah aplikasi.
- AWS Key Management Service (KMS) - Memungkinkan membuat hibah dan menghasilkan kunci data untuk enkripsi ECR saat menggunakan kunci KMS yang dikelola pelanggan.

Kebijakan ini menerapkan akses hak istimewa paling sedikit melalui izin tingkat sumber daya, kondisi berbasis tag, dan pembatasan tingkat akun untuk memastikan operasi terbatas pada sumber daya dalam akun yang sama. AWSTransform-managed AWS

Detail izin

Untuk melihat detail izin kebijakan, lihat AWSTransformApplicationECSDeploymentPolicydi Panduan Referensi Kebijakan AWS Terkelola.

AWS kebijakan terkelola: AWSTransformCustomFullAccess

Kebijakan ini menyediakan akses penuh ke AWS Transform custom.

Deskripsi

Kebijakan ini mencakup izin berikut:

- AWS Transform Custom - Memungkinkan semua tindakan pada semua AWS Transform sumber daya kustom. Ini memberikan akses administratif lengkap ke layanan.
- AWS Identity and Access Management (IAM) - Memungkinkan pembuatan peran terkait layanan kustom AWS Transform (). `AWSServiceRoleForAWSTransformCustom` Peran ini diperlukan untuk AWS Transform custom untuk memancarkan CloudWatch metrik dan log ke akun Anda. Izin dicakup untuk hanya mengizinkan pembuatan peran terkait layanan khusus ini.

Detail izin

Untuk melihat detail izin kebijakan, lihat AWSTransformCustomFullAccessdi Panduan Referensi Kebijakan AWS Terkelola.

AWS kebijakan terkelola: AWSTransformCustomExecuteTransformations

Kebijakan ini menyediakan akses untuk mengeksekusi transformasi dalam kustom AWS Transform.

Deskripsi

Kebijakan ini mencakup izin berikut:

- AWS Transform Custom - Memungkinkan streaming percakapan, mengeksekusi transformasi, dan mengelola kampanye. Termasuk izin untuk mendapatkan detail kampanye, memperbarui status repositori kampanye, dan memperbarui kampanye.

- AWS Identity and Access Management (IAM) - Memungkinkan pembuatan peran terkait layanan kustom AWS Transform (). `AWSServiceRoleForAWSTransformCustom` Peran ini diperlukan untuk AWS Transform custom untuk memancarkan CloudWatch metrik dan log ke akun Anda. Izin dicakup untuk hanya mengizinkan pembuatan peran terkait layanan khusus ini.

Detail izin

Untuk melihat detail izin kebijakan, lihat AWSTransformCustomExecuteTransformationsdi Panduan Referensi Kebijakan AWS Terkelola.

AWS kebijakan terkelola: AWSTransformCustomManageTransformations

Kebijakan ini menyediakan akses untuk membuat, memperbarui, membaca, dan menghapus sumber daya AWS transformasi di kustom Transform, serta menjalankan transformasi.

Deskripsi

Kebijakan ini mencakup izin berikut:

- AWS Transform Custom — Memungkinkan pengelolaan sumber daya transformasi yang komprehensif termasuk streaming percakapan, mengeksekusi transformasi, dan mengelola paket transformasi. Izin membuat, mendapatkan, dan menghapus URL paket transformasi dan menyelesaikan unggahan paket.
- Manajemen Pengetahuan — Memungkinkan daftar, mendapatkan, menghapus, dan memperbarui item pengetahuan serta konfigurasi dan statusnya.
- Manajemen Kampanye - Memungkinkan mendapatkan detail kampanye, memperbarui status repositori kampanye, dan memperbarui kampanye.
- Resource Tagging - Memungkinkan daftar, menambahkan, dan menghapus tag untuk AWS Transform sumber daya kustom.
- AWS Identity and Access Management (IAM) - Memungkinkan pembuatan peran terkait layanan kustom AWS Transform (). `AWSServiceRoleForAWSTransformCustom` Peran ini diperlukan untuk AWS Transform custom untuk memancarkan CloudWatch metrik dan log ke akun Anda. Izin dicakup untuk hanya mengizinkan pembuatan peran terkait layanan khusus ini.

Detail izin

Untuk melihat detail izin kebijakan, lihat AWSTransformCustomManageTransformationsdi Panduan Referensi Kebijakan AWS Terkelola.

AWS kebijakan terkelola: AWSTransformInfrastructureExecutorAccessBatch

Kebijakan ini memberikan izin yang diperlukan untuk menjalankan penilaian dan AWS transformasi Transform Continuous Modernization menggunakan Batch. AWS Lampirkan kebijakan ini ke peran IAM di akun Anda. CLI dan agen AWS Transform Continuous Modernization mengambil peran ini untuk mengunggah kode sumber, mengambil hasil, memantau status pekerjaan Batch, membaca log, dan mengelola jadwal transformasi.

Deskripsi

Kebijakan ini mencakup izin berikut:

- AWS Lambda - Memungkinkan memanggil dan mengambil konfigurasi fungsi `atx` Lambda -prefixed yang mengatur alur kerja transformasi.
- Amazon S3 - Memungkinkan mengunggah kode sumber ke `atx-source-code-*` bucket dan mengunduh hasil transformasi dari dan bucket. `atx-custom-output-*` `atx-ct-output-*`
- AWS Key Management Service (KMS) - Memungkinkan enkripsi dan dekripsi data menggunakan kunci enkripsi AWS Transform (). `alias/atx-encryption-key`
- Amazon CloudWatch Logs - Memungkinkan membaca pekerjaan Batch dan peristiwa log eksekusi Lambda untuk pemantauan dan debugging transformasi.
- Amazon CloudWatch — Memungkinkan mendapatkan dan mencantumkan dasbor AWS Transform CLI untuk visibilitas operasional.
- AWS Batch — Memungkinkan mendeskripsikan dan mencantumkan lingkungan komputasi, antrian pekerjaan, definisi pekerjaan, dan status pekerjaan.
- CloudFormation— Memungkinkan mendeskripsikan dan mencantumkan tumpukan infrastruktur AWS Transform untuk memeriksa status penerapan.
- Resource Groups Tagging API — Memungkinkan pengambilan sumber daya yang ditandai untuk menemukan infrastruktur AWS Transform di akun Anda.
- AWS Secrets Manager— Memungkinkan mengambil dan menjelaskan rahasia AWS Transform yang disimpan di bawah awalan. `atx/`
- Amazon EventBridge Scheduler - Memungkinkan mengelola pekerjaan transformasi terjadwal dalam grup `atx-ct` jadwal.
- Amazon EC2 - Memungkinkan mendeskripsikan VPC, subnet, grup keamanan, tabel rute, dan konfigurasi gateway NAT untuk jaringan lingkungan komputasi Batch.
- AWS Identity and Access Management (IAM) - Memungkinkan membaca konfigurasi peran `ATX` -prefixed dan `Atx` -prefixed untuk memvalidasi penyiapan infrastruktur, dan meneruskan ke

`AtxSchedulerInvocationRole` Amazon EventBridge Scheduler sehingga dapat memanggil pekerjaan terjadwal.

Kebijakan ini menerapkan akses hak istimewa paling sedikit melalui izin tingkat sumber daya, kondisi berbasis tag, kondisi, dan kondisi cakupan layanan untuk memastikan operasi terbatas pada `aws:ResourceAccount` Transform resource dalam akun `iam:PassedToService` yang sama. AWS AWS

Detail izin

Untuk melihat detail izin kebijakan, lihat AWSTransformInfrastructureExecutorAccessBatchdi Panduan Referensi Kebijakan AWS Terkelola.

AWS kebijakan terkelola: AWSTransformInfrastructureExecutorAccessEC2

Kebijakan ini memberikan izin yang diperlukan untuk menjalankan penilaian dan AWS transformasi Transform Continuous Modernization menggunakan Amazon EC2. Lampirkan kebijakan ini ke peran IAM di akun Anda. CLI dan agen AWS Transform Continuous Modernization mengambil peran ini untuk mengunggah kode sumber, mengambil hasil, memantau status pekerjaan, dan mengelola jadwal transformasi.

Deskripsi

Kebijakan ini mencakup izin berikut:

- CloudFormation— Memungkinkan mendeskripsikan tumpukan infrastruktur AWS Transform, peristiwa tumpukan, sumber daya tumpukan, dan status deteksi drift, daftar tumpukan, dan memvalidasi CloudFormation templat sebelum penerapan.
- Amazon EC2 — Memungkinkan mendeskripsikan instans, AMI, VPC, subnet, grup keamanan, pasangan kunci, tabel rute, gateway NAT, dan gateway internet. Izin memulai dan menghentikan instans EC2 yang ditandai dengan. `atx-remote-infra: true`
- Amazon EC2 Systems Manager (SSM) — Memungkinkan pemantauan status eksekusi perintah, serta menjalankan perintah dan sesi AWS Transform-tagged pada instance, termasuk menjalankan dokumen. `AWS-RunShellScript`
- Amazon S3 — Memungkinkan mengelola kode sumber dan artefak keluaran di bucket AWS Transform, termasuk mendapatkan, meletakkan, dan menghapus objek dan mencantumkan bucket.

- AWS Key Management Service (KMS) - Memungkinkan enkripsi dan dekripsi data menggunakan kunci enkripsi AWS Transform (). `alias/atx-encryption-key`
- AWS Secrets Manager— Memungkinkan mengambil dan mendeskripsikan rahasia transformasi, seperti kredensyal repositori, disimpan di bawah awalan. `atx/`
- Amazon EventBridge Scheduler - Memungkinkan mengelola pekerjaan transformasi terjadwal dalam grup `atx-ct` jadwal.
- Resource Groups Tagging API — Memungkinkan pengambilan sumber daya yang ditandai untuk menemukan infrastruktur AWS Transform di akun Anda.
- AWS Identity and Access Management (IAM) - Memungkinkan meneruskan `atx-transform-role*` peran ke Amazon EC2 dan `AtxSchedulerInvocationRole` ke EventBridge Amazon Scheduler, serta membaca konfigurasi profil peran dan instans untuk instans EC2.

Kebijakan ini menerapkan akses hak istimewa paling sedikit melalui izin tingkat sumber daya, kondisi berbasis tag, kondisi, dan kondisi cakupan layanan untuk memastikan operasi terbatas pada `aws:ResourceAccount` Transform resource dalam akun `iam:PassedToService` yang sama. AWS AWS

Detail izin

Untuk melihat detail izin kebijakan, lihat AWSTransformInfrastructureExecutorAccessEC2di Panduan Referensi Kebijakan AWS Terkelola.

AWS kebijakan terkelola: AWSTransformSecurityAgentExecutorAccess

Kebijakan ini memberikan izin AWS Transform Continuous Modernization yang diperlukan untuk memanggil layanan Agen AWS Keamanan untuk tinjauan dan remediasi keamanan kode otomatis. Lampirkan kebijakan ini ke peran IAM di akun Anda yang diasumsikan oleh CLI dan agen AWS Transform Continuous Modernization untuk mengunggah artefak pemindaian dan mengambil temuan.

Deskripsi

Kebijakan ini mencakup izin berikut:

- AWS Agen Keamanan — Memungkinkan daftar ruang agen, membuat ulasan kode, memulai pekerjaan peninjauan kode, mencantumkan dan mengambil temuan, dan memulai remediasi kode. Tindakan ini dicakup ke ruang agen di akun Anda.

- Amazon S3 — Memungkinkan mengunduh hasil pemindaian dan temuan dari `kct-security-agent-*` bucket, dan mengunggah kode sumber untuk pemindaian keamanan ke `security-scans/*` awalan di dalam bucket tersebut.
- AWS Identity and Access Management (IAM) — Memungkinkan meneruskan `security-agent-*` peran ke layanan Agen AWS Keamanan sehingga dapat melakukan pemindaian.

Kebijakan ini menerapkan akses hak istimewa paling sedikit melalui izin tingkat sumber daya, bucket S3 dan awalan kunci, dan kondisi, `aws:ResourceAccount` dan `s3:ResourceAccount` kondisi cakupan layanan untuk memastikan operasi terbatas pada sumber daya Agen Keamanan dalam akun yang sama`iam:PassedToService`. AWS AWS

Detail izin

Untuk melihat detail izin kebijakan, lihat AWSTransformSecurityAgentExecutorAccessdi Panduan Referensi Kebijakan AWS Terkelola.

AWS kebijakan terkelola: DBModDiscoveryAndAssessment

Kebijakan ini menyediakan penemuan modernisasi database yang komprehensif dan kemampuan penilaian untuk AWS Transform.

Deskripsi

Kebijakan ini mencakup izin berikut:

- Amazon EC2 — Memungkinkan mendeskripsikan komponen infrastruktur termasuk instans, VPC, subnet, grup keamanan, zona ketersediaan, titik akhir VPC, dan gateway internet.
- Amazon RDS - Memungkinkan mendeskripsikan instance database, cluster, dan grup subnet. Memungkinkan memodifikasi grup subnet DB dalam AWS akun yang sama untuk persiapan migrasi.
- Amazon RDS Data API - Memungkinkan mengaktifkan dan menonaktifkan titik akhir HTTP dan mengeksekusi pernyataan SQL pada cluster database yang ditandai untuk proyek modernisasi database.
- AWS DMS— Memungkinkan mendeskripsikan titik akhir, contoh replikasi, tugas, grup subnet, dan instance yang dapat diurutkan. Memungkinkan daftar penyedia data, profil instance, dan proyek migrasi. Memungkinkan mendeskripsikan statistik tabel, penilaian berjalan, dan operasi model metadata. Penjelasan rinci dan operasi metadata dibatasi untuk sumber daya yang ditandai untuk proyek modernisasi database.

- — Memungkinkan daftar rahasia untuk menemukan kredensil database.
- AWS Identity and Access Management (IAM) - Memungkinkan memeriksa peran AWS DMS layanan tertentu dan kebijakan terlampirnya. Termasuk akses ke AWS DMS kebijakan yang AWS dikelola baca.
- AWS Key Management Service (KMS) - Memungkinkan daftar alias kunci dan menjelaskan kunci. Memungkinkan dekripsi rahasia melalui integrasi, terbatas pada akun yang sama AWS .

Kebijakan ini menerapkan akses hak istimewa paling sedikit melalui izin tingkat sumber daya, kondisi berbasis tag, dan pembatasan tingkat akun untuk memastikan operasi terbatas pada sumber daya proyek modernisasi basis data dalam akun yang sama. AWS

Detail izin

Untuk melihat detail izin kebijakan, lihat DBModDiscoveryAndAssessmentdi Panduan Referensi Kebijakan AWS Terkelola.

AWS kebijakan terkelola: DBModProvisioningAndMigration

Kebijakan ini menyediakan kemampuan penyediaan dan migrasi database untuk proyek modernisasi database AWS Transform. Ini mencakup izin untuk membuat dan mengelola infrastruktur migrasi, menyediakan basis data target, dan menyimpan data migrasi.

Deskripsi

Kebijakan ini mencakup izin berikut:

- AWS DMS— Memungkinkan membuat dan mengelola grup subnet replikasi, profil instance, penyedia data, proyek migrasi, titik akhir, instance replikasi, dan tugas replikasi. Termasuk operasi konversi skema seperti impor model metadata, konversi, ekspor, dan penilaian. Memungkinkan manajemen siklus hidup instance replikasi (membuat, menghapus, memodifikasi, reboot) dan tugas replikasi (menghapus, memulai, menghentikan, menilai). Semua operasi tulis dibatasi pada sumber daya yang ditandai untuk proyek modernisasi database.
- Amazon RDS - Memungkinkan pembuatan grup subnet database, cluster database, dan instance database. Semua sumber daya harus ditandai untuk proyek modernisasi database.
- — Memungkinkan membuat, memperbarui, dan menandai rahasia dengan awalan penamaan modernisasi database. Memungkinkan mengambil nilai rahasia dan menjelaskan rahasia untuk sumber daya yang ditandai.

- Amazon S3 - Memungkinkan membuat dan mengelola bucket dan objek S3 untuk penyimpanan data migrasi. Termasuk operasi penandaan bucket, pembuatan versi, dan siklus hidup objek. Dibatasi untuk bucket dengan awalan penamaan modernisasi database.
- AWS Identity and Access Management (IAM) - Memungkinkan meneruskan peran AWS DMS layanan tertentu ke AWS DMS dan layanan konversi skema. Memungkinkan pembuatan peran terkait layanan Amazon RDS yang diperlukan untuk operasi database.

Kebijakan ini menerapkan akses hak istimewa paling sedikit melalui izin tingkat sumber daya, kondisi berbasis tag, dan pembatasan tingkat akun untuk memastikan operasi terbatas pada sumber daya proyek modernisasi basis data dalam akun yang sama. AWS

Detail izin

Untuk melihat detail izin kebijakan, lihat DBModProvisioningAndMigrationdi Panduan Referensi Kebijakan AWS Terkelola.

AWS Referensi izin transformasi

Bagian ini memberikan informasi tentang API yang digunakan oleh AWS Transform, dan apa yang mereka lakukan.

Topik

- AWS Transformasi API

AWS Transformasi API

- mengubah: BatchGetMessage
- mengubah: BatchGetUserDetails
- mengubah: CompleteArtifactUpload
- mengubah: CreateArtifactDownloadUrl
- mengubah: CreateArtifactUploadUrl
- mengubah: CreateAssetDownloadUrl
- mengubah: CreateConnector
- mengubah: CreateFeedback
- mengubah: CreateJob

- mengubah: CreateSession
- mengubah: CreateWorkspace
- mengubah: DeleteConnector
- mengubah: DeleteJob
- mengubah: DeleteSelfRoleMappings
- mengubah: DeleteUserRoleMappings
- mengubah: DeleteWorkspace
- mengubah: DetectIsAllowedForOperation
- mengubah: GetConnector
- mengubah: GetFeedbackQuestions
- mengubah: GetHitlTask
- mengubah: GetJob
- mengubah: GetLoginRedirectUri
- mengubah: GetUserDetails
- mengubah: GetUserPreferences
- mengubah: GetWebAppUrl
- mengubah: GetWorkspace
- mengubah: ListArtifacts
- mengubah: ListConnectors
- mengubah: ListHitlTasks
- mengubah: ListJobPlanSteps
- mengubah: ListJobs
- mengubah: ListMessages
- mengubah: ListPlanUpdates
- mengubah: ListUserRoleMappings
- mengubah: ListWorklogs
- mengubah: ListWorkspaces
- mengubah: PutUserPreferences
- mengubah: PutUserRoleMappings
- mengubah: RevokeSession

- mengubah: SearchUsers
- mengubah: SearchUsersTypeahead
- mengubah: SendMessage
- mengubah: StartJob
- mengubah: StopJob
- mengubah: SubmitCriticalHitlTask
- mengubah: SubmitStandardHitlTask
- mengubah: TestReleaseTraitPreProd
- mengubah: TestReleaseTraitProd
- mengubah: UpdateHitlTask
- mengubah: UpdateJob
- mengubah: UpdateWorkspace
- mengubah: VerifySession

Validasi kepatuhan untuk AWS Transformasi

Untuk mempelajari apakah an Layanan AWS berada dalam lingkup program kepatuhan tertentu, lihat Layanan AWS di Lingkup oleh Program Kepatuhan Layanan AWS dan pilih program kepatuhan yang Anda minati. Untuk informasi umum, lihat Program AWS Kepatuhan Program AWS .

Anda dapat mengunduh laporan audit pihak ketiga menggunakan AWS Artifact. Untuk informasi selengkapnya, lihat Mengunduh Laporan di AWS Artifact .

Tanggung jawab kepatuhan Anda saat menggunakan Layanan AWS ditentukan oleh sensitivitas data Anda, tujuan kepatuhan perusahaan Anda, dan hukum dan peraturan yang berlaku. Untuk informasi selengkapnya tentang tanggung jawab kepatuhan Anda saat menggunakan Layanan AWS, lihat Dokumentasi AWS Keamanan.

Ketahanan di AWS Transformasi

Infrastruktur AWS global dibangun di sekitar Wilayah AWS dan Availability Zones. Wilayah AWS menyediakan beberapa Availability Zone yang terpisah secara fisik dan terisolasi, yang terhubung dengan latensi rendah, throughput tinggi, dan jaringan yang sangat redundan. Dengan Zona Ketersediaan, Anda dapat merancang serta mengoperasikan aplikasi dan basis data yang secara otomatis melakukan fail over di antara zona tanpa gangguan. Zona Ketersediaan

memiliki ketersediaan dan toleransi kesalahan yang lebih baik, dan dapat diskalakan dibandingkan infrastruktur pusat data tunggal atau multi tradisional.

Untuk informasi selengkapnya tentang Wilayah AWS dan Availability Zone, lihat Infrastruktur AWS Global.

Selain infrastruktur AWS global, AWS Transform menawarkan beberapa fitur untuk membantu mendukung ketahanan data dan kebutuhan cadangan Anda.

AWS Transform dan titik akhir antarmuka (AWS PrivateLink)

Anda dapat membuat koneksi pribadi antara VPC Anda dan AWS Transform dengan membuat antarmuka VPC endpoint. Endpoint antarmuka didukung oleh AWS PrivateLink, teknologi yang memungkinkan Anda mengakses AWS Transform konsol secara pribadi tanpa gateway internet, perangkat NAT, koneksi VPN, atau koneksi. Direct Connect Lalu lintas antara VPC Anda dan AWS Transform tidak meninggalkan jaringan Amazon.

Setiap titik akhir antarmuka diwakili oleh satu atau beberapa Antarmuka Jaringan Elastis di subnet Anda.

Untuk informasi selengkapnya, lihat Antarmuka VPC endpoint (AWS PrivateLink) dalam Panduan Pengguna Amazon VPC.

> **Note**
>
> Untuk PrivateLink dokumentasi AWS Transform kustom, lihatAWS Transform titik akhir kustom dan antarmuka (AWS PrivateLink).

Pertimbangan untuk AWS Transform Titik akhir VPC

Sebelum menyiapkan titik akhir VPC antarmuka AWS Transform, pastikan Anda meninjau properti dan batasan titik akhir Antarmuka di Panduan Pengguna Amazon VPC.

Prasyarat

Sebelum Anda memulai salah satu prosedur di bawah ini, pastikan Anda memiliki yang berikut:

- AWS Akun dengan izin yang sesuai untuk membuat dan mengonfigurasi sumber daya.
- VPC sudah dibuat di akun Anda AWS .

- Keakraban dengan AWS layanan, terutama Amazon AWS Transform VPC dan.

Membuat titik akhir VPC antarmuka untuk AWS Transform

Anda dapat membuat titik akhir VPC untuk AWS Transform layanan menggunakan konsol VPC Amazon atau (). AWS Command Line Interface AWS CLI Untuk informasi selengkapnya, lihat Membuat titik akhir antarmuka dalam Panduan Pengguna Amazon VPC.

AWS Transform mendukung layanan endpoint VPC berikut:

Nama layanan	Titik akhir	Catatan
Bidang kontrol	`com.amazonaws.`*`region`*`.transform`	
API Agentik	`com.amazonaws.`*`region`*`.transform-agents`	
Aplikasi web	`com.amazonaws.`*`region`*`.api.transform`	Diperlukan untuk aplikasi AWS Transform web. Titik akhir ini harus mengaktifkan DNS pribadi (opsi Aktifkan nama DNS) sehingga `api.transform.`*`region`*`.on.aws` menyelesaikan ke alamat IP pribadi di VPC Anda.

Ganti *`region`* dengan AWS Wilayah tempat AWS Transform profil Anda diinstal, misalnya, com.amazonaws.us-east-1.transform.

> **Note**
>
> Jika Anda menggunakan aplikasi AWS Transform web, `api.transform` titik akhir diperlukan. Untuk panduan penyiapan lengkap, lihatMengakses AWS Transform aplikasi web dari VPC.

Untuk informasi selengkapnya, lihat Wilayah yang Didukung untuk AWS Transform dan Mengakses layanan melalui titik akhir antarmuka di Panduan Pengguna Amazon VPC.

Menggunakan komputer lokal untuk menyambung ke AWS Transform titik akhir

Bagian ini menjelaskan proses penggunaan komputer lokal untuk terhubung AWS Transform melalui AWS PrivateLink titik akhir di AWS VPC Anda.

1. Buat koneksi VPN antara perangkat lokal dan VPC Anda.
2. Buat titik akhir VPC antarmuka untuk. AWS Transform
3. Siapkan titik akhir Amazon Route 53 masuk. Ini akan memungkinkan Anda untuk menggunakan nama DNS AWS Transform titik akhir Anda dari perangkat lokal Anda.

Mengakses AWS Transform aplikasi web dari VPC

Saat Anda menggunakan AWS PrivateLink untuk mengakses AWS Transform API secara pribadi dari VPC Anda, aplikasi web memerlukan konfigurasi jaringan tambahan. Ini melayani konten statis (HTML, JavaScript, CSS) melalui CloudFront, yang membutuhkan konektivitas internet. Panggilan API dari aplikasi web melalui titik akhir VPC Anda dan tetap sepenuhnya pribadi.

Panduan ini menunjukkan cara mengonfigurasi jalan keluar internet yang dikontrol dari VPC Anda sehingga aplikasi web dapat memuat sambil menjaga VPC Anda terkunci hanya ke domain yang diperlukan.

Cara kerjanya

Aplikasi AWS Transform web menggunakan dua jalur jaringan:

- Panggilan API — Saat Anda berinteraksi dengan aplikasi web (memulai pekerjaan, melihat ruang kerja, dan sebagainya), browser mengirimkan permintaan API ke`api.transform.region.on.aws`. Dengan titik akhir `com.amazonaws.region.api.transform` VPC dan DNS pribadi diaktifkan, permintaan ini diselesaikan ke alamat IP pribadi di VPC Anda dan tidak pernah meninggalkan jaringan. AWS
- Konten statis — HTML aplikasi web JavaScript,, dan file CSS disajikan CloudFront melalui`tenant-id.transform.region.on.aws`. Memuat file-file ini memerlukan konektivitas internet karena pengiriman CloudFront konten hanya tersedia melalui internet publik.

- Otentikasi - Penggunaan alur masuk AWS IAM Identity Center*region*`.signin.aws`, yang juga membutuhkan konektivitas internet.

Untuk mengaktifkan aplikasi web sambil menjaga keamanan, Anda membuat jalur keluar terkontrol menggunakan AWS Network Firewall dengan pemfilteran berbasis domain. Hal ini memungkinkan VPC Anda untuk mencapai hanya domain tertentu yang diperlukan oleh aplikasi web sambil memblokir semua lalu lintas internet lainnya.

Arsitektur

Diagram berikut menunjukkan jalur jaringan untuk lalu lintas aplikasi web:

```
EC2 Instance / Workspace (Private Subnet)
    |
    | Route: 0.0.0.0/0 # Network Firewall Endpoint
    v
AWS Network Firewall (Firewall Subnet)
    | Allows: *.cloudfront.net, *.transform.<region>.on.aws,
    |         <region>.signin.aws, SSO domains, S3 presigned URLs
    | Blocks: everything else (TLS SNI inspection)
    |
    | Route: 0.0.0.0/0 # NAT Gateway
    v
NAT Gateway (Public Subnet)
    |
    | Route: 0.0.0.0/0 # Internet Gateway
    v
Internet Gateway # CloudFront Edge Locations
```

Important

Network Firewall harus melihat kedua arah lalu lintas (symmetric routing) untuk melakukan inspeksi TLS SNI. Anda harus mengonfigurasi rute kembali di subnet NAT Gateway yang mengirimkan lalu lintas yang ditujukan untuk subnet pribadi kembali melalui firewall. Tanpa ini, aturan penyaringan berbasis domain tidak akan berfungsi.

Prasyarat

Sebelum Anda mulai, pastikan Anda memiliki:

- AWS Akun dengan izin untuk membuat sumber daya VPC, Network Firewall, dan NAT Gateways.
- VPC dengan subnet pribadi tempat instance atau beban kerja Anda berjalan.
- Gateway internet yang terpasang pada VPC (atau izin untuk membuatnya).
- Titik akhir AWS Transform VPC `com.amazonaws.region.api.transform` dengan DNS pribadi diaktifkan. Ini adalah titik akhir yang digunakan oleh klien browser aplikasi web. Untuk petunjuk tentang cara membuat titik akhir, lihatAWS Transform dan titik akhir antarmuka (AWS PrivateLink).

Mengonfigurasi kebijakan titik akhir VPC

Jika titik akhir `com.amazonaws.region.api.transform` VPC Anda memiliki kebijakan titik akhir kustom, Anda harus menambahkan pernyataan yang memungkinkan operasi. AWS Transform Tanpa ini, aplikasi web tidak akan dapat melakukan panggilan API melalui titik akhir.

Tambahkan pernyataan berikut ke kebijakan titik akhir VPC Anda. Ganti *AWS_TRANSFORM_PROFILE_ARN* dengan profil AWS Transform layanan Anda ARN, yang dapat Anda temukan di AWS Transform konsol pada halaman Pengaturan.

```
{
    "Statement": [
        {
            "Sid": "AllowAWSTransform",
            "Effect": "Allow",
            "Principal": "*",
            "Action": "*",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:PrincipalArn": [
                        "AWS_TRANSFORM_PROFILE_ARN"
                    ]
                }
            }
        }
    ]
}
```

> **Note**
>
> Jika titik akhir VPC Anda menggunakan kebijakan default (akses penuh), tidak diperlukan perubahan dan Anda dapat melewati langkah ini.

Menyiapkan jalan keluar internet yang dikendalikan

Selesaikan langkah-langkah berikut untuk mengkonfigurasi Network Firewall dengan pemfilteran berbasis domain untuk aplikasi web. AWS Transform

Langkah 1: Buat subnet firewall

Buat subnet kecil /28 yang didedikasikan untuk titik akhir Network Firewall. Subnet ini harus berada di Availability Zone yang sama dengan subnet pribadi Anda.

```
aws ec2 create-subnet \
  --vpc-id your-vpc-id \
  --cidr-block firewall-subnet-cidr \
  --availability-zone your-az \
  --region region
```

Langkah 2: Buat subnet publik untuk NAT Gateway

```
aws ec2 create-subnet \
  --vpc-id your-vpc-id \
  --cidr-block public-subnet-cidr \
  --availability-zone your-az \
  --region region
```

Langkah 3: Konfigurasikan tabel rute subnet publik

Buat tabel rute untuk subnet publik yang mengarahkan lalu lintas internet ke gateway internet.

```
# Create route table
```

```
PUB_RTB=$(aws ec2 create-route-table \
  --vpc-id your-vpc-id \
  --region region \
  --query 'RouteTable.RouteTableId' --output text)

# Add default route to internet gateway
aws ec2 create-route \
  --route-table-id $PUB_RTB \
  --destination-cidr-block 0.0.0.0/0 \
  --gateway-id your-igw-id \
  --region region

# Associate with public subnet
aws ec2 associate-route-table \
  --route-table-id $PUB_RTB \
  --subnet-id public-subnet-id \
  --region region
```

Langkah 4: Buat NAT Gateway

```
# Allocate an Elastic IP
EIP=$(aws ec2 allocate-address --domain vpc \
  --region region --query 'AllocationId' --output text)

# Create NAT Gateway in the public subnet
NAT_ID=$(aws ec2 create-nat-gateway \
  --subnet-id public-subnet-id \
  --allocation-id $EIP \
  --region region \
  --query 'NatGateway.NatGatewayId' --output text)

# Wait for NAT Gateway to become available (~2 minutes)
aws ec2 wait nat-gateway-available \
  --nat-gateway-ids $NAT_ID --region region
```

Langkah 5: Buat grup aturan Network Firewall

Buat grup aturan stateful yang memungkinkan lalu lintas hanya ke domain yang diperlukan oleh aplikasi web. AWS Transform

```
aws network-firewall create-rule-group \
  --rule-group-name transform-webapp-domains \
  --type STATEFUL \
  --capacity 100 \
  --rule-group '{
    "StatefulRuleOptions": {
      "RuleOrder": "STRICT_ORDER"
    },
    "RulesSource": {
      "RulesSourceList": {
        "Targets": [
          ".cloudfront.net",
          ".transform.region.on.aws",
          "region.signin.aws",
          ".s3.region.amazonaws.com",
          "oidc.region.amazonaws.com",
          "portal.sso.region.amazonaws.com",
          "assets.sso-portal.region.amazonaws.com",
          "directory-id.awsapps.com"
        ],
        "TargetTypes": ["TLS_SNI", "HTTP_HOST"],
        "GeneratedRulesType": "ALLOWLIST"
      }
    }
  }' \
  --region region
```

Ganti *region* dengan AWS Wilayah tempat AWS Transform profil Anda diinstal (misalnya,`us-east-1`). Ganti *directory-id* dengan ID direktori Pusat Identitas IAM Anda (misalnya,`d-1234567890`). Anda dapat menemukan ID direktori Anda di konsol Pusat Identitas IAM.

Tabel berikut menjelaskan domain yang diizinkan.

Domain	Tujuan
`.cloudfront.net`	CloudFront CDN - melayani aset statis aplikasi web (JavaScript, CSS, gambar)

Domain	Tujuan
`.transform. region.on.aws`	URL penyewa aplikasi web — browser memuat halaman awal dari domain ini melalui CloudFront
`region.signin.aws`	Halaman pengalihan masuk SSO
`.s3.region.amazonaws.com`	URL presigned S3 — unggahan dan unduhan artefak
`oidc.region.amazonaws.com`	Pertukaran token OIDC untuk otentikasi SSO
`portal.sso. region.amazonaws.com`	Halaman login portal SSO
`assets.sso-portal. region.amazonaws.com`	Aset statis portal SSO (CSS, JavaScript)
`directory-id .awsapps.com`	Portal Pusat Identitas IAM untuk organisasi Anda

> **Note**
>
> `.cloudfront.net`Wildcard memungkinkan lalu lintas ke CloudFront distribusi apa pun, tidak hanya aplikasi AWS Transform web. Filter domain yang lebih sempit tidak dimungkinkan karena IP CloudFront edge dibagi di seluruh distribusi dan inspeksi TLS SNI tidak dapat membedakan distribusi individu di belakang domain yang sama.

> **Note**
>
> Panggilan API untuk `api.transform.region.on.aws` melewati AWS PrivateLink dan tidak memerlukan jalan keluar internet. Mereka tidak terpengaruh oleh firewall.

Langkah 6: Buat kebijakan firewall

Kebijakan harus menggunakan evaluasi `STRICT_ORDER` aturan `drop_established` sebagai tindakan default. Ini memastikan bahwa lalu lintas apa pun yang tidak cocok dengan daftar yang diizinkan dijatuhkan.

```
# Get the rule group ARN
RG_ARN=$(aws network-firewall describe-rule-group \
  --rule-group-name transform-webapp-domains \
  --type STATEFUL --region region \
  --query 'RuleGroupResponse.RuleGroupArn' --output text)

# Create the firewall policy
aws network-firewall create-firewall-policy \
  --firewall-policy-name transform-webapp-policy \
  --firewall-policy "{
    \"StatelessDefaultActions\": [\"aws:forward_to_sfe\"],
    \"StatelessFragmentDefaultActions\": [\"aws:forward_to_sfe\"],
    \"StatefulRuleGroupReferences\": [
      {
        \"ResourceArn\": \"$RG_ARN\",
        \"Priority\": 1
      }
    ],
    \"StatefulEngineOptions\": {
      \"RuleOrder\": \"STRICT_ORDER\"
    },
    \"StatefulDefaultActions\": [\"aws:drop_established\", \"aws:alert_established\"]
  }" \
  --region region
```

Langkah 7: Buat Network Firewall

```
# Get the policy ARN
FW_POLICY_ARN=$(aws network-firewall describe-firewall-policy \
  --firewall-policy-name transform-webapp-policy \
  --region region \
  --query 'FirewallPolicyResponse.FirewallPolicyArn' --output text)

# Create the firewall
aws network-firewall create-firewall \
  --firewall-name transform-webapp-firewall \
  --firewall-policy-arn $FW_POLICY_ARN \
  --vpc-id your-vpc-id \
  --subnet-mappings SubnetId=firewall-subnet-id \
  --region region
```

```
# Wait for the firewall to become READY (3-5 minutes)
while true; do
  STATUS=$(aws network-firewall describe-firewall \
    --firewall-name transform-webapp-firewall \
    --region region \
    --query 'FirewallStatus.Status' --output text)
  echo "Status: $STATUS"
  if [ "$STATUS" = "READY" ]; then break; fi
  sleep 15
done
```

Langkah 8: Dapatkan ID endpoint firewall

```
FW_ENDPOINT=$(aws network-firewall describe-firewall \
  --firewall-name transform-webapp-firewall \
  --region region \
  --query "FirewallStatus.SyncStates.\"your-az\".Attachment.EndpointId" \
  --output text)
echo "Firewall endpoint: $FW_ENDPOINT"
```

Langkah 9: Konfigurasikan tabel rute subnet firewall

Rutekan lalu lintas yang terikat internet dari subnet firewall ke NAT Gateway.

```
FW_RTB=$(aws ec2 create-route-table \
  --vpc-id your-vpc-id \
  --region region \
  --query 'RouteTable.RouteTableId' --output text)

aws ec2 create-route \
  --route-table-id $FW_RTB \
  --destination-cidr-block 0.0.0.0/0 \
  --nat-gateway-id $NAT_ID \
  --region region

aws ec2 associate-route-table \
  --route-table-id $FW_RTB \
```

```
  --subnet-id firewall-subnet-id \
  --region region
```

Langkah 10: Perbarui tabel rute subnet pribadi

Rutekan semua lalu lintas yang terikat internet dari subnet pribadi melalui firewall.

```
aws ec2 create-route \
  --route-table-id private-subnet-route-table-id \
  --destination-cidr-block 0.0.0.0/0 \
  --vpc-endpoint-id $FW_ENDPOINT \
  --region region
```

Jika rute default sudah ada, gunakan `replace-route` sebagai penggantic`reate-route`.

Langkah 11: Tambahkan rute kembali simetris (wajib)

> **Important**
>
> Langkah ini sangat penting. Network Firewall menggunakan inspeksi TLS SNI dan harus melihat kedua arah koneksi TCP. Tambahkan rute di tabel rute subnet NAT Gateway yang mengirimkan lalu lintas kembali yang ditujukan untuk subnet pribadi kembali melalui firewall.

```
aws ec2 create-route \
  --route-table-id $PUB_RTB \
  --destination-cidr-block private-subnet-cidr \
  --vpc-endpoint-id $FW_ENDPOINT \
  --region region
```

Ganti *`private-subnet-cidr`* dengan blok CIDR subnet pribadi Anda (misalnya,`10.0.144.0/20`).

Tanpa routing simetris, firewall hanya melihat satu arah lalu lintas. Mesin inspeksi TLS tidak dapat mengekstrak Indikasi Nama Server (SNI) dari jabat tangan TLS, dan semua aturan berbasis domain akan gagal secara diam-diam.

Langkah 12: Hapus rute default IPv6 (jika ada)

Jika tabel rute subnet pribadi memiliki rute default IPv6 (`::/0`) yang menunjuk langsung ke gateway internet, itu akan melewati firewall untuk tujuan. IPv6-capable Hapus itu:

```
aws ec2 delete-route \
  --route-table-id private-subnet-route-table-id \
  --destination-ipv6-cidr-block ::/0 \
  --region region
```

Verifikasi

Dari instance di subnet pribadi, verifikasi konfigurasi:

```
# Should SUCCEED - web application content via CloudFront (allowed)
curl -vL --connect-timeout 15 \
  'https://tenant-id.transform.region.on.aws'

# Should SUCCEED - API via PrivateLink (does not use firewall)
curl -v --connect-timeout 15 \
  'https://api.transform.region.on.aws/'

# Should FAIL - non-allowlisted domain (blocked by firewall)
curl -v --connect-timeout 15 'https://www.example.com'
# Expected: TLS connection error (firewall drops after SNI inspection)
```

Pemecahan masalah

Panggilan API ke api.transform diblokir oleh firewall Anda

Domain `api.transform.region.on.aws` harus menyelesaikan ke alamat IP pribadi melalui titik akhir VPC dan tidak boleh mencapai firewall internet Anda.

- Verifikasi bahwa Anda telah membuat `com.amazonaws.region.api.transform` titik akhir.
- Verifikasi DNS pribadi diaktifkan pada titik akhir:

```
aws ec2 describe-vpc-endpoints \
  --filters "Name=service-name,Values=com.amazonaws.region.api.transform" \
  --query 'VpcEndpoints[*].[State,PrivateDnsEnabled]' \
  --output table
```

Output yang diharapkan: `available | True`

Aplikasi web tidak memuat (batas waktu koneksi)

- Verifikasi tabel rute subnet pribadi memiliki `0.0.0.0/0` rute yang menunjuk ke titik akhir firewall.
- Verifikasi bahwa tabel rute subnet firewall memiliki `0.0.0.0/0` rute yang menunjuk ke NAT Gateway.
- Verifikasi NAT Gateway dalam `available` keadaan.

Non-allowlisted Domain tidak diblokir

- Periksa `::/0` rute IPv6 yang menunjuk ke gateway internet. Ini melewati firewall. Hapus (Langkah 12).
- Verifikasi routing simetris dikonfigurasi. Tabel rute subnet NAT Gateway harus memiliki rute kembali melalui firewall untuk CIDR subnet pribadi (Langkah 11).
- Verifikasi kebijakan firewall yang digunakan `STRICT_ORDER` dengan `drop_established` dan `alert_established` sebagai tindakan default.

Pertimbangan Biaya

Sumber daya	Perkiraan biaya
Network Firewall	~ $0. 395/hr (~$288/month) per Availability Zone
Gerbang NAT	~ $0. 045/hr (~$33/month) +biaya pemrosesan data
IP elastis (IPv4 publik)	~ $0. 005/hr (~ $3. 60/month)
Pemrosesan data Network Firewall	$0. 065/GB
Pemrosesan data NAT Gateway	$0. 045/GB

Perkiraan biaya dasar kira-kira $ 325/month untuk satu penyebaran Availability Zone. Untuk penerapan produksi, gunakan firewall, NAT Gateway, dan subnet terkait di setiap Availability Zone di mana subnet pribadi ada.

Pembersihan

Untuk menghapus semua sumber daya yang dibuat oleh panduan ini, jalankan perintah berikut dalam urutan terbalik. Ganti placeholder dengan ID sumber daya dari penerapan Anda. Anda dapat menemukan nilai-nilai ini di AWS Management Console atau dengan menggunakan `describe` perintah dari langkah-langkah penyiapan.

```
# Remove return route from NAT Gateway subnet
aws ec2 delete-route --route-table-id pub-rtb-id \
  --destination-cidr-block private-subnet-cidr \
  --region region

# Remove route from private subnet
aws ec2 delete-route \
  --route-table-id private-subnet-route-table-id \
  --destination-cidr-block 0.0.0.0/0 --region region

# Delete Network Firewall (takes ~5 minutes)
aws network-firewall delete-firewall \
  --firewall-name transform-webapp-firewall \
  --region region

# Delete firewall policy and rule group
aws network-firewall delete-firewall-policy \
  --firewall-policy-name transform-webapp-policy \
  --region region
aws network-firewall delete-rule-group \
  --rule-group-name transform-webapp-domains \
  --type STATEFUL --region region

# Delete NAT Gateway (wait ~5 minutes for full deletion)
aws ec2 delete-nat-gateway --nat-gateway-id nat-gateway-id \
  --region region

# Release Elastic IP
aws ec2 release-address --allocation-id eip-allocation-id \
  --region region
```

```
# Delete subnets
aws ec2 delete-subnet --subnet-id firewall-subnet-id \
  --region region
aws ec2 delete-subnet --subnet-id public-subnet-id \
  --region region
```

Memantau AWS Transformasi

Pemantauan adalah bagian penting dalam menjaga keandalan, ketersediaan, dan kinerja AWS Transform dan AWS solusi Anda yang lain. AWS menyediakan alat pemantauan berikut untuk menonton AWS Transform, melaporkan ketika ada sesuatu yang salah, dan mengambil tindakan otomatis bila perlu:

- Amazon CloudWatch memantau AWS sumber daya Anda dan dan aplikasi yang Anda jalankan AWS secara real time. Anda dapat mengumpulkan dan melacak metrik, membuat dasbor yang disesuaikan, dan mengatur alarm yang memberi tahu Anda atau mengambil tindakan saat metrik tertentu mencapai ambang batas yang ditentukan. Misalnya, Anda dapat CloudWatch melacak penggunaan CPU atau metrik lain dari instans Amazon EC2 Anda dan secara otomatis meluncurkan instans baru bila diperlukan. Untuk informasi selengkapnya, lihat Panduan CloudWatch Pengguna Amazon.
- Amazon CloudWatch Logs memungkinkan Anda memantau, menyimpan, dan mengakses file log Anda dari instans Amazon EC2, CloudTrail, dan sumber lainnya. CloudWatch Log dapat memantau informasi dalam file log dan memberi tahu Anda ketika ambang batas tertentu terpenuhi. Anda juga dapat mengarsipkan data log dalam penyimpanan yang sangat durabel. Untuk informasi selengkapnya, lihat Panduan Pengguna Amazon CloudWatch Logs.
- Amazon EventBridge dapat digunakan untuk mengotomatiskan AWS layanan Anda dan merespons secara otomatis peristiwa sistem, seperti masalah ketersediaan aplikasi atau perubahan sumber daya. Acara dari AWS layanan dikirim ke EventBridge dalam waktu dekat. Anda dapat menuliskan aturan sederhana untuk menunjukkan peristiwa mana yang sesuai kepentingan Anda, dan tindakan otomatis mana yang diambil ketika suatu peristiwa sesuai dengan suatu aturan. Untuk informasi selengkapnya, lihat Panduan EventBridge Pengguna Amazon.
- AWS CloudTrailmenangkap panggilan API dan peristiwa terkait yang dibuat oleh atau atas nama AWS akun Anda dan mengirimkan file log ke bucket Amazon S3 yang Anda tentukan. Anda dapat mengidentifikasi pengguna dan akun mana yang dipanggil AWS, alamat IP sumber dari mana panggilan dilakukan, dan kapan panggilan terjadi. Untuk informasi selengkapnya, lihat Panduan Pengguna AWS CloudTrail.

Pencatatan log AWS Ubah panggilan API menggunakan AWS CloudTrail

AWS Transform terintegrasi dengan AWS CloudTrail, layanan yang menyediakan catatan tindakan yang diambil oleh pengguna, peran, atau AWS layanan di AWS Transform. CloudTrail menangkap semua panggilan API untuk AWS Transform sebagai peristiwa. Panggilan yang diambil mencakup panggilan dari konsol AWS Transform dan panggilan kode ke operasi AWS Transform API. Jika Anda membuat jejak, Anda dapat mengaktifkan pengiriman CloudTrail peristiwa secara terus menerus ke bucket Amazon S3, termasuk peristiwa untuk AWS Transform. Jika Anda tidak mengonfigurasi jejak, Anda masih dapat melihat peristiwa terbaru di CloudTrail konsol dalam Riwayat acara. Dengan menggunakan informasi yang dikumpulkan oleh CloudTrail, Anda dapat menentukan permintaan yang dibuat untuk AWS Transform, alamat IP dari mana permintaan dibuat, siapa yang membuat permintaan, kapan dibuat, dan detail tambahan.

Untuk mempelajari selengkapnya CloudTrail, lihat Panduan AWS CloudTrail Pengguna.

AWS Mengubah informasi di CloudTrail

CloudTrail diaktifkan pada Akun AWS saat Anda membuat akun. Ketika aktivitas terjadi di AWS Transform, aktivitas tersebut direkam dalam suatu CloudTrail peristiwa bersama dengan peristiwa AWS layanan lainnya dalam riwayat Peristiwa. Anda dapat melihat, mencari, dan mengunduh acara terbaru di situs Anda Akun AWS. Untuk informasi selengkapnya, lihat Melihat peristiwa dengan Riwayat CloudTrail acara.

Untuk catatan acara yang sedang berlangsung di Anda Akun AWS, termasuk acara untuk AWS Transform, buat jejak. Jejak memungkinkan CloudTrail untuk mengirimkan file log ke bucket Amazon S3. Secara default, saat Anda membuat jejak di konsol, jejak tersebut berlaku untuk semua Wilayah AWS. Jejak mencatat peristiwa dari semua Wilayah di AWS partisi dan mengirimkan file log ke bucket Amazon S3 yang Anda tentukan. Selain itu, Anda dapat mengonfigurasi AWS layanan lain untuk menganalisis lebih lanjut dan menindaklanjuti data peristiwa yang dikumpulkan dalam CloudTrail log. Untuk informasi selengkapnya, lihat berikut:

- Gambaran umum untuk membuat jejak
- CloudTrail layanan dan integrasi yang didukung
- Mengonfigurasi notifikasi Amazon SNS untuk CloudTrail
- Menerima file CloudTrail log dari beberapa wilayah dan Menerima file CloudTrail log dari beberapa akun

Setiap entri peristiwa atau log berisi informasi tentang entitas yang membuat permintaan tersebut. Informasi identitas membantu Anda menentukan hal berikut ini:

- Apakah permintaan itu dibuat dengan kredenal pengguna root atau AWS Identity and Access Management (IAM).
- Apakah permintaan tersebut dibuat dengan kredensial keamanan sementara untuk satu peran atau pengguna gabungan.
- Apakah permintaan itu dibuat oleh AWS layanan lain.

Untuk informasi selengkapnya, lihat Elemen userIdentity CloudTrail .

Memahami AWS Ubah entri file log

Trail adalah konfigurasi yang memungkinkan pengiriman peristiwa sebagai file log ke bucket Amazon S3 yang Anda tentukan. CloudTrail file log berisi satu atau lebih entri log. Peristiwa mewakili permintaan tunggal dari sumber manapun dan mencakup informasi tentang tindakan yang diminta, tanggal dan waktu tindakan, parameter permintaan, dan sebagainya. CloudTrail file log bukanlah jejak tumpukan yang diurutkan dari panggilan API publik, jadi file tersebut tidak muncul dalam urutan tertentu.

Notifikasi

AWS Transform menyediakan pemberitahuan email untuk memberi Anda informasi tentang perubahan akses ruang kerja, pembaruan status pekerjaan, dan permintaan kolaborator yang tertunda.

Pemberitahuan email

Pemberitahuan email diaktifkan secara default, dan dikirim ke pengguna yang memiliki izin baca saja atau lebih tinggi pada suatu pekerjaan. Pengguna dapat mengubah preferensi notifikasi mereka. Ada tiga kategori notifikasi email:

- Pembaruan akses ruang kerja

  Anda menerima pemberitahuan saat ditambahkan ke ruang kerja baru, dan saat peran Anda diubah dalam ruang kerja.
- Email intisari harian

Anda menerima email intisari harian pada pukul 9:00 waktu Pasifik yang merangkum pekerjaan Anda yang sedang berlangsung jika Anda memiliki pekerjaan aktif dengan satu atau lebih permintaan kolaborator terbuka yang diperlukan. Email mencakup tabel yang menunjukkan nama ruang kerja, nama pekerjaan, status pekerjaan, detail permintaan kolaborator, dan tindakan yang diperlukan (terbatas pada 10 baris)

> **Note**
>
> Email intisari harian hanya dikirim untuk pekerjaan yang membutuhkan permintaan kolaborator. Permintaan kolaborator opsional tidak memicu email intisari.

- Pembaruan status Job

  Anda menerima email ketika pekerjaan yang Anda akses selesai atau gagal.

- Kolaborator meminta pembaruan

  Anda menerima email setiap kali tugas membutuhkan masukan, ulasan, atau persetujuan Anda.

Mengelola notifikasi email

Anda dapat mengubah preferensi notifikasi email Anda di aplikasi web dengan mengklik roda penggerak Pengaturan di sudut kanan atas aplikasi, dan memilih Pengaturan Pemberitahuan.

Halaman pengaturan notifikasi menyediakan kontrol berikut:

Jenis Pemberitahuan	Deskripsi	Pengaturan default
Pembaruan Akses Ruang Kerja	Menerima email saat ditambahkan ke ruang kerja, atau saat peran Anda diubah dalam ruang kerja	Diaktifkan
Intisari Permintaan Kolaborator Harian	Terima intisari harian permintaan kolaborator terbuka untuk pekerjaan di ruang kerja yang dapat diakses	Diaktifkan
Pembaruan Penyelesaian Pekerjaan	Menerima email saat pekerjaan selesai atau gagal	Diaktifkan

Jenis Pemberitahuan	Deskripsi	Pengaturan default
Pembaruan Permintaan Kolaborator	Menerima email ketika masukan atau persetujuan diperlukan dari Anda	Nonaktif

Anda dapat menggunakan pengaturan umum untuk mengaktifkan atau menonaktifkan semua notifikasi, atau mengonfigurasi jenis pemberitahuan individual secara terpisah.

Kuota untuk AWS Transformasi

AWS Akun Anda memiliki kuota default, sebelumnya disebut sebagai batas, untuk setiap layanan. AWS Kecuali dinyatakan lain, setiap kuota adalah Region-specific. Anda dapat meminta peningkatan untuk beberapa kuota dan kuota lainnya tidak dapat ditingkatkan.

Untuk melihat kuota AWS Transform, buka konsol Service Quotas. Di panel navigasi, pilih AWS layanan dan pilih AWS Transform.

Untuk meminta peningkatan kuota, lihat Meminta Peningkatan Kuota dalam Panduan Pengguna Service Quotas. Jika kuota belum tersedia di Service Quotas gunakan formulir kenaikan batas.

AWS Akun Anda memiliki kuota berikut yang terkait dengan AWS Transform.

Beban kerja	Dimensi	Kuota	Dapat disesuaikan	Deskripsi
Platform	Ruang kerja Wilayah AWS per akun yang AWS Transform diaktifkan	100 ruang kerja untuk masing-masing didukung Wilayah AWS	Ya	Jumlah maksimum ruang kerja per akun yang AWS Transform diaktifkan per Wilayah AWS
Migrasi VMware	Jumlah server untuk setiap gelombang	Setiap gelombang: 150 server	Tidak	Jumlah maksimum server dalam gelombang
Mainframe	Pekerjaan bersamaan untuk setiap akun Pusat Identitas IAM	10 pekerjaan untuk setiap Wilayah yang didukung	Ya	Jumlah maksimum pekerjaan Mainframe bersamaan di akun IAM Identity Center

Beban kerja	Dimensi	Kuota	Dapat disesuaikan	Deskripsi
	Baris kode untuk setiap pekerjaan	3.000.000 baris kode untuk setiap wilayah yang didukung	Tidak	Jumlah maksimum baris kode untuk setiap pekerjaan.
	Analisis baris kode bulanan	100.000.000 baris kode untuk setiap wilayah yang didukung	Ya	Jumlah maksimum baris kode untuk tujuan pekerjaan Analyze Code dalam satu bulan kalender
	Pembuatan dokumen bisnis baris kode bulanan	50.000.000 baris kode untuk setiap wilayah yang didukung	Ya	Jumlah maksimum baris kode untuk tujuan pekerjaan Generasi Dokumentasi Bisnis dalam satu bulan kalender
	Pembuatan dokumen teknis baris kode bulanan	50.000.000 baris kode untuk setiap wilayah yang didukung	Ya	Jumlah maksimum baris kode untuk tujuan pekerjaan Generasi Dokumentasi Teknis dalam satu bulan kalender

Beban kerja	Dimensi	Kuota	Dapat disesuaikan	Deskripsi
	Transformasi baris kode	10.000.000 baris kode untuk setiap wilayah yang didukung	Ya	Jumlah maksimum baris kode yang diubah dalam satu bulan kalender
	Dekomposisi baris kode bulanan	50.000.000 baris kode untuk setiap wilayah yang didukung	Ya	Jumlah maksimum baris kode untuk tujuan pekerjaan Dekomposisi dalam satu bulan kalender
	Reformasi baris kode bulanan	50.000.000 baris kode untuk setiap wilayah yang didukung	Ya	Jumlah maksimum baris kode untuk tujuan pekerjaan Reforge dalam satu bulan kalender
	Garis kode bulanan logika bisnis yang diekstraksi	50.000.000 baris kode untuk setiap wilayah yang didukung	Ya	Jumlah maksimum baris kode untuk tujuan pekerjaan Extracted Business Logic dalam satu bulan kalender

Beban kerja	Dimensi	Kuota	Dapat disesuaikan	Deskripsi
.NET	.NET baris kode bulanan (web & IDE)	1.000.000 baris kode untuk setiap Wilayah yang didukung	Ya	Baris maksimum kode aplikasi. NET yang dapat diubah dalam satu bulan kalender di setiap akun IAM Identity Center yang didukung Wilayah AWS
	Pekerjaan .NET bersamaan dengan Microsoft Visual Studio Extension untuk setiap akun IAM Identity Center	10 pekerjaan untuk setiap Wilayah yang didukung	Ya	Jumlah maksimum pekerjaan transformasi.NET bersamaan di setiap akun IAM Identity Center yang didukung Wilayah AWS saat menggunakan Microsoft Visual Studio Extension.

Beban kerja	Dimensi	Kuota	Dapat disesuaikan	Deskripsi
	Pekerjaan web.NET bersamaan untuk setiap akun IAM Identity Center	5 pekerjaan untuk setiap Wilayah yang didukung	Ya	Jumlah maksimum pekerjaan transformasi.NET bersamaan di setiap akun Pusat Identitas IAM yang didukung Wilayah AWS saat menggunakan aplikasi web AWS Transform.
Transformasi kustom	Definisi transformasi khusus	1.000	Ya	Jumlah maksimum definisi transformasi kustom
	Ukuran definisi transformasi kustom	10 MB	Tidak	Ukuran total maksimum semua file yang termasuk dalam definisi transformasi, termasuk referensi dokumen.

Beban kerja	Dimensi	Kuota	Dapat disesuaikan	Deskripsi
	Transformasi kustom permintaan percakapan CLI bulanan	1.000.000	Ya	Jumlah maksimum permintaan percakapan transformasi kustom yang dibuat di CLI diizinkan dalam satu bulan kalender
Penilaian	Batas jumlah server untuk setiap penilaian	Setiap penilaian: 30.000 server	Tidak	Jumlah maksimum server dalam satu pekerjaan penilaian

Wilayah yang Didukung untuk AWS Transform

> **Note**
>
> Jika Anda membuat permintaan yang mengharuskan AWS Transform untuk mengambil informasi dari Wilayah keikutsertaan yang tidak tercantum di halaman ini, AWS Transform dapat melakukan panggilan ke Wilayah tersebut. Untuk mengelola akses ke Wilayah AWS Transform dapat melakukan panggilan ke, lihatKeamanan di AWS Transformasi.

Topik ini menjelaskan di Wilayah AWS mana Anda dapat menggunakan AWS Transform. Untuk informasi selengkapnya Wilayah AWS, lihat Menentukan Wilayah AWS akun mana yang dapat digunakan dalam Panduan AWS Account Management Referensi.

Data Anda mungkin diproses di Wilayah yang berbeda dari Wilayah tempat Anda menggunakan AWS Transform. Untuk informasi tentang pemrosesan lintas wilayah di AWS Transform, lihatCross-region pengolahan. Untuk informasi tentang tempat penyimpanan data selama pemrosesan, lihat Perlindungan data.

Didukung Wilayah AWS (diaktifkan secara default)

Anda dapat membuat AWS Transform ruang kerja di bawah ini. Wilayah AWS Wilayah ini diaktifkan secara default - Anda tidak perlu mengaktifkannya sebelum digunakan. Untuk informasi selengkapnya, lihat Wilayah yang diaktifkan secara default.

- AS Timur (Virginia Utara)
- Eropa (Frankfurt)
- Asia Pasifik (Mumbai)
- Asia Pasifik (Sydney)
- Asia Pasifik (Tokyo)
- Eropa (London)
- Asia Pasifik (Seoul)
- Kanada (Pusat)
- Amerika Selatan (São Paulo) - Hanya agen modernisasi mainframe

Ruang kerja tempat Anda membuat pekerjaan menentukan Wilayah AWS pekerjaan. Untuk membuat pekerjaan di Wilayah yang berbeda, Anda harus menggunakan ruang kerja berbeda yang ada di Wilayah yang Anda inginkan.

Untuk proyek VMware, Wilayah ruang kerja digunakan untuk penemuan. Namun, Anda dapat menentukan Wilayah yang berbeda sebagai target migrasi. Wilayah target tersebut adalah tempat beban kerja Anda di-host saat migrasi selesai. Untuk informasi selengkapnya tentang Wilayah AWS pertimbangan migrasi VMware, termasuk daftar kemungkinan Wilayah target, lihat. the section called “Wilayah target yang didukung”

AWS dukungan untuk AWS Transform

AWS Transform memungkinkan Anda untuk membuat dan mengelola kasus AWS Support melalui obrolan jika Anda mengalami masalah selama pekerjaan transformasi Anda. Kasus Anda secara otomatis diperkaya dengan konteks pekerjaan dan diarahkan ke tim dukungan yang sesuai. Jika Anda adalah pelanggan Countdown Premium yang aktif, teknisi yang Anda tunjuk akan menghubungi untuk mengelola penyelesaian masalah. Pelajari lebih lanjut di AWS Countdown Premium. Jika tidak, kasus Anda akan diselidiki oleh tim AWS Transform dukungan dalam antrian dukungan mereka.

Prasyarat

Sebelum Anda dapat menggunakan manajemen kasus dukungan, Anda perlu:

- Paket AWS Support aktif - Bisnis, Perusahaan On-Ramp, atau Perusahaan
- AWS Transform Aplikasi yang ada

Mengaktifkan Support Case Management

1. Buka AWS Transform konsol.
2. Pada panel navigasi, silakan pilih Pengaturan.
3. Di bawah Support, pilih Aktifkan manajemen kasus Support.
4. Pilih Simpan perubahan.

AWS Transform menggunakan peran terkait layanan (SLR) untuk membuat dan mengelola kasus dukungan atas nama Anda. Tidak diperlukan izin IAM tambahan.

Membuat Kasus Support

Untuk membuat kasus dukungan, gunakan antarmuka obrolan:

1. Mintalah untuk membuat kasus dukungan. Contoh:
    - “Buat kasus dukungan untuk pekerjaan transformasi saya yang gagal”
    - “Saya butuh bantuan dengan kesalahan pekerjaan, buat kasus”
2. Obrolan meminta Anda untuk informasi yang diperlukan:

- Penjelasan singkat tentang masalah Anda
- Penjelasan terperinci

3. Konfirmasikan pembuatan kasus saat diminta.

Kasus ini secara otomatis diisi dengan konteks dari pekerjaan transformasi Anda, termasuk:

- ID dan status Job
- Metadata Job
- Log eksekusi Job (log kerja)
- Rencana Job

Kasus ini secara otomatis ditetapkan berdasarkan jenis pekerjaan dan kategori masalah Anda.

Melihat Kasus Support

Untuk melihat kasus dukungan Anda, minta obrolan:

- “Buat daftar kasus dukungan saya”
- “Tunjukkan pada saya kasus dukungan terbuka”
- “Bagaimana status kasus saya?”

Obrolan akan menampilkan:

- ID Kasus
- Subjek
- Status, seperti Terbuka, Tertunda, Terselesaikan, atau Ditutup
- Tanggal terakhir diperbarui
- ID pekerjaan terkait

Untuk melihat detail kasus tertentu:

- “Tunjukkan detail untuk kasus [case-id]”
- “Apa pembaruan terbaru pada case [case-id]?”

Menambahkan Komunikasi ke Kasus

Untuk menambahkan pesan ke kasus yang ada:

1. Tanyakan obrolan: “Tambahkan pesan ke case [case-id]”
2. Berikan pesan Anda saat diminta.
3. Obrolan akan mengonfirmasi bahwa pesan telah ditambahkan.

Anda akan menerima notifikasi saat AWS Support merespons kasus Anda.

Menyelesaikan Kasus

Untuk menyelesaikan kasus dukungan:

1. Tanyakan obrolan: “Selesaikan kasus [case-id]”
2. Konfirmasikan resolusi saat diminta.

> **Note**
>
> Anda hanya dapat menyelesaikan kasus yang telah ditangani oleh AWS Support.

Menonaktifkan Support Case Management

1. Buka AWS Transform konsol.
2. Pada panel navigasi, silakan pilih Pengaturan.
3. Di bawah Support, pilih Disable Support case management.
4. Pilih Simpan perubahan.

> **Note**
>
> Menonaktifkan manajemen kasus Support menghapus antarmuka manajemen kasus berbasis obrolan tetapi tidak menutup kasus yang ada. Anda masih dapat mengakses kasus Anda melalui AWS Support Center.

Penggunaan Telemetri

Plugin AWS Transform IDE, AWS Transform Agent Skill, dan AWS Transform Kiro Power mengumpulkan telemetri penggunaan secara default selama eksekusi transformasi. Telemetri terdiri dari titik data yang berbeda, seperti, nama IDE (misalnya, VS Code atau Kiro), nama agen AI (misalnya, Claude Code atau OpenAI Codex), dan mode eksekusi (lokal atau jarak jauh). Data ini digunakan oleh AWS Transform untuk memprioritaskan pengujian kompatibilitas, serta latensi dan keandalan.

Untuk menonaktifkan telemetri, beri tahu agen selama sesi obrolan Anda bahwa Anda tidak ingin telemetri dipancarkan. Penyisihan ini hanya berlaku untuk sesi obrolan saat ini dan harus diulang untuk setiap sesi baru. Jika Anda menjalankan AWS Transform CLI secara langsung, atur variabel `ATX_DISABLE_TELEMETRY=true` lingkungan.

Data Operasional

Kami dapat mengumpulkan sinyal operasional terkait penggunaan AWS Transform oleh Anda, seperti fitur yang Anda gunakan, log kesalahan, dan metrik berdasarkan umpan balik bahasa alami Anda (seperti apakah Anda menyetujui melanjutkan dengan rencana transformasi atau memberikan instruksi untuk mengulangi rencana yang diusulkan) untuk menyediakan layanan dan menilai kinerja layanan.

Ubah log

Tabel berikut menjelaskan rilis penting dan update untuk AWS Transform.

Date	judul	Deskripsi	Tautan
2026-07-16	AWS Transform for Migrations menambahkan laporan ringkasan ruang kerja yang dapat diunduh	AWS Transform for Migrations sekarang memungkinkan Anda membuat laporan ringkasan ruang kerja sebagai PDF yang dapat diunduh. Laporan mengkonso lidasikan data dari semua pekerjaan migrasi di ruang kerja Anda, termasuk status pekerjaan, kemajuan langkah alur kerja, perencana an gelombang, migrasi jaringan, konfigurasi landing zone, kemajuan rehost, dan artefak yang dihasilkan.	Mengunduh laporan ringkasan ruang kerja
2026-07-03	AWS Alat penemuan Transform menambahkan dukungan Oracle Database	Alat penemuan AWS Transform sekarang mengumpulkan metadata Oracle terperinci langsung dari database Anda di seluruh VMware,, dan server bare	Penemuan Oracle Database

Date	judul	Deskripsi	Tautan
		metal. Hyper-V Anda dapat menemukan instance Oracle melalui koneksi SQL langsung. Alat ini mengumpulkan enumerasi Container Database/Pluggable Database (CDB/PDB), inventaris komponen, dan ukuran file data. Ini juga menangkap bendera topologi untuk Real Application Clusters (RAC), Data Guard, dan arsitektur multitenant. Jika Anda belum mengonfigurasi kredensi database, alat akan kembali ke OS-level deteksi. Semua versi Oracle didukung.	

Date	judul	Deskripsi	Tautan
2026-06-22	Dukungan wilayah Amerika Selatan (São Paulo) untuk agen modernisasi mainframe	AWS Transformasi untuk agen modernisasi mainframe sekarang tersedia di Wilayah Amerika Selatan (São Paulo) (sa-east-1). Menambahkan sa-east-1 ke wilayah yang didukung dan tabel inferensi lintas wilayah.	Wilayah yang Didukung, Cross-region inferensi
2026-06-22	AWS Transform untuk migrasi VMware mendukung pelokalan	AWS Transform untuk migrasi VMware sekarang mendukung pelokalan. Anda dapat mengubah bahasa tampilan aplikasi web saat bekerja dengan alur kerja migrasi VMware dengan memilih ikon Pengaturan dan memilih bahasa pilihan Anda.	Pengaturan lokalisasi

Date	judul	Deskripsi	Tautan
2026-06-18	AWS Transform for Migrations mendukung semua wilayah AWS komersial sebagai target migrasi	AWS Transform untuk migrasi VMware mendukung semua wilayah AWS komersial sebagai target migrasi, tidak termasuk Timur Tengah (Bahrain) dan Timur Tengah (UEA). Wilayah target migrasi adalah AWS wilayah tempat sumber daya yang dimigrasi digunakan, termasuk zona pendaratan, infrastruktur jaringan, dan rehosting server.	Wilayah target migrasi yang didukung

Date	judul	Deskripsi	Tautan
2026-06-18	AWS Transform for Migrations memungkinkan Anda untuk melampirkan ENI yang ada ke template peluncuran Anda	AWS Transform memungkinkan Anda untuk melampirk an Elastic Network Interfaces (ENIs) yang ada ke template peluncuran untuk migrasi Anda. AWS Transform menampilk an semua ENI yang ada yang ditemukan di akun target Anda, dan Anda dapat menandai mereka sehingga tersedia untuk digunakan melalui konfigura si peluncuran saat instance diluncurkan.	Penandaan sumber daya ENI untuk migrasi
2026-06-18	AWS Transform for Migrations memungkinkan Anda mengonfigurasi pengaturan replikasi dan peluncuran	AWS Transform memungkinkan Anda mengonfigurasi pengaturan replikasi dan pengaturan peluncuran untuk migrasi Anda. Anda dapat mengatur konfigurasi per akun target Anda atau di seluruh server sumber tertentu.	Pengaturan replikasi dan peluncuran

Date	judul	Deskripsi	Tautan
2026-06-17	Model-to-model penilaian migrasi untuk beban kerja AI generatif	AWS Transform sekarang menawarkan transformasi kustom migrasi model-ke-model. Transformasi menilai beban kerja AI generatif Anda. Ini menghasilkan rencana migrasi untuk pindah dari penyedia pihak ketiga ke Amazon Bedrock. AI-powered Agen memindai basis kode Anda untuk mengidentifikasi setiap AI SDK dan model yang digunakan. Agen mengumpulkan persyaratan migrasi melalui pertanyaan interaktif. Kemudian memetakan model ke Amazon Bedrock yang setara dengan perbandingan biaya transparan dan perubahan kode siap produksi. Mendukung migrasi dari OpenAI, Google Gemini, penggunaan SDK Antropik langsung,	Apa posting baru, Transformasi kustom

Date	judul	Deskripsi	Tautan
		dan model sumber terbuka melalui LitellM atau Ollama. Menangani integrasi SDK langsung dan pola yang dibungkus kerangka kerja termasuk LangChain,, CreWai, LlamaIndex, dan lapisan perutean multi-penyedia. LangGraph Tersedia tanpa biaya tambahan di luar harga AWS Transform standar.	

Date	judul	Deskripsi	Tautan
2026-06-16	AWS Transform untuk mainframe memberikan alur kerja penataan ulang yang dapat dilacak	AWS Transform untuk mainframe sekarang memberikan pengalaman penataan ulang yang terhubung dan dapat dilacak dari penilaian hingga pembuatan kode. Perusahaan yang menjalankan z/OS COBOL dan PL/I beban kerja dapat menilai portofolio mereka untuk mengidentifikasi fungsi bisnis diskrit, mengekstrak aturan bisnis, menghasilkan persyaratan siap pengembangan, dan menghasilkan kode cloud-native yang dapat dilacak dalam satu alur kerja yang terhubung. Setiap persyaratan ditelusuri kembali ke kode sumber untuk auditabilitas penuh.	Apa posting baru, Dokumentasi

Date	judul	Deskripsi	Tautan
2026-06-16	AWS Transform sekarang mendukung Amazon FSx untuk NetApp ONTAP sebagai penyimpan an target (Pratinjau Publik)	Anda sekarang dapat memigrasikan beban kerja penyimpanan blok ke Amazon FSx untuk NetApp ONTAP selain Amazon EBS sebagai bagian dari rehosting komputasi . AWS Transform mereplikasi data penyimpanan blok langsung ke FSx untuk volume ONTAP dalam gelombang migrasi yang sama yang menangani komputasi dan jaringan.	Apa posting baru, Dokumentasi

Date	judul	Deskripsi	Tautan
2026-06-05	Dukungan penilaian Amazon RDS for SQL Server TCO	Sekarang Anda dapat memperkirakan biaya saat memigrasikan database SQL Server lokal ke Amazon RDS for SQL Server. Gunakan penilaian AI-powered TCO di AWS Transform untuk menganalisis lingkungan Anda dan mendapatkan rekomendasi instans database yang optimal. Penilaian ini mendukung opsi Bring Your Own Media (BYOM) dan License Included (LI) dengan Database Savings Plans untuk penghematan hingga 20%.	Apa posting baru, Dokumentasi

Date	judul	Deskripsi	Tautan
2026-06-04	AWS Transform alat penemuan sekarang didukung sebagai sumber input migrasi jaringan	Anda sekarang dapat menggunakan alat penemuan AWS Transform sebagai sumber untuk migrasi jaringan bersama ModelizeIt, memungkinkan migrasi jaringan hybrid untuk lingkungan yang menjalankan beban kerja VMware dan non-VMware.	Panduan Pengguna
2026-05-27	AWS Transform dokumentasi kustom menambahkan contoh keterampilan sisi klien dan praktik terbaik	Dokumentasi keterampilan sisi klien sekarang mencakup contoh praktis untuk kepatuhan Dockerfile, keamanan migrasi basis data, penegakan kebijakan Terraform, dan pembantu penghentian API, bersama dengan praktik terbaik untuk membangun dan berbagi keterampilan.	Dokumentasi

Date	judul	Deskripsi	Tautan
2026-05-22	Kemampuan penilaian migrasi agen baru sekarang tersedia dengan Transform AWS	AWS Transform sekarang menawarkan kemampuan penilaian migrasi tingkat lanjut termasuk skenario bagaimana-jika, asumsi yang dapat disesuaikan, dukungan format file yang fleksibel, dan beberapa fitur penilaian biaya kepemilikan total (TCO) baru.	Apa posting baru, Dokumentasi
2026-05-20	AWS Transform mendeteksi VPC yang ada selama tinjauan migrasi jaringan	AWS Transform sekarang mendeteksi VPC yang ada di akun target Anda selama tinjauan migrasi jaringan. Anda dapat melihat VPC yang ada di samping VPC yang dipetakan, mengidentifikasi konflik CIDR, dan menyelesaikannya sebelum penerapan.	Dokumentasi

Date	judul	Deskripsi	Tautan
2026-05-14	AWS Agen transformasi sekarang tersedia di Kiro, Claude, Cursor, dan Codex	AWS Agen transformasi sekarang dapat diakses melalui kekuatan Kiro, plugin agen, dan server AWS Transform MCP, memungkinkan pengembang mengkonsumsi kemampuan transformasi langsung dari lingkungan pengembangan pilihan mereka.	Apa itu posting baru
2026-05-14	AWS Transform memperkenalkan toolkit pembangun agen untuk membangun agen transformasi yang disesuaikan	Mitra dan pelanggan sekarang dapat membangun, berbagi, dan mendaftarkan agen transformasi khusus menggunakan toolkit pembuat agen Kiro power, mengintegrasikan alat kustom dan alur kerja dengan Transform. AWS	Apa itu posting baru

Date	judul	Deskripsi	Tautan
2026-05-14	AWS Transform menambahkan asisten AI agen ke AWS Toolkit for Visual Studio	Pengembang.NET sekarang dapat memodernisasi aplikasi melalui pengalaman percakapan, langkah demi langkah yang dipandu langsung di Visual Studio, dengan konteks penuh dipertahankan di seluruh IDE dan konsol web.	Apa itu posting baru
2026-05-14	AWS Transform sekarang mendukung toko artefak milik pelanggan	Anda dapat mengonfigurasi bucket Amazon S3 Anda sendiri dan secara opsional mengenkripsi artefak dengan kunci AWS KMS Anda sendiri, memberikan kontrol penuh atas tempat artefak transformasi disimpan.	Apa posting baru, Panduan Pengguna

Date	judul	Deskripsi	Tautan
2026-05-14	AWS Transform menambahkan rekomendasi strategi migrasi (7Rs)	Dengan AWS Transform, Anda sekarang dapat menganalisis inventaris yang Anda temukan dan mendapatkan rekomendasi strategi migrasi untuk setiap server dan aplikasi. Rekomendasi mengikuti tujuh strategi migrasi standar industri (7R). Setiap rekomendasi mencakup layanan AWS target yang disarankan, skor kepercayaan diri, dan alasan di baliknya. Anda dapat menghasilkan dasbor HTML interaktif, PDF, atau PowerPoint output Microsoft.	Rekomendasi 7Rs

Date	judul	Deskripsi	Tautan
2026-05-11	AWS Transform menambahkan kemampuan containerization selama migrasi	AWS Transform sekarang mendukung replatforming aplikasi ke container selama migrasi, mengotomatiskan pembuatan Dockerfile, pembuatan image container dengan pemindaian CVE, dan penerapan ke Amazon ECS atau Amazon EKS.	Apa posting baru, Panduan Pengguna
2026-05-01	AWS Transform sekarang menawarkan agen migrasi BI untuk Power BI dan Tableau ke Quick	Partner-built agen dari Wavicle Data Solutions mengonversi dasbor Power BI dan Tableau menjadi aset Cepat dalam alur kerja Transform. AWS	Apa itu posting baru
2026-04-21	AWS Transform custom sekarang tersedia di enam AWS Wilayah tambahan	AWS Transform custom berkembang ke Asia Pasifik, Kanada, dan Eropa (London), mencapai total delapan AWS Wilayah.	Apa itu posting baru

Date	judul	Deskripsi	Tautan
2026-04-15	AWS Transform menambahkan pembuatan landing zone dalam alur kerja migrasi	Menambahkan pembuatan landing zone langsung dalam alur kerja migrasi. AWS Transform memeriksa fondasi yang ada, merekomendasikan struktur Unit Organisasi (OU) dan akun target, dan menawarkan penerapan otomatis atau Infrastructure as Code (IAc) output (CloudFormation, AWS CDK, atau Landing Zone Accelerator (LZA) format).	Apa posting baru, Dokumentasi

Date	judul	Deskripsi	Tautan
2026-04-15	AWS Transform menambahkan diagram dan laporan perencanaan migrasi	Anda sekarang dapat membuat diagram interaktif dan laporan analitis selama perencanaan migrasi. Jenis diagram meliputi peta topologi jaringan, grafik ketergant ungan aplikasi, grafik gelombang Gantt, dan bagan umum. Jenis laporan meliputi penilaian risiko, rekomendasi 7R, dan laporan analitik khusus. Output tersedia sebagai file HTML, PDF, atau Microsoft interaktif PowerPoint (.pptx).	Diagram dan pelaporan
2026-04-14	AWS Transform sekarang tersedia di Kiro dan VS Code	AWS Transformasi kustom Transform sekarang dapat diakses dari Kiro dan VS Code, dengan status pekerjaan dibagikan di seluruh permukaan IDE, CLI, dan konsol web.	Apa itu posting baru

Date	judul	Deskripsi	Tautan
2026-04-10	AWS Transform menambahkan DHCP dengan pemetaan grup keamanan untuk migrasi jaringan	Menambahkan dukungan untuk DHCP dengan pemetaan grup keamanan selama migrasi jaringan. Jika memiliki beban kerja DHCP-enabled migrasi, kini Anda dapat memastikan aturan keamanan tetap valid setelah migrasi. Ini berlaku untuk referensi grup keamanan in-VPC dan aturan keluar lintas VPC dan lintas akun melalui Transit Gateway.	Dokumentasi
2026-03-31	AWS Transform custom mengumumkan ketersediaan umum analisis basis kode otomatis	Transformasi analisis basis kode yang komprehensif mencapai ketersediaan umum, menghasilkan dokumen arsitektur, laporan utang teknis, dan rekomendasi modernisasi dalam bahasa apa pun.	Apa itu posting baru

Date	judul	Deskripsi	Tautan
2026-03-31	AWS Transform custom memperken alkan transformasi terkelola baru untuk memodernisasi kode dalam skala besar	Tujuh transform asi terkelola baru ditambahkan, termasuk Node.js peningkatan (ketersed iaan umum), optimasi Java, migrasi Log4j, peningkatan Angular Angular-to-React, dan peningkatan Vue (akses awal).	Apa itu posting baru
2026-03-20	AWS Transform menambahkan API publik untuk migrasi jaringan	Menambahkan API publik untuk migrasi jaringan, memungkinkan mitra dan akses terprogra m ke kemampuan migrasi jaringan.	Referensi API
2026-01-14	AWS Transform custom menambahk an AWS PrivateLi nk dukungan dan memperluas ke Eropa (Frankfurt)	AWS Transform custom sekarang mendukung akses VPC pribadi melalui AWS PrivateLink dan tersedia di Eropa (Frankfurt) selain US East (Virginia N.).	Apa itu posting baru

Date	judul	Deskripsi	Tautan
2025-12-23	AWS Transform memungkinkan konversi jaringan untuk migrasi pusat data hybrid	AWS Transform for VMware secara otomatis mengonversi konfigurasi jaringan pusat data hybrid (VLAN, rentang IP) ke AWS VPC, subnet, dan grup keamanan tanpa pemetaan manual.	Apa itu posting baru
2025-12-01	AWS meluncurkan AWS Transform custom untuk mempercepat modernisasi seluruh organisasi	AWS Transform custom mencapai ketersediaan umum, memungkinkan organisasi untuk mengotomatiskan transformasi kode berulang dalam skala melalui satu perintah CLI.	Apa itu posting baru
2025-12-01	AWS Transform menambahkan kemampuan AI agen baru untuk migrasi VMware perusahaan	AWS Agen VMware Transform sekarang dapat menemukan lingkungan lokal, memetakan dependensi, menghasilkan rencana gelombang migrasi, dan bermigrasi melintasi target HyperV, Nutanix, KVM, dan bare-metal.	Apa itu posting baru

Date	judul	Deskripsi	Tautan
2025-12-01	AWS Transform meluncurkan agen AI untuk modernisasi Windows full-stack	Agen Windows full-stack baru mengotomatiskan transformasi aplikasi. NET dan database SQL Server ke Aurora PostgreSQ L, menyebarkan ke wadah di Amazon ECS atau EC2 Linux.	Apa itu posting baru
2025-12-01	AWS Transform memperluas kemampuan transformasi.NET dan meningkat kan pengalaman pengembang	AWS Transform menambahk an dukungan. NET 10, Formulir ASP.NET Web ke porting Blazor, dan pengalaman IDE interaktif yang disempurnakan melalui AWS Toolkit for Visual Studio.	Apa itu posting baru
2025-12-01	AWS Transform untuk mainframe memberika n kemampuan otomatisasi pengujian baru	AWS Transform untuk mainframe memperkenalkan pembuatan rencana pengujian otomatis, skrip pengumpul an data uji, dan otomatisasi kasus uji untuk mempercepat pengujian modernisa si mainframe.	Apa itu posting baru

Date	judul	Deskripsi	Tautan
2025-12-01	AWS Transform untuk mainframe sekarang mendukung penataan ulang aplikasi	Analisis data dan aktivitas baru ditambah kemampuan ekstraksi logika bisnis memungkinkan dekomposisi aplikasi mainframe lama menjadi arsitektur cloud-native.	Apa itu posting baru
2025-11-13	AWS Transform mengotomatiskan konfigurasi jaringan Landing Zone Accelerator	AWS Transform untuk VMware secara otomatis menghasilkan file YAMM LZA-compatible jaringan, merampingkan pengaturan infrastruktur cloud multi-akun dari lingkungan VMware.	Apa itu posting baru
2025-07-16	AWS Transform untuk mainframe memperkenalkan kemampuan refactoring kode dan logika bisnis yang disempurnakan	AWS Transform untuk mainframe menambahkan ketersediaan umum reforge (restrukturisasi kode) dan ekstraksi logika bisnis tingkat aplikasi yang diperluas untuk meningkatkan kualitas kode.	Apa itu posting baru

Date	judul	Deskripsi	Tautan
2025-05-15	AWS Transform untuk VMware sekarang tersedia secara umum	AWS Transform for VMware, layanan AI agen yang mengotoma tiskan siklus hidup modernisasi VMware penuh dari penemuan hingga penerapan, mencapai ketersedi aan umum.	Apa itu posting baru
2025-05-15	AWS Transform untuk mainframe sekarang tersedia secara umum	AWS Transform for mainframe, layanan AI agen pertama untuk memodernisasi z/OS aplikasi IBM dalam skala besar, mencapai ketersedi aan umum dengan kemampuan analisis, dekomposisi, dan manajemen pekerjaan baru.	Apa itu posting baru

Unduh data ini.

Sejarah dokumen untuk AWS Panduan Pengguna Transform

Tabel berikut menjelaskan riwayat dokumen untuk AWS Transform User Guide.

Perubahan	Deskripsi	Tanggal
Analisis dan remediasi portofolio modernisasi berkelanjutan	Menambahkan babak baru yang mendokumentasikan fitur modernisasi berkelanj utan AWS Transform (`atx ct`subperintah) untuk analisis dan remediasi kode otomatis. Lihat modernisasi berkelanj utan.	3 Juni 2026
Jaringan Brownfield: visibilitas VPC yang ada	Menambahkan deteksi VPC yang ada selama tinjauan migrasi jaringan. AWS Transform secara otomatis mendeteksi VPC yang sudah digunakan di akun target Anda dan menandai konflik CIDR dengan VPC yang dipetakan . Lihat VPC yang ada di akun target Anda.	18 Mei 2026
Menambahkan format keterampilan, keterampilan sisi klien, dan log subagent untuk Transform custom AWS	Definisi transformasi sekarang menggunak an format keterampilan (`SKILL.md`dengan frontmatt er YAMM dan folder opsional) `references/` . Menambahk an keterampilan sisi klien untuk memperluas agen dengan kemampuan	11 Mei 2026

	tambahan, dan log subagent untuk debugging yang lebih baik. Membutuhkan CLI versi 2.0 atau yang lebih baru. Lihat Definisi dan Client-Side Keterampilan Transformasi.	
Menambahkan kontainerisasi kode sumber	AWS Transform sekarang dapat mengkontainerisasi aplikasi Anda menggunakan AI generatif. Berikan kode sumber Anda dari repositor i Git atau unggahan zip, dan AWS Transform menganali sisnya, menghasilkan artefak Docker, menerbitkan gambar kontainer ke Amazon ECR, membuat infrastruktur sebagai kode, dan menyebarkan ke Amazon Elastic Container Service atau Amazon Elastic Kubernetes Service. Lihat Kontainerisasi kode sumber.	7 Mei 2026

Menambahkan operasi pengoptimalan jaringan dan rekomendasi terpandu	Ditambahkan Langkah 5: Tinjau dan optimalkan jaringan, dengan operasi VPC dan subnet (hapus, kecualikan, gabungkan, pisahkan, ubah ukuran, ganti nama, ubah alamat IP) dan rekomendasi terpandu untuk standardisasi penamaan, tinjauan ruang lingkup, ukuran kanan kapasitas, tinjauan keamanan, dan konsolidasi VPC. Lihat Meninjau dan mengoptimalkan jaringan.	April 27, 2026
AWS Ubah kustom tersedia di 6 wilayah baru	Menambahkan dukungan untuk ca-central-1, eu-west-2, ap-northeast-1, ap-northeast-1, ap-northeast-2, ap-southeast-2, ap-southeast-2, dan ap-south-1 di kustom Transform.AWS	April 21, 2026
Dokumentasi izin konektor yang diperbarui	Menambahkan rincian izin terperinci untuk Akun AWS konektor target, yang mencakup izin S3, MGN, Migration Hub, migrasi beban kerja, dan penyediaan infrastruktur landing zone. Lihat Connect target Akun AWS s.	April 19, 2026

Jenis pekerjaan migrasi VMware yang diperbarui	Menambahkan Discovery dan perencanaan migrasi, zona pendaratan, dan jenis pekerjaan migrasi zona pendaratan, jaringan, dan server. Mengganti nama Migrasi jaringan hanya menjadi Migrasi jaringan dan perencanaan Migrasi untuk Membangun rencana migrasi. Lihat Jenis pekerjaan migrasi VMware.	April 19, 2026
Target Akun AWS Connect yang direstrukturisasi s	Merestrukturisasi pengaturan konektor menjadi tiga langkah: Pemilihan tipe migrasi, perjanjian MAP, dan konfigura si Konektor, termasuk wilayah target yang didukung, peran akun, panduan administrator yang didelegasikan, dan peran IAM. Lihat Connect target Akun AWS s.	April 19, 2026
Menambahkan wilayah target server-migrasi tambahan	Mendokumentasikan 14 wilayah komersial tambahan yang tersedia untuk pekerjaan migrasi server tanpa migrasi jaringan, dengan persyaratan daftar izin hingga Q3 2026. Lihat Wilayah target yang didukung.	April 19, 2026

Ditambahkan Migrasi server chapter	Menambahkan chapter Migrate server baru yang mencakup alur kerja rehost lengkap termasuk struktur akun, peran IAM, penyiapan administrator yang didelegasikan, penyebaran agen replikasi, pengujian, dan pemotongan.	April 19, 2026
Ditambahkan Build landing zone chapter	Menambahkan chapter Build landing zone baru dengan detail izin konektor untuk penyediaan landing zone.	April 19, 2026
Pemetaan dan referensi kelompok keamanan	Menambahkan strategi pemetaan grup keamanan (MAP, MAP_DHCP, SKIP) dan dokumentasi referensi grup keamanan ke jaringan Migrasi.	April 13, 2026
Auto-conversion untuk jenis file jaringan yang tidak didukung	Menambahkan dukungan untuk konversi otomatis jenis file konfigurasi jaringan yang tidak didukung di jaringan Migrasi.	April 6, 2026
Menambahkan kebijakan kunci KMS untuk migrasi database	Menambahkan Enkripsi untuk bagian migrasi database.	26 Maret 2026
Kebijakan AWS terkelola baru yang didokumentasikan	DBModProvisioningAndMigrationBagian yang ditambahkan.	24 Maret 2026
Kebijakan AWS terkelola baru yang didokumentasikan	DBModDiscoveryAndAssessmentBagian yang ditambahkan.	24 Maret 2026

Pembuatan diagram jaringan	Menambahkan pembuatan diagram jaringan opsional (format Mermaid dan PNG) ke jaringan Migrasi.	Maret 18, 2026
Dukungan file konfigurasi firewall dan SDN	Menambahkan dukungan untuk file konfigurasi Palo Alto Networks, Fortinet FortiGate, dan Cisco ACI untuk migrasi jaringan di jaringan Migrasi.	Februari 11, 2026
Penandaan khusus untuk sumber daya jaringan	Menambahkan dukungan penandaan tingkat pekerjaan dan VPC-level kustom untuk migrasi jaringan di jaringan Migrasi.	Januari 28, 2026
Ditambahkan dokumentasi dukungan	Ditambahkan AWS dukungan untuk bagian.	Desember 23, 2025
Kebijakan AWS terkelola baru yang didokumentasikan	Ditambahkan AWSTransformCustomFullAccess, AWSTransformCustomExecuteTransformations, dan AWSTransformCustomManageTransformationsbagian.	Desember 7, 2025
Migrasi VMware	Memperbarui bagian migrasi VMware.	Desember 1, 2025
Kebijakan AWS terkelola yang diperbarui	Diperbarui AWSServiceRoleForAWSTransformdokumentasi untuk mencerminkan izin tambahan.	Desember 1, 2025
Wilayah yang didukung	Memperbarui topik Wilayah yang didukung.	Desember 1, 2025

Modernisasi mainframe	Memperbarui bagian modernisasi Mainframe.	Desember 1, 2025
Memulai	Memperbarui topik Memulai.	Desember 1, 2025
Full-stack Modernisasi Windows	Menambahkan topik modernisasi Full-stack Windows, yang mencakup bagian .NET yang diperbarui dan bagian modernisasi SQL Server baru.	Desember 1, 2025
Alat penemuan	Menambahkan bagian alat Discovery.	Desember 1, 2025
Kebijakan AWS terkelola yang diperbarui	Memperbarui AWSTransformApplicationECSDeploymentPolicydengan izin inspeksi peran IAM yang diperluas, pembuatan peran terkait layanan ECS, dan izin KMS untuk dukungan enkripsi ECR.	November 22, 2025
Kebijakan AWS terkelola yang diperbarui	Memperbarui AWSTransformApplicationDeploymentPolicydengan jaringan EC2 tambahan, inspeksi peran IAM, manajemen bucket S3, dan izin enkripsi KMS.	November 22, 2025
Kolektor penemuan	Menambahkan alat penemuan AWS Transform > dokumentasi.	Oktober 31, 2025
Diagram arsitektur migrasi VMware ditambahkan	Menambahkan diagram arsitektur migrasi VMware di jaringan Migrasi.	Oktober 23, 2025

Pembaruan hasil migrasi VMware	Migrasi yang diperbarui menghasilkan informasi di jaringan Migrasi.	Oktober 12, 2025
Pembaruan hasil migrasi VMware	Migrasi yang diperbarui menghasilkan informasi di jaringan Migrasi.	Oktober 12, 2025
Dokumentasi konektor S3	Menambahkan dokumentasi konektor S3 untuk aplikasi web.NET.	Oktober 8, 2025
Topik konektor pengguna baru	Ditambahkan topik konektor pengguna.	September 21, 2025
Kebijakan AWS terkelola yang diperbarui	Diperbarui AWSServiceRoleForAWSTransformdokumentasi.	September 17, 2025
Dukungan penilaian penyimpanan	Menambahkan dukungan file NetApp Data Infrastructure Insights (DII) untuk penilaian penyimpanan.	September 2, 2025
Pembaruan wilayah target	Menambahkan tiga wilayah target yang didukung.	Agustus 31, 2025
Pembaruan kebijakan terkelola	Menambahkan izin read-only IAM Identity Center untuk digunakan oleh. AWSServiceRoleForAWSTransform	Agustus 29, 2025
Kebijakan AWS terkelola baru yang didokumentasikan	AWSTransformApplicationDeploymentPolicyBagian yang ditambahkan.	Agustus 28, 2025
titik akhir antarmuka	Ditambahkan topik titik akhir antarmuka.	Juli 9, 2025

Rilis awal	Rilis awal Panduan Pengguna AWS Transform.	15 Mei 2025

Terjemahan disediakan oleh mesin penerjemah. Jika konten terjemahan yang diberikan bertentangan dengan versi bahasa Inggris aslinya, utamakan versi bahasa Inggris.